Handling inconsistency in partially preordered ontologies: the Elect method

Sihem Belabbes, Salem Benferhat, Jan Chomicki

Handling Inconsistency in Partially Preordered Ontologies: The Elect Method

Sihem Belabbes[1], Salem Benferhat[2], and Jan Chomicki[3]

[1]*LIASD, IUT de Montreuil, Université Paris 8, France*
[2]*CRIL Université d'Artois & CNRS, Lens, France*
[3]*SUNY at Buffalo, Buffalo, NY, USA*

Abstract

We focus on the problem of handling inconsistency in light-weight ontologies. We assume that the terminological knowledge base (TBox) is specified in DL-Lite and that the set of assertional facts (ABox) is partially preordered and may be inconsistent with respect to the TBox. One of the main contributions of this paper is the provision of an efficient and safe method, called Elect, to restore the consistency of the ABox with respect to the TBox. In the case where the assertional base is flat (i.e., no priorities are associated with the ABox) or totally preordered, we show that our method collapses with the well-known IAR semantics (Intersection ABox Repair) and the non-defeated semantics, respectively. The semantic justification of the Elect method is obtained by first viewing a partially preordered ABox as a family of totally preordered ABoxes, and then applying non-defeated inference to each of the totally preordered ABoxes. We introduce the notion of elected assertions which allows us to provide an equivalent characterization of the Elect method without explicitly generating all the totally preordered ABoxes. We show that computing the set of elected assertions is done in polynomial time with respect to the size of the ABox. The second part of the paper discusses how to go beyond the Elect method. In particular, we discuss to what extent the Elect method can be generalized to Description Logics that are more expressive than DL-Lite.

1 Introduction

In this paper, we are interested in handling inconsistencies arising in ontologies that are specified in DL-Lite [32], a family of lightweight fragments of Description Logics (DLs) with good computational properties. In the context of Description Logics, a knowledge base (KB) consists of two components, namely the TBox which contains the terminological knowledge, and the ABox which is an assertional base (i.e., a set of ground facts). The content of the TBox is oftentimes considered as correct and free of conflicts. Here we adopt such reasonable assumption, therefore we assume that the elements of the TBox are not questionable in the presence of conflicts. However, the assertions in the ABox may be questionable when the whole KB is inconsistent.

The problem of inconsistency management has received considerable attention in the literature and has been studied in the context of propositional logic [1,15,16,29,48], databases [20,34,39,57,67] alongside Description Logics.

In the context of DL frameworks, anything can be derived from an inconsistent KB, which trivializes the task of query answering. To circumvent this issue, various inconsistency-tolerant semantics have been proposed to allow for meaningful reasoning with inconsistent KBs [9,26–28,31,55,71,72]. For instance, the ABox Repair (AR), the Intersection ABox Repair (IAR), the Closed ABox Repair (CAR) and the Intersection Closed ABox Repair (ICAR) semantics [51] are the best known inconsistency-tolerant semantics. They are based on the notion of a repair, defined as a maximal subset (in terms of set inclusion) of the ABox that is consistent with the TBox.

The ABox Repair (AR) semantics amounts to repairing the ABox in a minimal way (in terms of set inclusion) without modifying the TBox. It produces several repairs for the ABox, and queries are evaluated separately on each of the repairs before intersecting the sets of answers. In other words, a query answer is considered as valid if it follows from each repair of the ABox. The AR semantics is often viewed as a safe and appropriate way for dealing with conflicts. The Closed ABox Repair (CAR) semantics proceeds similarly but it starts by performing a deductive closure on the initial ABox before computing the repairs.

The IAR semantics is more cautious than the AR semantics. It evaluates queries over one consistent sub-base of the ABox obtained from the intersection of all the repairs. On the other hand, the ICAR semantics involves a closure operation on the initial ABox before computing the repairs and intersecting them. Both IAR and ICAR are tractable, whereas AR and CAR are computationally expensive, even for lightweight ontology logics such as DL-Lite. It is also known that AR and CAR are more productive, IAR is more cautious and ICAR may return undesirable results (i.e., obtained from questionable assertions).

Another line of research has focused on defining inconsistency-tolerant semantics when a priority relation is applied to ABox assertions. This caters for situations where information is obtained from various sources with different reliability levels. For instance, the so-called preferred repair semantics [23] introduces variants of the AR and IAR semantics by applying a total preorder to ABox assertions. It identifies different types of preferred repairs based on: set cardinality, partitioning the ABox according to priority levels, and assigning weights to the assertions. Similarly to the AR and CAR semantics, it produces several repairs for any given ABox but at a high computational cost, which makes query answering computationally hard.

The so-called non-defeated semantics [13] has also been proposed for ABoxes that are prioritized (i.e., partitioned into strata) by way of a total preorder. The idea is to iteratively apply the IAR semantics to a cumulative sequence of strata of the ABox. Similarly to the IAR and ICAR semantics, it produces a single consistent sub-base for any given ABox. Furthermore, it has been shown that the non-defeated semantics generalizes the IAR semantics when the priority relation is flat, and that its computation is polynomial in the context of DL-Lite [13,68].

In this work, we address the problem of the tractable computation of a single consistent sub-base for an inconsistent DL-Lite knowledge base, with a focus on the case where the priority relation over ABox assertions is a partial preorder. Namely, some statements are deemed as more reliable than others and there are statements with incomparable reliability levels. We provide an efficient and safe method, called Elect, to restore the consistency of the ABox with respect to the TBox [12]. We show that Elect generalizes both the IAR semantics and the non-defeated semantics. This is achieved in the case where the ABox is flat (i.e., no priorities are associated with the assertions) for the former, and when the ABox is totally preordered for the latter.

The semantic justification of Elect is obtained by first viewing the partial preorder over the ABox as a family of total preorders, then applying non-defeated inference to each of the totally preordered ABoxes, and lastly computing their intersection to produce a single consistent sub-base. The Elect method is safe since there is no arbitrary choice between the total preorders, thus all of them are taken into account for defining Elect by means of the intersection operator. We introduce the notion of elected assertions which intuitively correspond to those assertions that are strictly preferred to all their opponents (in terms of conflict). This allows us to provide an equivalent characterization of Elect, thus a single consistent sub-base is obtained without explicitly computing all the total preorders. Finally we show that the computation of Elect is done in polynomial time with respect to the size of the ABox. Therefore Elect maintains the tractability of both the IAR semantics and the non-defeated semantics while scaling the results to the more general case of partially preordered ABoxes.

