

How to reason from inconsistent ontologies with prioritized assertional bases?

Salem Benferhat, Zied Bouraoui, Huma Chadhry, Mohd Shafry Bin Mohd Rahim, Karim Tabia, Abdelmoutia Telli

▶ **To cite this version:**

Salem Benferhat, Zied Bouraoui, Huma Chadhry, Mohd Shafry Bin Mohd Rahim, Karim Tabia, et al.. How to reason from inconsistent ontologies with prioritized assertional bases?. The 12th International Conference on SIGNAL IMAGE TECHNOLOGY & INTERNET BASED SYSTEMS, Nov 2016, Naples, Italy. hal-02896411

HAL Id: hal-02896411
https://hal.science/hal-02896411

Submitted on 10 Jul 2020

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.

How to reason from inconsistent ontologies with prioritized assertional bases?

Salem Benferhat
Artois University - Nord de France, CRIL CNRS UMR 8188, Artois, F-62307 Lens, France.
Email: benferhat@cril.univ-artois.fr

Zied Bouraoui
Computer Science and Informatics Cardiff University, UK
Email: BouraouiZ@cardiff.ac.uk

Huma Chadhry
Technology Malaysiay 81310 UTM Johor Bahru, Johor, Malaysia
Email: huma.bicse@gmail.com

Mohd Shafry Bin Mohd Rahim Fc
Technology Malaysiay 81310 UTM Johor Bahru, Johor, Malaysia
Email: shafry@utm.my

Karim Tabia
Artois University - Nord de France, CRIL CNRS UMR 8188, Artois, F-62307 Lens, France.
Email: tabia@cril.univ-artois.fr

Abdelmoutia Telli
Artois University - Nord de France, CRIL CNRS UMR 8188, Artois, F-62307 Lens, France.
Email: telli@cril.univ-artois.fr

***Abstract*—In many applications, such as video processing, assertions are often provided by several and potentially conflicting sources having different reliability levels. This paper deals with the problem of handling inconsistency in lightweight ontologies when the set of assertions (facts) is prioritized. We propose a safe and efficient way to restore consistency using the concept of free assertions; assertions that are not involved in conflicts. Our approach allows the selection of one consistent assertional base, called a preferred repair. Selecting a unique repair is important since it allows an efficient handling of queries. The last part of the paper contains an illustrative example using video data.**

***Keywords*-Inconsistency; Prioritized knowledge bases; Lightweight Ontologies**

I. Introduction

Description Logics (DLs) are formal frameworks for representing and reasoning with ontologies. A DL knowledge base is built upon two distinct components: a terminological base (called *TBox*) representing generic knowledge, and an assertional base (called *ABox*) containing facts or assertions.

Recently, a particular interest was given to *Ontology Based Data Access* (OBDA), in which the ontological view (i.e. the *TBox*) is used to offer a better exploitation of assertions (i.e. the ABox) when querying them (*e.g.* [18], [23]). A crucially important problem that arises in OBDA is how to manage conflicting information. In such a setting, an ontology is usually verified and validated while the assertions can be provided in large quantities by various and unreliable sources that may be inconsistent with respect to the ontology. Moreover, it is often too expensive to manually check and validate all the assertions. This is why it is very important in OBDA to reason in the presence of inconsistency. Many works (*e.g.* [9], [11], [17], [19], [20]), basically inspired by the ones in the database area (*e.g.* [1], [8], [14]) and propositional logic approaches (*e.g.* [5]), deal with inconsistency in DLs by proposing several inconsistency-tolerant inferences, called *semantics*. These semantics are based on the notion of a maximally assertional (or ABox) repair which is closely related to the notion of a database repair [16] or a maximally consistent subbase used in the propositional logic setting (*e.g.* [22]). An ABox repair is simply an assertional subbase which is consistent with an ontology.

In many applications, assertions are often provided by several and potentially conflicting sources having different reliability levels. Moreover, a given source may provide different sets of uncertain assertions with different confidence levels. Gathering such sets of assertions gives a prioritized or a stratified assertional base (*i.e.* ABox). The role of priorities in handling inconsistency is very important and it is largely studied in the literature within the propositional logic setting (*e.g.* [7]). Several works have also studied the notion of priority when querying inconsistent databases (*e.g.* [21], [24]) or DLs knowledge bases (*e.g.* [4], [10], [15]).

The context of this paper is the one of handling inconsistency in lightweight ontologies when the ABox is prioritized. We use *DL-Lite* [2], an important tractable fragment of DLs, as an example of lightweight ontologies which is well-suited for OBDA [17].

In the presence of conflicting information, there is always a compromise that one needs to reach between the expressiveness and computational issues. Having multiple repairs often allows to derive more conclusions than if one repair is only used. However, query answering from multiple repairs is generally more expensive than query answering from a single repair. In fact, reasoning from a single repair can be viewed as an approximation of reasoning from multiple repairs. Recently, a so-called non-defeated repair has been proposed in [4]. This paper proposes a characterization of the non-defeated repair using the concept of accepted assertions. This leads to select a unique preferred repair from initial inconsistent

knowledge base. Selecting only one preferred repair is important since, once computed, it allows an efficient query answering.

The rest of this paper is organized as follows: Section II-A provides the needed background on *DL-Lite*. Section III presents some elementary concepts on inconsistency handling such as the concepts of conflicts, repairs and free assertions. Section IV presents the so-called non-defeated repair and its characterization using the concept of accepted assertions. Section V introduces the notion of a prioritized deductive closure and its use in non-defeated repair. The last section provides a potential application of our approach to video data.

II. *DL-Lite* AND PRIORITIZED ASSERTIONAL BASE

A. DL-Lite: A Brief Refresher

This section briefly recalls *DL-Lite* logics. For the sake of simplicity, we only consider $DL\text{-}Lite_R$ language [12] and we will simply use *DL-Lite* instead of *DL-Lite*$_R$. Note that the results of this paper can be extended in a straightforward way to any tractable *DL-Lite* as far as computing ABox conflicts is done in polynomial time. This is true for *DL-Lite*$_{core}$ (a particular case of *DL-Lite*$_R$) and *DL-Lite*$_F$. The *DL-Lite* language is defined as follows:

$$\begin{array}{lllll} R \longrightarrow & P \mid P^- & & E \longrightarrow & R \mid \neg R \\ B \longrightarrow & A \mid \exists R & & C \longrightarrow & B \mid \neg B \end{array}$$

where A is an atomic concept, P is an atomic role and P^- is the inverse of P. B (*resp.* C) is called basic (*resp.* complex) concept and role R (*resp.* E) is called basic (*resp.* complex) role. A knowledge base (KB) is a couple $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.

A TBox includes a finite set of inclusion axioms on concepts and on roles respectively of the form: $B \sqsubseteq C$ and $R \sqsubseteq E$. The $ABox$ contains a finite set of atomic concepts and role assertions respectively of the form $A(a)$ and $P(a,b)$ where a and b are two individuals.
The semantics of a *DL-Lite* knowledge base is given in terms of interpretations. An interpretation $\mathcal{I}=(\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $.^{\mathcal{I}}$ that maps each individual a to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each A to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role P to $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Furthermore, the interpretation function $.^I$ is extended in a straightforward way for concepts and roles as follows:

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

An interpretation I is said to be a model of a concept (*resp.* role) inclusion axiom, denoted by $I \models B \sqsubseteq C$ (*resp.* $I \models R \sqsubseteq E$), if and only if $B^I \subseteq C^I$ (*resp.* $R^I \subseteq E^I$). Similarly, we say that an interpretation I is a model of a membership assertion $A(a)$ (*resp.* $P(a,b)$), denoted by $I \models A(a)$ (*resp.* $I \models P(a,b)$), if and only if $a^I \in A^I$ (*resp.* $(a^I, b^I) \in P^I$). A knowledge base $\mathcal{K}$ is called consistent if it admits at least one model, otherwise $\mathcal{K}$ is said to be inconsistent. A TBox $\mathcal{T}$ is said to be incoherent if there exists at least a concept C such that for each interpretation $\mathcal{I}$ which is a model of $\mathcal{T}$, we have $C^{\mathcal{I}}=\emptyset$.