The second part of the paper briefly shows how to go beyond Elect in three different ways. First, instead of considering the non-defeated semantics as underlying the definition of Elect, we consider a preferred-repair semantics based on prioritized set inclusion proposed in [23] (in the spirit of early work on prioritized propositional logic [29,64]). We call this method $\text{Partial}_{\text{PR}}$ (where the subscript 'PR' stands for Preferred Repair) [11]. We show that although $\text{Partial}_{\text{PR}}$ is more productive than Elect (i.e., it produces a larger base), its computational complexity is no longer polynomial.

Second, we go beyond DL-Lite and consider more expressive DLs [11]. The main difference in this case is that conflicts between the assertions are not necessarily binary as it is the case in DL-Lite [31]. We adapt the definitions of the Elect method and discuss the repercussions on computational complexity.

Third, we introduce a method for computing a larger but safe base, in the context of DL-Lite. We call this new method CElect and define it over the intersection of the deductive closure of the bases computed with the non-defeated semantics. We show that a base computed with Elect is indeed included in the base obtained by the CElect method.

This paper is structured as follows. Section 2 contains preliminaries on DL-Lite. Section 3 presents the IAR semantics for non-prioritized ABoxes. Section 4 discusses the non-defeated semantics for ABoxes prioritized with a total preorder. Section 5 introduces our method Elect for partially preordered ABoxes and Section 6 provides a characterization for it. In Sections 7 to 9, we introduce the methods $\text{Partial}_{\text{PR}}$ and CElect and the adaptation of Elect to DLs in general. In Section 10 we provide an overview of related work. We then conclude and discuss future work.

2 The Description Logic DL-Lite

Description Logics (DLs) [7] are a family of decidable knowledge representation languages, based on first-order predicate logic, and meeting many applications, notably in the formalisation of ontologies. The so-called lightweight fragments of DLs, of which

DL-Lite [32] is an example, are particularly interesting since they provide a good trade-off between expressive power and (data) computational complexity. In fact, DL-Lite languages have been designed to capture conceptual modeling constructs in applications with large datasets and where query answering is the most important reasoning task. Indeed, query answering from a DL-Lite knowledge base can be carried out efficiently, by reducing the problem to standard database query evaluation using query rewriting [50].

There are several variants of DL-Lite [6] such as DL-Lite$_{core}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$, DL-Lite$_{\mathcal{A}}$ and DL-Lite$_{Horn}$. In the rest of this paper, we shall focus on DL-Lite$_{\mathcal{R}}$, the dialect that provides the logical underpinning for the OWL 2 QL profile [58] which is devoted to query answering.

The DL-Lite$_{\mathcal{R}}$ language is built upon a vocabulary consisting of a set C of *concept names*, a set R of *role names* and a set I of *individual names*, where the sets C, R and I are countably infinite and pairwise disjoint. Let $A \in \mathsf{C}$, $P \in \mathsf{R}$, and $P^- \in \mathsf{R}$ is the *inverse* of P. Let the negation symbol "$\neg$" encode complement sets (of concepts and of roles), and the existential quantifier symbol "$\exists$" denote existential restriction. The DL-Lite$_{\mathcal{R}}$ language is defined according to the following rules:

$$R \longrightarrow P \mid P^- \qquad E \longrightarrow R \mid \neg R$$

$$B \longrightarrow A \mid \exists R \qquad C \longrightarrow B \mid \neg B$$

where R denotes a *basic role*, while E stands for a *complex role*. Moreover, B denotes a *basic concept* and C is a *complex concept*.

Next, we describe a domain related to dances in order to introduce the running example that we shall use throughout the paper.

Example 1. *Consider the following sets of concept, role and individual names:*

- $\mathsf{C} = \{\mathsf{Dance}, \mathsf{MDance}, \mathsf{TDance}, \mathsf{FDance}, \mathsf{WProp}, \mathsf{WoProp}, \mathsf{Prop}\}$*, standing for the following basic concepts: dance, modern dance, traditional dance, festival dance, dance with props, dance without props, as well as the props that may be used in some dances, respectively.*
- $\mathsf{R} = \{\mathsf{hasProp}, \mathsf{hasInst}\}$*, contains two basic roles. The first one links dances to the props that may be used during the performance. The second one links dances to the instruments.*
- $\mathsf{I} = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, u, h, r, m, c\}$*. Each d_i ($i = 1, \dots, 7$) represents a dance. The elements 'u', 'h', 'r' stand for umbrella, hat, ribbon, respectively, and represent the props. The elements 'm' and 'c' stand for drums and cymbals, and represent the instruments used during the performance.*

Some examples of complex concepts are: $\neg\mathsf{WProp}$ *and* $\neg\exists\mathsf{hasInst}$.
Examples of complex roles are: $\neg\mathsf{hasProp}$ *and* $\neg\mathsf{hasInst}^-$.

□

The semantics of DL-Lite$_{\mathcal{R}}$ is the standard set-theoretic semantics. An *interpretation* is a structure $\mathcal{I} =_{\text{def}} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain*, and $\cdot^{\mathcal{I}}$ is an *interpretation function* mapping concept names A to subsets $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, role names P to binary relations $P^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$, and individual names a to elements of the

domain $\Delta^{\mathcal{I}}$. Therefore, $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
We extend the interpretation function $\cdot^{\mathcal{I}}$ to interpret complex concepts and complex roles of DL-Lite$_{\mathcal{R}}$ as follows:

$$
\begin{aligned}
&(P^{-})^{\mathcal{I}} =_{\text{def}} \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in P^{\mathcal{I}}\};\\
&(\exists R)^{\mathcal{I}} =_{\text{def}} \{x \in \Delta^{\mathcal{I}} \mid \text{there is } y \in \Delta^{\mathcal{I}} \text{ s.t. } (x,y) \in R^{\mathcal{I}}\};\\
&(\neg B)^{\mathcal{I}} =_{\text{def}} \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}};\\
&(\neg R)^{\mathcal{I}} =_{\text{def}} (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}}.
\end{aligned}
$$