B. Prioritized Assertional Bases

A prioritized assertional base (or a prioritized ABox), simply denoted by $\mathcal{A}=(\mathcal{S}_1, \cdots, \mathcal{S}_n)$, is a tuple of sets assertions. The sets S_i's are called layers or strata. Each layer $\mathcal{S}_i$ contains the set of assertions having the same level of priority i and they are considered as more reliable than the ones present in a layer $\mathcal{S}_j$ when $j > i$. Hence, $\mathcal{S}_1$ contains the most important assertions while $\mathcal{S}_n$ contains the least important assertions.

Throughout this paper and when there is no ambiguity we simply use 'prioritized *DL-Lite* KB $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$' to refer to a *DL-Lite* KB with a prioritized ABox of the form $\mathcal{A}=(\mathcal{S}_1, \cdots, \mathcal{S}_n)$.

This paper proposes methods to deal with inconsistent *DL-Lite* KB. The input of our method is a prioritized *DL-Lite* knowledge base $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$ with $\mathcal{A}=(\mathcal{S}_1, \cdots, \mathcal{S}_n)$. The output of our approaches is a standard *DL-Lite* knowledge base $\mathcal{K}'=\langle \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{R}$ is not prioritized (namely, just a set of assertions). $\mathcal{K}$ and $\mathcal{K}'$ have the same terminological base. $\mathcal{R}$ will be called a preferred repair. Then a query q is said to follow from $\mathcal{K}$ if it can be derived, using the standard *DL-Lite* inference, from $\mathcal{K}'$. Let us first recall important concepts for handling inconsistency when no priority is available between assertions.

III. INCONSISTENCY-TOLERANT REASONING FOR PRIORITIZED *DL-Lite* ASSERTIONAL BASES.

A. The concept of repairs

Within the OBDA setting, we assume that $\mathcal{T}$ is coherent and hence its elements are not questionable in the presence of conflicts. Coping with inconsistency can be done by first computing the set of consistent subsets of assertions (not necessarily maximal), called repairs, then using them to perform inference (i.e. query answering). More formally a repair is defined as follows:

Definition 1: Let $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$ be a flat (not prioritized) *DL-Lite* KB. A subbase $\mathcal{R} \subseteq \mathcal{A}$ is said to be a repair if $\langle \mathcal{T}, \mathcal{R} \rangle$ is consistent. And $\mathcal{R}$ is said to be a maximally inclusion-based repair of $\mathcal{K}$, denoted by *MAR*, if $\langle \mathcal{T}, \mathcal{R} \rangle$ is consistent and $\forall \mathcal{R}'$: $\mathcal{R} \subsetneq \mathcal{R}'$,$\langle \mathcal{T}, \mathcal{R}' \rangle$ is inconsistent.

In the rest of this paper, we will use the term 'flat' to express the fact that there is no priority between different assertions of an ABox. According to the definition of *MAR*, adding any assertion f from $\mathcal{A} \setminus \mathcal{R}$ to $\mathcal{R}$ entails the inconsistency of $\langle \mathcal{T}, \mathcal{R} \cup \{f\} \rangle$. Moreover, the maximality in *MAR* is used in the sense of set inclusion. We denote by *MAR*($\mathcal{A}$) the set of *MAR* of $\mathcal{A}$ with respect to $\mathcal{T}$. The definition of *MAR* coincides with the definition of $ABox$ repair proposed in [16].

Using the notion of repair, coping with inconsistency in flat *DL-Lite* knowledge bases can be done by applying standard query answering either using the whole set of repairs (universal entailment or AR-entailment [16]) or only using one repair.

B. Free Assertions and Conflict Sets

We now introduce the notion of a conflict. It is a minimal subset C of assertions of $\mathcal{A}$ such that $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{C} \rangle$ is inconsistent.

Definition 2: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a *DL-Lite* KB. A subbase $\mathcal{C} \subseteq \mathcal{A}$ is said to be an assertional conflict of $\mathcal{K}$ iff $\langle \mathcal{T}, \mathcal{C} \rangle$ is inconsistent and $\forall\ f \in \mathcal{C}$, $\langle \mathcal{T}, \mathcal{C} \setminus \{f\} \rangle$ is consistent.