An *inclusion axiom* on concepts (resp. on roles) is a statement of the form $B \sqsubseteq C$ (resp. $R \sqsubseteq E$). A concept inclusion of the form $\exists P \sqsubseteq B$ (resp. $\exists P^{-} \sqsubseteq B$), which requires that the domain (resp. the range) of a role P be included in a concept B, is a "domain restriction" (resp. "range restriction"). Inclusions on concepts (resp. on roles) of the type $B_1 \sqsubseteq B_2$ (resp. $R_1 \sqsubseteq R_2$) are called "*positive inclusion axioms*". Inclusions on concepts (resp. on roles) with the negation symbol "$\neg$" in the right-hand side of the inclusion, such as $B_1 \sqsubseteq \neg B_2$ (resp. $R_1 \sqsubseteq \neg R_2$), are called "*negative inclusion axioms*". In DL-Lite$_{\mathcal{R}}$, negative axioms allow to specify the disjointness of at most two concepts (resp. two roles) [31].
Examples of negative and positive inclusion axioms on concepts are:

$$
\begin{array}{lcl}
\mathsf{WoProp} & \sqsubseteq & \neg\mathsf{WProp}\\
\exists\mathsf{hasProp}^{-} & \sqsubseteq & \mathsf{Prop}
\end{array}
$$

An example of a negative inclusion axiom on roles is:

$$\mathsf{hasInst} \sqsubseteq \neg\mathsf{hasProp}$$

A DL-Lite$_{\mathcal{R}}$ *TBox* $\mathcal{T}$ is a finite set of positive and negative inclusion axioms on concepts and on roles.

An *assertion* is a statement of the form $A(a)$ or $P(a,b)$, with $a, b \in \mathsf{I}$.
Examples of assertions are:

$$
\begin{array}{l}
\mathsf{MDance}(d_1)\\
\mathsf{hasProp}(d_3, h)
\end{array}
$$

A DL-Lite$_{\mathcal{R}}$ *ABox* $\mathcal{A}$ is a finite set of assertions.

A DL-Lite$_{\mathcal{R}}$ *knowledge base* (KB) $\mathcal{K}$ is composed of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. It is denoted as a tuple $\mathcal{K} =_{\text{def}} \langle \mathcal{T}, \mathcal{A} \rangle$.

Example 1. *(continued) Assume that we have the following TBox:*

$$
\mathcal{T} = \left\{
\begin{array}{llllll}
1.\ \mathsf{MDance} & \sqsubseteq & \mathsf{Dance} & 2.\ \mathsf{TDance} & \sqsubseteq & \mathsf{Dance}\\
3.\ \mathsf{TDance} & \sqsubseteq & \mathsf{WProp} & 4.\ \mathsf{MDance} & \sqsubseteq & \mathsf{WoProp}\\
5.\ \mathsf{WoProp} & \sqsubseteq & \neg\mathsf{WProp} & 6.\ \mathsf{WoProp} & \sqsubseteq & \neg\exists\mathsf{hasProp}\\
7.\ \exists\mathsf{hasProp} & \sqsubseteq & \mathsf{WProp} & 8.\ \exists\mathsf{hasProp}^{-} & \sqsubseteq & \mathsf{Prop}
\end{array}
\right\}
$$

The first two axioms express that modern dances and traditional dances are dances. Axiom 3 (resp. 4) means that traditional (resp. modern) dances are dances with (resp. without) props. Axiom 5 indicates that the set of dances without props and the set of dances with props are disjoint. Axiom 6 states that a dance without props does not have props. Axiom 7 requires that anything having props should be a dance with props. Axiom 8 requires that something used by an element that has props should indeed be a prop. Axioms 5 and 6 are negative inclusion axioms on concepts.

Let us now describe the ABox given by the following assertions:

$$\mathcal{A} = \left\{ \begin{array}{l} \mathsf{MDance}(d_1), \mathsf{MDance}(d_2), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \\ \mathsf{TDance}(d_4), \mathsf{WProp}(d_3), \mathsf{WProp}(d_5), \mathsf{WoProp}(d_5), \\ \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r) \end{array} \right\}$$

□

An interpretation $\mathcal{I}$ *satisfies* an inclusion axiom $B \sqsubseteq C$ (resp. $R \sqsubseteq E$), denoted by $\mathcal{I} \Vdash B \sqsubseteq C$ (resp. $\mathcal{I} \Vdash R \sqsubseteq E$), if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (resp. $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$). $\mathcal{I}$ *satisfies* an assertion $A(a)$ (resp. $P(a, b)$), denoted by $\mathcal{I} \Vdash A(a)$ (resp. $\mathcal{I} \Vdash P(a, b)$), if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$).

We say that an interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ (resp. an ABox $\mathcal{A}$), denoted by $\mathcal{I} \Vdash \mathcal{T}$ (resp. $\mathcal{I} \Vdash \mathcal{A}$), if $\mathcal{I} \Vdash \alpha$ for every α in $\mathcal{T}$ (resp. in $\mathcal{A}$). We say that $\mathcal{I}$ is a model of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if $\mathcal{I} \Vdash \mathcal{T}$ and $\mathcal{I} \Vdash \mathcal{A}$.

A knowledge base $\mathcal{K}$ is said to be *consistent* if it admits at least one model. It is *inconsistent* otherwise.
A TBox $\mathcal{T}$ is *incoherent* if there is $A \in \mathsf{C}$ such that for each interpretation $\mathcal{I}$ which is a model of $\mathcal{T}$, $A^{\mathcal{I}} = \emptyset$. It is *coherent* otherwise.

Positive and negative inclusion axioms of a coherent TBox do not play the same role in the corresponding KB. On the one hand, it is obvious that a KB with only positive axioms in the TBox is always consistent. This means that positive axioms can be used to enlarge the ABox by deriving new assertions from the original ones. On the other hand, the inconsistency of the KB occurs when subsets of the ABox violate some negative axioms. In other words, negative axioms can be used to check the consistency of the ABox w.r.t. the TBox. In this work, we focus on the case of a coherent TBox and assume there is no singleton ABox that violates negative axioms. Hence the disjointness property of DL-Lite$_{\mathcal{R}}$ applies to subsets of the ABox which contain pairs of concepts or pairs of roles.

For instance, note that the TBox given in Example 1 is coherent whereas the KB is inconsistent, i.e., the ABox is inconsistent w.r.t. the TBox. For instance, Axiom 5, namely $\mathsf{WoProp} \sqsubseteq \neg\mathsf{WProp}$, requires the concepts WoProp and WProp to be disjoint. However, the individual "d_5" belongs to both concepts. The set $\{\mathsf{WProp}(d_5), \mathsf{WoProp}(d_5)\}$ violates a negative axiom, which means that the two assertions cannot both hold.

For more details on the DL-Lite family of DLs, we refer the reader to the work of Calvanese et al. [32].

3 IAR Semantics for Flat Assertional Bases

In the rest of this paper, we consider a KB $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$ that may be inconsistent. We assume that the TBox $\mathcal{T}$ is coherent and reliable (i.e., validated by the ontology's designers). Therefore, the axioms in $\mathcal{T}$ are not questionable in the presence of conflicts, unlike the assertions in $\mathcal{A}$ which may be questionable. Moreover, in this section we assume that the ABox $\mathcal{A}$ is flat (or non-prioritized), that is, all assertions have the same priority level. A standard way for dealing with inconsistency proceeds by first computing the set of maximal subsets of $\mathcal{A}$ that are consistent with $\mathcal{T}$, called maximal repairs, then using them to perform inference (i.e., query answering). More formally, a maximal repair[1] is defined as follows [51]:

Definition 1. *Let $\mathcal{K}{=}\langle\mathcal{T},\mathcal{A}\rangle$ be a flat DL-Lite$_\mathcal{R}$ KB. A sub-base $\mathcal{R}\subseteq\mathcal{A}$ is a* maximal repair *if $\langle\mathcal{T},\mathcal{R}\rangle$ is consistent, and for every $\mathcal{R}'\subseteq\mathcal{A}$, $\mathcal{R}\subsetneq\mathcal{R}'$, $\langle\mathcal{T},\mathcal{R}'\rangle$ is inconsistent. Hence, if $\mathcal{K}$ is consistent, there is a single maximal repair $\mathcal{R}=\mathcal{A}$.*

Consequently, when $\mathcal{K}$ is inconsistent, given a maximal repair $\mathcal{R}$, adding any assertion f from $\mathcal{A}\setminus\mathcal{R}$ to $\mathcal{R}$ entails the inconsistency of $\langle\mathcal{T},\mathcal{R}\cup\{f\}\rangle$. We denote by $\mathsf{MAR}(\mathcal{A})$ (Maximal ABox Repair) the set of maximal repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$.

Using the notion of maximal repairs, handling inconsistency from a flat DL-Lite$_\mathcal{R}$ KB can be done by applying standard query answering, using either (i) the whole set of maximal repairs (e.g. universal entailment or AR entailment [51]), (ii) some maximal repairs (e.g. cardinality-based repairs) (iii) or only one maximal repair (e.g. brave entailment [24]). It is well known that the brave semantics is very adventurous and may return unsafe conclusions, while the AR and cardinality-based repair semantics are safe but computationally expensive.

An alternative is the IAR semantics [51] which selects one consistent sub-base of $\mathcal{A}$, denoted by $\mathsf{IAR}(\mathcal{A})$. Before introducing the IAR semantics, let us first introduce the notion of an assertional conflict. Basically, it is a minimal inconsistent subset of assertions that contradicts the TBox.

Definition 2. *Let $\mathcal{K}{=}\langle\mathcal{T},\mathcal{A}\rangle$ be a flat DL-Lite$_\mathcal{R}$ KB. A sub-base $\mathcal{C}\subseteq\mathcal{A}$ is said to be an* assertional conflict *of $\mathcal{K}$ if $\langle\mathcal{T},\mathcal{C}\rangle$ is inconsistent and for every $f\in\mathcal{C}$, $\langle\mathcal{T},\mathcal{C}\setminus\{f\}\rangle$ is consistent.*

We denote by $\mathsf{Conf}(\mathcal{A})$ the set of all conflicts in $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$. From Definition 2, we see that removing any fact f from $\mathcal{C}$ restores the consistency of $\langle\mathcal{T},\mathcal{C}\rangle$. A nice feature of DL-Lite$_\mathcal{R}$ is that computing and generating the conflict set is done in polynomial time w.r.t. the size of the ABox [28,31]. Namely, there exists an algorithm that takes as input the knowledge base $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$, and generates/produces the set of all the assertional conflicts of $\mathcal{K}$. The space and time complexity of such algorithm is polynomial w.r.t. the size of the ABox. Furthermore, any conflict involves at most two assertions [31]. As stated previously, we assume there is no single assertion $f\in\mathcal{A}$ such that $\langle\mathcal{T},\{f\}\rangle$ is inconsistent, and we also assume the TBox to be coherent. This implies that the assertional conflicts are binary. In this case, if f and g are two assertions that belong to a conflict $\mathcal{C}$, we simply denote it as a pair $\mathcal{C}=\{f,g\}$, and we say that f and g are conflicting.

Next, we introduce the notion of non-conflicting or free assertions.

Definition 3. *Let $\mathcal{K}{=}\langle\mathcal{T},\mathcal{A}\rangle$ be a flat DL-Lite$_\mathcal{R}$. An assertion $f\in\mathcal{A}$ is* free *if for every $\mathcal{C}\in\mathsf{Conf}(\mathcal{A})$, $f\notin\mathcal{C}$.*

Intuitively, free assertions correspond to the assertions that are not involved in any conflict. The notion of free elements was originally proposed in [14] in the context of propositional logic.

[1] In the literature, a repair is often defined as a maximal consistent subset of assertions. Here, we distinguish between a maximal repair and a repair which is simply a consistent subset of assertions. This is analogous to the notions of maximal consistent subtheory and consistent subtheory in propositional logic.

Henceforth, we shall denote by $\mathsf{IAR}(\mathcal{A})$ (Intersection ABox Repair) the set of free assertions in $\mathcal{A}$. Namely:

$$\mathsf{IAR}(\mathcal{A}) = \{f \mid f \in \mathcal{A} \text{ and } f \text{ is free}\},$$

which is an equivalent rewriting of the standard definition of $\mathsf{IAR}(\mathcal{A})$ [14, 51]:

$$\mathsf{IAR}(\mathcal{A}) = \bigcap\{\mathcal{R} \mid \mathcal{R} \in \mathsf{MAR}(\mathcal{A})\}.$$

In other words, the repair $\mathsf{IAR}(\mathcal{A})$ is the intersection of all the maximal repairs. Clearly, since computing the set of assertional conflicts $\mathsf{Conf}(\mathcal{A})$ is done in polynomial time w.r.t. the size of the ABox, then computing the repair $\mathsf{IAR}(\mathcal{A})$ is also done in polynomial time w.r.t. the size of the ABox $\mathcal{A}$.