From Definition 2, removing any fact f from $\mathcal{C}$ restores the consistency of $\langle \mathcal{T}, \mathcal{C} \rangle$. In *DL-Lite*, when the TBox is coherent, a conflict involves exactly two assertions [13]. We denote by $\mathcal{C}(\mathcal{A})$ the set of conflicts in $\mathcal{A}$.

A nice feature of *DL-Lite* is that computing the set of conflicts is done in polynomial time [3], namely $Conf$ is in P.

We now introduce the notion of non-conflicting or free elements.

Definition 3: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be *DL-Lite* KB. An assertion $f \in \mathcal{A}$ is said to be *free* if and only if $\forall c \in \mathcal{C}(\mathcal{A})$: $f \notin c$.

Intuitively, *free* assertions correspond to elements that are not involved in any conflict. We denote by *free*($\mathcal{A}$) the set of *free* assertions in $\mathcal{A}$. The notion of *free* elements is originally proposed in [6] in a propositional logic setting. Within a *DL-Lite* setting, $free(\mathcal{A})$ is computed in polynomial time thanks to the fact that computing conflicts is done in polynomial time.

The folowing Lemma 1 rephrases the set of free elements using the set of maximally inclusion-based repairs:

Lemma 1: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a *DL-Lite* KB. Then:

$$\text{free}(\mathcal{A}) = \bigcap_{X \in MAR(\mathcal{A})} X.$$

For flat *DL-Lite* knowledge bases, the $free$-entailment (entailment based on free assertions) is equivalent to the IAR-entailment proposed in [16]. In fact, in the context of propositional logic the concept of a maximal consistent subset has been introduced before the concept of repairs [22]. Besides, the concept of $free$ entailment have been introduced in [6]. In the rest of this paper, we will only use the notaion $free(\mathcal{A})$ to describe the set of assertions that are not responssible of conflict in $\langle \mathcal{T}, \mathcal{A} \rangle$.

IV. Acceped Prioritized Assertions

The $free$ entailment repair can be viewed as a safe way to deal with inconsistency. The term *safe* is used by opposition to the term *risky* or *adventurous* with respect to the derived conclusions. This section provides an extension of $free$ entailment repair when assertional bases are prioritized. We first need to introduce the concept of defeated assertion.

Definition 4: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a prioritized *DL-Lite* KB. Let f be an assertion of $\mathcal{A}$. Then the assertion $f \in S_i$ is said to be defeated if:

- $\mathcal{K}$=$\langle \mathcal{T}, \{f\} \rangle$ is inconsistent, or
- There exists an assertion $g \in S_j$ such that $\langle \mathcal{T}, \{f, g\} \rangle$ is inconsistent and $j \in i$.

Basically, an assertion f is said to be defeated if it is contradicted by some priority assertion. In the case where an assertion is not defeated, it is called accepted. More precisely:

Definition 5: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a prioritized *DL-Lite* KB. Let f be an assertion of $\mathcal{A}$. Then the assertion $f \in S_i$ is said to be accepted if and only if it is not defeated.

It turns out that the set of accepted assertions is exactly equal to the so-called no-defeated repair introduced in [4]. The idea in the construction of no-defeated repair is to iteratively retrieve, layer per layer, the set of $free$ elements. More precisely:

Definition 6: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a prioritized *DL-Lite* KB. We define the non-defeated repair, denoted by $nd(\mathcal{A})$=$\mathcal{S}'_1 \cup \ldots \cup \mathcal{S}'_n$, as follows:

$$\forall i = 1, .., n : \mathcal{S}'_i = free(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i).$$

Namely, $nd(\mathcal{A}) =$

$$free(\mathcal{S}_1) \cup free(\mathcal{S}_1 \cup \mathcal{S}_2) \cup \ldots \cup free(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n).$$

Hence an important result of this paper is to provide a characterization of the so-called non-defeated repair using the concept of accepted beliefs.

The definition of non-defeated subbase is an adaptation of the definition proposed in [7] within a propositional logic setting. However, contrarily to the propositional setting, as we will see later, the non-defeated repair can be applied on $\mathcal{A}$ or its deductive closure $cl(\mathcal{A})$ which leads to two different ways to select a single preferred repair. Besides the non-defeated repair is computed in polynomial time in a *DL-Lite* setting while its computation is hard in a propositional logic setting.