Query answering in the IAR semantics comes down to performing standard query answering from $\langle\mathcal{T}, \mathsf{IAR}(\mathcal{A})\rangle$ (due to the fact that $\langle\mathcal{T}, \mathsf{IAR}(\mathcal{A})\rangle$ is consistent).

Example 2. *Let us consider again Example 1. The conflict set in* $\langle\mathcal{T}, \mathcal{A}\rangle$ *is:* $\mathsf{Conf}(\mathcal{A}) = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$*, where:*

$$\begin{aligned}\mathcal{C}_1 &= \{\mathsf{MDance}(d_2), \mathsf{TDance}(d_2)\},\\ \mathcal{C}_2 &= \{\mathsf{MDance}(d_2), \mathsf{hasProp}(d_2, u)\},\\ \mathcal{C}_3 &= \{\mathsf{WProp}(d_5), \mathsf{WoProp}(d_5)\}.\end{aligned}$$

In order to define $\mathsf{IAR}(\mathcal{A})$*, it is enough to remove all the assertions of* $\mathcal{C}_1, \mathcal{C}_2$ *and* $\mathcal{C}_3$ *from* $\mathcal{A}$*. This yields:*

$$\begin{aligned}\mathsf{IAR}(\mathcal{A}) = \{\quad & \mathsf{MDance}(d_1), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \mathsf{WProp}(d_3), \mathsf{hasProp}(d_3, h),\\ & \mathsf{hasProp}(d_4, r)\}.\end{aligned}$$

□

4 Non-defeated Repair for Prioritized Assertional Bases

In this section, we consider prioritized DL-Lite$_\mathcal{R}$ KBs wherein a total preorder relation $\geq$ is applied only to the ABox component and which we denote by $(\mathcal{A}, \geq)$. The relation $\geq$ is reflexive, transitive and satisfies:

For every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, either $f \geq g$ or $g \geq f$.

Let $>$ stand for the strict relation and $\equiv$ stand for the equivalence relation associated with $\geq$. Intuitively, $f \geq g$ means that the assertion f is at least as important as the assertion g.
Moreover, for convenience, we also represent $(\mathcal{A}, \geq)$ by a well-ordered partition of $\mathcal{A}$ induced by $\geq$. Namely, given $(\mathcal{A}, \geq)$, we view $\mathcal{A}$ as being partitioned into n non-empty layers (or strata) of the form $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$, such that:

- $\mathcal{S}_1 = \{f \mid \text{for every } g \in \mathcal{A}, f \geq g\}$, and
- $\mathcal{S}_i = \{f \mid \text{for every } g \in \mathcal{A} \setminus (\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{i-1}), f \geq g\}$, for $i = 2, \ldots, n$.

In other words, the assertions in each layer $\mathcal{S}_i$ have the same priority level i and they are considered as more reliable than the ones contained in a layer $\mathcal{S}_j$ for $j > i$. Thus $\mathcal{S}_1$ contains the most important assertions, while $\mathcal{S}_n$ contains the least important ones. Obviously, by construction, $\mathcal{A} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$.

Several studies consider the notion of priority when querying inconsistent databases (e.g. [57, 67]) or DL knowledge bases (e.g. [23, 39]). Most of these frameworks extend the notions of maximal repair and AR semantics, therefore they are computationally expensive. In particular, the concepts of preferred repairs semantics were introduced in [23] (in the spirit of what has been done in prioritized propositional logic [29, 64]). It revisits the AR and IAR semantics by replacing the notion of repair by different types of preferred repairs based on: set cardinality, priority levels on the ABox and weights on the assertions (see Section 7). However, this formalism often induces an increase in computational complexity for the proposed semantics. Most notably, the tractability of the IAR semantics in a flat context (i.e., polynomial time) is lost when a total preorder is applied to the ABox.

In [13], a particular attention was devoted to approaches that select a single preferred repair. One of such approaches is the so-called non-defeated repair which is tractable without being adventurous. Basically, the non-defeated repair consists of iteratively collecting, layer per layer, the set of free assertions like so:

Definition 4. *Let $\mathcal{K}$ be a prioritized DL-Lite$_\mathcal{R}$ KB where the ABox $(\mathcal{A}, \geq)$ is totally preordered. Let $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be the well-ordered partition associated with $\geq$. The non-defeated repair of $(\mathcal{A}, \geq)$ is given by:*

$$\begin{aligned} &\mathsf{nd}(\mathcal{A}, \geq) = \mathcal{S}'_1 \cup \ldots \cup \mathcal{S}'_n, \textit{ such that} \\ &\mathcal{S}'_i = \mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i), \ \textit{for } i = 1, \ldots, n, \end{aligned}$$

where $\mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$ denotes the free elements of the set $(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$.

The definition of the non-defeated sub-base is an adaptation of the one proposed in [16] within a propositional logic framework. However, the non-defeated repair is computed in polynomial time in DL-Lite$_\mathcal{R}$ while its computation is hard in propositional logic. Lastly, in [17] a rewriting (similar to that of $\mathsf{IAR}(\mathcal{A})$) is given for $\mathsf{nd}(\mathcal{A}, \geq)$. Basically, an assertion $f \in \mathcal{S}_i$ is said to be defeated if there is an assertion $g \in \mathcal{S}_j$ such that $j \leq i$ and g conflicts with f. It has been shown in [17] that $\mathsf{nd}(\mathcal{A}, \geq)$ consists of all the non-defeated assertions.

$$\begin{array}{c} \mathsf{TDance}(d_2) \equiv \mathsf{MDance}(d_1) \equiv \mathsf{TDance}(d_3) \equiv \mathsf{hasProp}(d_3, h) \\ \downarrow \\ \mathsf{hasProp}(d_2, u) \equiv \mathsf{WProp}(d_3) \equiv \mathsf{WProp}(d_5) \equiv \mathsf{WoProp}(d_5) \\ \downarrow \\ \mathsf{MDance}(d_2) \equiv \mathsf{TDance}(d_4) \equiv \mathsf{hasProp}(d_4, r) \end{array}$$

Figure 1: A total preorder $\geq$ over $\mathcal{A}$. A solid arrow (resp. dotted line) from an assertion f to an assertion g means that $f > g$ (resp. $\{f, g\} \in \mathsf{Conf}(\mathcal{A})$).

Example 3. *We continue our running example and assume that a total preorder $\geq$ is applied to the assertions of $\mathcal{A}$, as depicted in Figure 1. Then $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ where:*

$$\begin{aligned} \mathcal{S}_1 = &\quad \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{hasProp}(d_3, h)\}, \\ \mathcal{S}_2 = &\quad \{\mathsf{hasProp}(d_2, u), \mathsf{WProp}(d_3), \mathsf{WProp}(d_5), \mathsf{WoProp}(d_5)\}, \\ \mathcal{S}_3 = &\quad \{\mathsf{MDance}(d_2), \mathsf{TDance}(d_4), \mathsf{hasProp}(d_4, r)\}. \end{aligned}$$

The non-defeated subclass of $\mathcal{A}$ is given by:

$$\mathsf{nd}(\mathcal{A}, \geq) = \mathsf{IAR}(\mathcal{S}_1) \cup \mathsf{IAR}(\mathcal{S}_1 \cup \mathcal{S}_2) \cup \mathsf{IAR}(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3).$$

The partition $\mathcal{S}_1$ is free of conflicts, hence:

$$\mathsf{IAR}(\mathcal{S}_1) = \mathcal{S}_1 = \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{hasProp}(d_3, h)\}.$$

We remove the conflicting assertions in $(\mathcal{S}_1 \cup \mathcal{S}_2)$ and obtain:

$$\begin{aligned} \mathsf{IAR}(\mathcal{S}_1 \cup \mathcal{S}_2) &= (\mathcal{S}_1 \cup \mathcal{S}_2) \setminus \{\mathsf{WProp}(d_5), \mathsf{WoProp}(d_5)\} \\ &= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{WProp}(d_3), \\ &\quad \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h)\}. \end{aligned}$$

We do the same for $(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3)$ and obtain:

$\mathsf{IAR}(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) = \mathsf{IAR}(\mathcal{A})$ *(given in Example 2).*

It follows that the non-defeated repair of $\mathcal{A}$ is:

$$\begin{aligned} \mathsf{nd}(\mathcal{A}, \geq) = \{ \; & \mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{WProp}(d_3), \mathsf{TDance}(d_4), \\ & \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}. \end{aligned}$$

Since each $\mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$ (with $1 \leq i \leq 3$) is computed in polynomial time, then $\mathsf{nd}(\mathcal{A}, \geq)$ is also computed in polynomial time w.r.t. the size of $\mathcal{A}$.

□

5 Partially Preordered Assertional Bases

A nice feature about the IAR semantics (for a flat ABox) and the non-defeated semantics (for a totally preordered ABox) is their efficiency in dealing with inconsistency since they produce a single consistent sub-base of the ABox as a repair and they do so in polynomial time. In this section, we also aim at producing a single repair when only a partial preorder, denoted by $\trianglerighteq$, is applied to the assertions of the ABox and which we denote by $(\mathcal{A}, \trianglerighteq)$.
We denote by $\triangleright$ over $\mathcal{A}$ the strict order (irreflexive and transitive) associated with $(\mathcal{A}, \trianglerighteq)$, and which is defined by:

For every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, $f \triangleright g$ if $f \trianglerighteq g$ holds and $g \trianglerighteq f$ does not hold.

We also denote by $\triangleq$ over $\mathcal{A}$ the equivalence order associated with $(\mathcal{A}, \trianglerighteq)$, and which is defined by:

For every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, $f \triangleq g$ if $f \trianglerighteq g$ and $g \trianglerighteq f$ both hold.

When neither $f \trianglerighteq g$ nor $g \trianglerighteq f$ holds, we say that the assertions f and g are incomparable and denote it by $f \bowtie g$.

A natural minimal requirement is to maintain tractability. Namely, we seek a tractable method that also returns one (preferred) repair for a partially preordered ABox. We call our method Elect and denote by $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ the repair it returns [12]. As we shall see later, Elect extends both the IAR semantics and the non-defeated semantics in the cases where the relation $\trianglerighteq$ is flat and totally preordered respectively. Henceforth, we do not make explicit the TBox $\mathcal{T}$.

5.1 From a partial preorder to a family of total preorders

In order to achieve our aim, we first view a partial preorder $\trianglerighteq$ as a family of total preorders, each of which should be a total extension of $\trianglerighteq$ defined like so:

Definition 5. *A total preorder $\geq$ over $\mathcal{A}$ is a total extension of $\trianglerighteq$ over $\mathcal{A}$ if for every $f \in \mathcal{A}$ and for every $g \in \mathcal{A}$:*

- *if $f \rhd g$ then $f > g$, and*
- *if $f \trianglerighteq g$ then $f \geq g$.*

It is trivial that when the partial preorder $\trianglerighteq$ is a total preorder, there is only one total extension which is the relation itself. The situation differs when there are cases of incomparability between the assertions. Roughly speaking, stating that two assertions f and g are incomparable means that either f is strictly preferred to g, or g is strictly preferred to f, or f and g are equally preferred, but we do not know which case holds. (This makes sense only when the elements can effectively be compared.) Thus, extending a partial preorder consists in replacing each case where two assertions are incomparable, such as $f \bowtie g$, by the three cases where the two assertions are comparable, namely $f > g$, $g > f$ and $f \equiv g$.

Viewing a partially preordered KB as a family of totally preordered KBs is a natural representation that has been used in other frameworks such as partially ordered possibilistic logic [19, 69] and credal probabilistic networks [37].