Algorithm 1 gives the computation of the non-defeated repair. Algorithm 1 first computes the set of conflicts (step 1). Step 2 simply initializes $nd(\mathcal{A})$ to an empty set. The expression:

$$\{f : f \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i \text{ and } \exists\ g \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i \text{ such that } \{f, g\} \in \mathscr{C}\}$$

represents the set of conflicting elements in $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$. Hence, Step 4 computes the set of free elements in $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$.
Step 5 adds this result to $nd(\mathcal{A})$. Clearly, Algorithm 1 straightforwardly implements Definition **??**. In Algorithm 1 the set of conflicts is computed once. Hence, the compexity of Algorithm 1 is
$Conf$ (step 1) plus $\mathcal{O}(n)$ (step 2-6), where n is the number of strata in the *DL-Lite* knowledge base $\mathcal{K}$.

The first nice feature of non-defeated repair is that it is consistent. Indeed, Recall that:

$$nd(\mathcal{A}) = free(\mathcal{S}_1) \cup free(\mathcal{S}_1 \cup \mathcal{S}_2) \cup \ldots \cup free(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n).$$

Recall also that if C is a conflict then either C is singleton or C is a doubleton. Now, assume that $nd(\mathcal{A})$ is inconsistent. Then this means that there exists a conflict C of $\langle \mathcal{T}, \mathcal{A} \rangle$ such that $C \subseteq nd(\mathcal{A})$.

Assume that $C = \{f\}$ is a singleton and S_i is the first layer where $f \in S_i$ (namely, $\forall j < i, f \notin S_j$). This means that:

- $\forall\ j < i,\ f \notin free(S_1 \cup \cdots \cup free(S_1 \cup \cdots \cup S_j)$ (since $f \notin S_1 \cup \cdots \cup S_{i-1}$),
- $\forall j \geq i, f \notin free(S_1, \cdots, S_j)$ (since $free$ only contains non-conflicting information).

Hence, $f \notin free(S_1) \cup \cdots \cup free(S_1 \cup \cdots \cup S_n)$. Namely, $f \notin nd(\mathcal{A})$.

Now, assume that $C = \{f, g\}$ is a doubleton. Let S_i $(resp.\ S_j)$ be the first layer containing f $(resp.\ g)$. Let us assume that $i \leq j$. Then clearl

$$C \nsubseteq free(S_1) \cup \cdots \cup free(S_1 \cup \cdots \cup S_{j-1})$$

since $f \nsubseteq S_1 \cup \cdots \cup S_1 \cup \cdots \cup S_{j-1}$. Besides, for all $k \geq j$, we have:

$$free(S_1 \cup \cdots \cup S_k) \cap C = \emptyset.$$

Hence, using the definition of $free$ assertion, we get:

$$C \nsubseteq free(S_1) \cup \cdots \cup free(S_1 \cup \cdots \cup S_n).$$

Hence $nd(\mathcal{A})$ contains no conflict and it is consistent. The second nice feature when using the set of accepted belief is that the complexity of computing $nd(\mathcal{A})$ is in P. Indeed, recall the computing conflicts is done in a polynomial time. Since the set of free assertions can be obtained in a linear time with respect to the set of conflicts, then the whole computation of $nd(\mathcal{A})$ is also done in polynomial time.

Algorithm 1 Non-Defeated Repair

Input: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{A} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$
Output: A flat assertional base $nd(\mathcal{A})$
1: $\mathscr{C} \leftarrow C(\mathcal{A})$ {List of conflicts}
2: $nd(\mathcal{A}) \leftarrow \emptyset$
3: **for** $i = 1$ to n **do**
4: $\quad F_i = (\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i) - \{f : f \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$ and $\exists\ g \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$ such that $\{f, g\} \in \mathscr{C}\}$
5: $\quad nd(\mathcal{A}) \leftarrow nd(\mathcal{A}) \cup F_i$
6: **return** $nd(\mathcal{A})$

V. Accepted Assertions on a Prioritized Closure

We now introduce the concept of a prioritized closure. In fact, the three preferred repairs given in the previous sections can be either defined on $\langle \mathcal{T}, \mathcal{A} \rangle$ or on $\langle \mathcal{T}, cl(\mathcal{A}) \rangle$ where $cl(\mathcal{A})$ denotes the deductive closure of a set of assertions $\mathcal{A}$ and is defined as follows.