Example 4. *We assume a partial preorder $\trianglerighteq$ over $\mathcal{A}$ such that the assertions are split up into the following four disjoint subsets:*

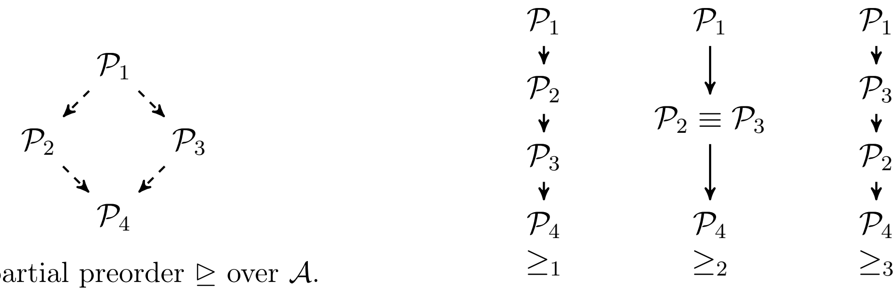
$$\begin{aligned}
\mathcal{P}_1 &= \{\mathsf{MDance}(d_1) \triangleq \mathsf{TDance}(d_2) \triangleq \mathsf{TDance}(d_3) \triangleq \mathsf{hasProp}(d_3, h)\},\\
\mathcal{P}_2 &= \{\mathsf{hasProp}(d_2, u) \triangleq \mathsf{WProp}(d_3) \triangleq \mathsf{WProp}(d_5) \triangleq \mathsf{WoProp}(d_5)\},\\
\mathcal{P}_3 &= \{\mathsf{MDance}(d_2)\},\\
\mathcal{P}_4 &= \{\mathsf{TDance}(d_4) \triangleq \mathsf{hasProp}(d_4, r)\}.
\end{aligned}$$

The relation $\trianglerighteq$ is depicted in Figure 2(a). The assertions within the same set are equally preferred. The assertions of $\mathcal{P}_1$ (resp. $\mathcal{P}_4$) are the most (resp. the least) preferred. The preference levels of the assertions of $\mathcal{P}_2$ are incomparable to those of $\mathcal{P}_3$ and vice versa. We resolve incomparability by viewing the partial preorder $\trianglerighteq$ as a family of three total preorders $\geq_1$, $\geq_2$ and $\geq_3$ which preserve the strict preferences between the assertions and such that:

- *The relation $\geq_1$ considers that the assertions of the set $\mathcal{P}_2$ are strictly preferred to those of the set $\mathcal{P}_3$. Let $(\mathcal{A}, \geq_1)$ be the resulting ABox.*
- *The relation $\geq_2$ considers that the assertions of the sets $\mathcal{P}_2$ and $\mathcal{P}_3$ are equally preferred. Let $(\mathcal{A}, \geq_2)$ be the resulting ABox.*
- *The relation $\geq_3$ considers that the assertions of the set $\mathcal{P}_3$ are strictly preferred to those of the set $\mathcal{P}_2$. Let $(\mathcal{A}, \geq_3)$ be the resulting ABox.*

This is depicted in Figure 2(b).

□



(a) A partial preorder $\trianglerighteq$ over $\mathcal{A}$.

(b) The three total extensions of $\trianglerighteq$ over $\mathcal{A}$.

Figure 2: In $\trianglerighteq$ (resp. each $\geq_k$), a dashed (resp. solid) arrow from $\mathcal{P}_i$ to $\mathcal{P}_j$ means that for every $f \in \mathcal{P}_i$ and for every $g \in \mathcal{P}_j$, $f \triangleright g$ (resp. $f >_k g$). In $\trianglerighteq$ (resp. $\geq_2$), for every $f \in \mathcal{P}_2$ and for every $g \in \mathcal{P}_3$, $f \bowtie g$ (resp. $f \equiv g$).

5.2 The Elect method

Once the cases of incomparability in the partially preordered ABox have been resolved, the question is how to handle the corresponding family of totally preordered ABoxes. We would like to avoid an arbitrary choice consisting in the selection of one total preorder over the others. Thus all the total preorders should be equally taken into account. A safe way to derive a single consistent assertional sub-base from the ABox is to consider the intersection of all the non-defeated repairs associated with all the total preorders. Formally:

Definition 6. *Let $\mathcal{K}$ be a DL-Lite$_\mathcal{R}$ KB with a partially preordered ABox $(\mathcal{A}, \trianglerighteq)$.*

- $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq} \{\mathsf{nd}(\mathcal{A}, \geq) \mid \geq$ *is a total extension of* $\trianglerighteq\}$, *where* $\mathsf{nd}(\mathcal{A}, \geq)$ *is given by Definition 4.*
- *Let q be a query. Then q is an* Elect-consequence *of $\mathcal{K}$ if q follows from* $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ *(using standard DL-Lite inference).*

We illustrate Definition 6 using our running example.

Example 5. *Using Definition 4, one can check that the non-defeated repairs associated with the totally preordered ABoxes $(\mathcal{A}, \geq_1)$, $(\mathcal{A}, \geq_2)$ and $(\mathcal{A}, \geq_3)$ obtained in Example 4 are:*

- $\mathsf{nd}(\mathcal{A}, \geq_1) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3), \mathsf{hasProp}(d_2, u)\} \cup \mathcal{P}_4$
- $\mathsf{nd}(\mathcal{A}, \geq_2) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4$
- $\mathsf{nd}(\mathcal{A}, \geq_3) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4$

Therefore, using Definition 6, we obtain:

$$\begin{aligned}
\mathsf{Elect}(\mathcal{A}, \trianglerighteq) &= \mathsf{nd}(\mathcal{A}, \geq_1) \cap \mathsf{nd}(\mathcal{A}, \geq_2) \cap \mathsf{nd}(\mathcal{A}, \geq_3) \\
&= \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4 \\
&= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \mathsf{WProp}(d_3), \\
&\quad\ \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}.
\end{aligned}$$

$\square$

An important result stated in Proposition 1 is that the computation of the repair $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ can be achieved in polynomial time. In fact, we shall show later that in order to compute $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$, there is no need to exhibit all the possible extensions of the partial preorder $\trianglerighteq$.

Proposition 1. *Computing $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ is done in polynomial time (w.r.t. the size of the ABox).*

The next proposition states that, as expected, the KB where the ABox component is the repair $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ is consistent.

Proposition 2. *$\langle \mathcal{T}, \mathsf{Elect}(\mathcal{A}, \trianglerighteq) \rangle$ is consistent.*

Another interesting feature of Elect is that it collapses with the IAR semantics (resp. non-defeated semantics) when the ABox is flat (resp. totally preordered).