Definition 7: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a flat *DL-Lite* KB. Let $\mathcal{T}_p$ be the set of all positive inclusion axioms of $\mathcal{T}$[1]. We define the deductive closure of $\mathcal{A}$ with respect to $\mathcal{T}$ as follows: $cl(\mathcal{A})$=$\{B(a)$: $\langle \mathcal{T}_p, \mathcal{A} \rangle \models B(a)$ where, B is a concept of $\mathcal{T}$ and a is an individual of $\mathcal{A}\} \cup \{R(a, b)$: $\langle \mathcal{T}_p, \mathcal{A} \rangle \models R(a, b)$, where R is a role of $\mathcal{T}$ and a,b are individuals of $\mathcal{A}\}$.

The use of a deductive closure of an ABox fully makes sense in DL languages. The following definition extends Definition 7 to the prioritized case.

Definition 8: Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a prioritized *DL-Lite* KB. We define a prioritized closure of $\mathcal{A}$ with respect to $\mathcal{T}$, simply denoted by $cl(\mathcal{A})$, as follows:
$cl(\mathcal{A}) = (\mathcal{S}'_1, \ldots, \mathcal{S}'_n)$ where:

$$\forall i = 1, .., n : \mathcal{S}'_i \quad = \quad cl(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{i-1} \cup \mathcal{S}_i).$$

The first motivation of Definition 8 is that if an assertion f is derived from $\langle \mathcal{K}, S_1 \cup \cdots \cup S_n \rangle$ then f should belong to $cl(\mathcal{A})$. The second motivation is that if an assertion f is believed to some rank i then it should also be believed to all ranks that are higher than i. Namely, if f is derived from $\langle \mathcal{K}, S_1 \cup \cdots \cup S_i \rangle$ then $\forall j \geqslant i, f \in S'_j$.

One way to get a larger set of accepted assertions is to use $cl(\mathcal{A})$ instead of $\mathcal{A}$. Namely, we define a closed non-defeated repair, denoted by $clnd(\mathcal{A})$, as $\mathcal{S}'_1 \cup \ldots \cup \mathcal{S}'_n$, such that:

$$\forall i = 1, .., n : S'_i = free(cl(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)). \quad (1)$$

Clearly, $nd(\mathcal{A}) \subseteq clnd(\mathcal{A})$ and $cl(nd(\mathcal{A})) \subseteq clnd(\mathcal{A})$. The converse is false.

Example 1: Let us the following knowledge base where: $\mathcal{T} = \{A \sqsubseteq \neg B, B \sqsubseteq C, A \sqsubseteq C\}$ and

[1]Positive inclusion axioms are of the form $B_1 \sqsubseteq B_2$.

$\mathcal{A} = \mathcal{S}_1 = \{A(a), B(a)\}$.
One can check that $nd(\mathcal{A}) = \emptyset$, $cl(nd(\mathcal{A})) = \emptyset$ while $nd(cl(\mathcal{A})) = clnd(\mathcal{A}) = \{C(a)\}$. ■

VI. Example

This section contains a brief description of a potential application of our approach to classifying and solving conflicts in a collection of dance videos issued from different sources. Motion in video carries important information which is of a multi-fold nature. Motion sometimes need to be interpreted for understanding or anticipating any immediate reaction to the motion. Hence, the significance of motion perception in a video is seen in several of the present systems and it remains to be active research area. Movements in dances encompass different types of information; sacred rituals, social dialogue or cultural expressions and more. This calls for the need to completely and precisely describe and process motion information and make it digitally available for further processing. Several applications can be reached by dance video automatic annotations, such as dance video retrieval, classification of video databases or animation of dances using modeling of stored information.

A video is a sequence of frames which extend over time. A set of frames can be referred to as a dance segment once a human completes a dance step in a composed dance. Sequence of actions that display a motion which is isolated and makes up a dancing vocabulary can be referred to as a dance step. Improvisation in a dance refers to simultaneous dance moves, which are not previously composed.

Dance moves need to categorized as expressions such as Time Steps in Cha-Cha-Cha or Passo Basico in Samba dance [2]. Our aime is to explore how different dance steps in a Malaysian dance video can be categorized and described to extract knowledge from the video. For instance, in the dance video Inang, we can describe the step Side Bend as a position in which the upper body and the raised arms arch in left or right direction and the feet tap toe in the opposite direction, creating the form of an arch on right and then left side. The dance step with the name Side Bend are segmented from the entire video and this segment of dance movement is labeled as Side Bend.

In the presence of a collection of dance segments issued from different sources (cameras for instance), the problem of conflict may arise. To illustrate this situation, let us consider an example where we assume that we only have the following concepts:

- Forwardmove,
- Backgroundmove,
- FastForwardmove,
- SlowForwardmove,
- FastBackgroundmove,
- SlowBackgroundmove,
- StartForwardmove, and
- StartBackgroundmove.

These concepts concern dances segment that represent direct moves. The concept Forwardmove (resp. SlowForwardmove and FastForwardmove) lists the set of dance segments that represent a forward move (resp. a slow and fast forward move). Similarly, The concept Backgroundmove (resp. Slowbackgroundmove and Fastbackgroundmove) lists the set of dance segments that represent a background move (resp. a slow and fast background move). The concept StartForwardmove (resp. StartBackgroundmove) lists the set of dance segments that represent a starting forward (resp. background) move in a video.

We also assume that we only have one relation:

- DanceSegment: gives for each of dance video the list of dance segments

The terminological base is expressed by the following TBox:

Paxiom 1:	FastForwardmove $\sqsubseteq$ Forwardmove
Paxiom 2:	SlowForwardmove $\sqsubseteq$ Forwardmove
Paxiom 3:	StartForwardmove $\sqsubseteq$ Forwardmove
Paxiom 4:	FastBackgroundmove $\sqsubseteq$ Backgroundmove
Paxiom 5:	SlowBackgroundmove $\sqsubseteq$ Backgroundmove
Paxiom 6:	StartBackgroundmove $\sqsubseteq$ Backgroundmove
Paxiom 7:	StartForwardmove $\sqsubseteq$ SlowForwardmove
Paxiom 8:	StartBackgroundmove $\sqsubseteq$ SlowBackgroundmove
Paxiom 9:	Forwardmove $\sqsubseteq \exists$ DanceSegment$^-$
Paxiom 10:	backgrounddmove $\sqsubseteq \exists$ DanceSegment$^-$
Naxiom1:	Forwardmove $\sqsubseteq \neg$ Backgroundmove

Assume that we have four individuals: i) two dances segments s_1 and s_2, and ii) two videos v_1 and v_2. Assume that we only have four assertions facts given by the ABox $\mathcal{A} = (S_1, S_2)$ with
$S_1 = \{Backgroundmove(s_1), DanceSegment(v_1, s_1)\}$,
$S_2 = \{StartForwardmove(s_1), DanceSegment(v_1, s_2)\}$, and
$S_3 = \{SlowBackgroundmove(s_1), FastBackgroundmove(s_2)\}$.

This assertional base is assumed to be provided by different sources (different cameras for example) having different reliability levels. This knowledge base is clearly inconsistent. Indeed, using the assertion $StartForwardmove(s_1)$ and the positive inclusion axiom $Paxiom3$ one can derive $Forwardmove(s_1)$. This derived fact together with the assertion $Backgroundmove(s_1)$ contradict the negative axiom $Naxiom1$. Without the use of priorities, the free entailment will simple lead to:

$Free(\mathcal{A}) = \{DanceSegment(v_1, s_1), DanceSegment(v_1, s_2), FastBackgroundmove(s_2)\}$ Namely,

only assertions that are not conserved by the conflict are preserved. When there is a priority relation between assertional facts, one can go one step further in solving conflicts. In our example, $Backgroundmove(s_1)$ is assumed to be issued from a source more reliable than the one that delivers the assertion $StartForwardmove(s_1)$. Hence, using the non-defeated entailment, we get :

[2]http://www.dancadesalao.com/agenda/ingles.php

$$
\begin{aligned}
nd(\mathcal{A}) \quad = \quad & \{Backgroundmove(s_1), \\
& SlowBackgroundmove(s_1), \\
& DanceSegment(v_1, s_1), \\
& DanceSegment(v_1, s_2), \\
& FastBackgroundmove(s_2)\}.
\end{aligned}
$$

Clearly, $nd(\mathcal{A})$ is clearly larger than $Free(\mathcal{A})$. The nice feature of our approach is once $nd(\mathcal{A})$ is computed, query the initial inconsistent knowledge base can be done efficiently.

VII. Conclusion

This paper focused on how to select a single preferred repair from a prioritized inconsistent *DL-Lite* knowledge base. Selecting only one repair is important since it allows efficient querying answering once the preferred repair is computed. A future work is to apply our approach to query Malaysian dance videos in presence of possible conflicts.

Acknowledgments

This work is supported by Research and Innovation Staff Exchange (RISE) H2020-MSCA-RISE-2015 project called High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content or AniAge Project [EU H2020 project-AniAge (691215)].

References

[1] M. Arenas, E. Leopoldo. 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.

[2] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)*, 36:1–69, 2009.

[3] S. Benferhat, Z. Bouraoui, O. Papini, and E. Würbel. A prioritized assertional-based revision for DL-Lite knowledge bases. In *European Conference on Logics in Artificial Intelligence*, volume 8761 of *LNCS*, pages 442–456. Springer, 2014.

[4] S. Benferhat, Z. Bouraoui, and K. Tabia. How to select one preferred assertional-based repair from inconsistent and prioritized DL-Lite knowledge bases? In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1450–1456. AAAI Press, 2015.

[5] S. Benferhat, D. Didier, and P. Henri. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study part 1: The flat case. *Studia Logica*, 58(1):17–45, 1997.

[6] S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Knowledge Representation and Reasoning*, pages 673–684. Morgan Kaufmann, 1992.

[7] S. Benferhat, D. Dubois, and H. Prade. *Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study. Part 2 : the prioritized case*, volume 24, pages 473–511. Physica-Verlag, Heidelberg, 1998.

[8] L.E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[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] M. Bienvenu, C. Bourgaux, and F. Goasdoué. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *AAAI*, pages 996–1002, 2014.

[11] M. Bienvenu and R. Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2013.

[12] D. Calvanese, G. De 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 *International Semantic Web Conference (1)*, pages 112–128, 2010.

[14] J. Chomicki. Consistent query answering: Five easy pieces. In *Database Theory - ICDT 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2007.

[15] J. Du, G. Qi, and Y. Shen. Weight-based consistent query answering over inconsistent SHIQ knowledge bases. *Knowledge and Information Systems*, 34(2):335–371, 2013.

[16] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. Fabio Savo. Inconsistency-tolerant semantics for description logics. In Pascal Hitzler and Thomas Lukasiewicz, editors, *RR*, volume 6333 of *LNCS*, pages 103–117. Springer, 2010.

[17] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics*, 33:3–29, 2015.

[18] M. Lenzerini. Ontology-based data management. In *Proceedings of the 6th Alberto Mendelzon International Workshop on Foundations of Data Management 2012*, pages 12–15, 2012.

[19] T. Lukasiewicz, M. Vanina Martinez, A. Pieris, and Gerardo I. Simari. From classical to consistent query answering under existential rules. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence 2015*, pages 1546–1552, 2015.

[20] T. Lukasiewicz, M. Vanina Martinez, and Gerardo I. Simari. Inconsistency handling in datalog+/- ontologies. In *20th European Conference on Artificial Intelligence ECAI, 2012*, pages 558–563, 2012.

[21] M. V. Martinez, F. Parisi, A. Pugliese, G. I. Simari, and V. S. Subrahmanian. Inconsistency management policies. In *Knowledge representation and reasoning*, pages 367–377. AAAI Press, 2008.

[22] R. Nicholas and M. Ruth. On inference from inconsistent premisses. *Theory and Decision*, 1(2):179–217, 1970.

[23] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.

[24] S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Annals of Mathematics and Artificial Intelligence*, 64(2-3):209–246, 2012.