Proposition 3. *If the partial preorder $\trianglerighteq$ is flat, then $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \mathsf{IAR}(\mathcal{A})$. If the partial preorder $\trianglerighteq$ is a total preorder, then $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \mathsf{nd}(\mathcal{A}, \trianglerighteq)$.*

The proofs of Propositions 1 to 3 are established by providing a characterization of $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ and which is presented in the next section.

6 Characterizing $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$

In this section, we provide a characterization of $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ without having to compute all total extensions of $\trianglerighteq$. This is done by introducing the notion of elected assertions. Intuitively, an assertion f is said to be elected in $(\mathcal{A}, \trianglerighteq)$ if it is strictly preferred to all the assertions that conflict with it. Formally:

Definition 7. *An assertion $f \in \mathcal{A}$ is said to be elected if for every $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ where $\mathcal{C} = \{f, g\}$, $f \neq g$, then $f \rhd g$ (i.e., f is strictly preferred to g).*

Note that if an assertion f is free in $\mathcal{A}$ then f is elected, since free assertions are not involved in any conflict. However the converse is false. A simple counterexample is to consider the TBox with one negative axiom $\{B_1 \sqsubseteq \neg B_2\}$ and the ABox $\{B_1(a), B_2(a)\}$ with the strict preference $B_1(a) \rhd B_2(a)$. The assertion $B_1(a)$ is elected but it is not free.

In fact, Definition 7 extends the notion of free assertions given in Definition 3. Indeed, when the relation $\trianglerighteq$ is flat, namely :

$$\text{For every } f \in \mathcal{A}, \text{ for every } g \in \mathcal{A}, f \trianglerighteq g \text{ and } g \trianglerighteq f,$$

then no assertion in $\mathcal{A}$ is strictly preferred to another assertion (i.e., for every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, neither $f \rhd g$ nor $g \rhd f$ holds). Then one can easily check that f is elected in $(\mathcal{A}, \trianglerighteq)$ if and only if f is free. The converse does not hold when the relation $\trianglerighteq$ is no longer flat. This is due to the fact that an elected assertion may not be a free assertion, however its reliability is strictly more important than that of its opponents, as illustrated by the above counterexample.

Definition 7 also extends the notion of non-defeated assertions given for non-defeated repairs in totally preordered KBs [17]. Lastly, the notion of elected assertions is in the spirit of the notion of accepted beliefs introduced in uncertainty theories [40]. Note that from Definition 7, it is trivial to see that if $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent, then $\mathsf{IAR}(\mathcal{A}) = \mathcal{A}$ and all the assertions in $\mathcal{A}$ are elected.

As shown in Proposition 4, it turns out that the set of elected assertions matches exactly the set of assertions in $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$.

Proposition 4. *An assertion $f \in \mathcal{A}$ is elected in $(\mathcal{A}, \trianglerighteq)$ (using Definition 7) if and only if $f \in \mathsf{Elect}(\mathcal{A}, \trianglerighteq)$, where $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ is given by Definition 6.*

Proof. Let $(\mathcal{A}, \trianglerighteq)$ be a partially preordered assertional base.

i) Let $f \in \mathcal{A}$ be an elected assertion. Let us show that for each total extension $(\mathcal{A}, \geq)$ of $(\mathcal{A}, \trianglerighteq)$, we have $f \in \mathsf{nd}(\mathcal{A}, \geq)$. Let $\geq$ be a total extension of $\trianglerighteq$, and let $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be the well-ordered partition associated with $\geq$. Let i be the first stratum where $f \in \mathcal{S}_i$.

   Recall that f is elected in $(\mathcal{A}, \trianglerighteq)$ means that for every $g \in \mathcal{A}$, if $\{f, g\}$ are conflicting then $f \triangleright g$ (i.e., f is strictly preferred to g w.r.t. $\trianglerighteq$). And since $\geq$ is a total extension of $\trianglerighteq$, then this also means that $f > g$. This also means that for every $g \in \mathcal{A}$ such that $\{f, g\}$ are conflicting, $g \in \mathcal{S}_j$ with $j > i$. Hence, $f \in \mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$. Therefore $f \in \mathsf{nd}(\mathcal{A}, \geq)$.

ii) Let us now show the converse. Assume that $f \in \mathcal{A}$ is not elected and let us build a total extension $(\mathcal{A}, \geq)$ of $(\mathcal{A}, \trianglerighteq)$ such that $f \notin \mathsf{nd}(\mathcal{A}, \geq)$.

   The assertion f is not elected means that there is $g \in \mathcal{A}$ such that $\{f, g\}$ are conflicting but $f \triangleright g$ does not hold. This means that there is a total extension $\geq$ of $\trianglerighteq$ where $g \geq f$. If $\{f, g\}$ are conflicting and $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ is the well-ordered partition associated with $\geq$, then if $f \in \mathcal{S}_i$ it follows that $g \in \mathcal{S}_j$ with $j \leq i$. Hence, for $k = 1, \ldots, n$, $f \notin \mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k)$ which means that $f \notin \mathsf{nd}(\mathcal{A}, \geq)$.

$\square$

$\square$

Algorithm: ComputeElected

Input: a TBox $\mathcal{T}$, a partially preordered ABox $(\mathcal{A}, \trianglerighteq)$.
Output: a set of elected assertions $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$.

Compute the conflict set $\mathsf{Conf}(\mathcal{A})$
NotElected $\leftarrow \emptyset$
foreach $\{f, g\}$ **in** $\mathsf{Conf}(\mathcal{A})$ **do**
    if $(f \bowtie g)$ **or** $(f \triangleq g)$ **then**
        NotElected $\leftarrow$ NotElected $\cup \{f, g\}$
    else
        if $(f \triangleright g)$ **then**
            NotElected $\leftarrow$ NotElected $\cup \{g\}$
        else
            NotElected $\leftarrow$ NotElected $\cup \{f\}$
return $(\mathcal{A} \setminus$ NotElected$)$

Thanks to Proposition 4, we can prove Propositions 1, 2 and 3.

Proof.

Proof of Proposition 1: Regarding the computational complexity, we first provide the basic algorithm $\mathsf{ComputeElected}$ which computes the set of elected assertions according to the characterization of Proposition 4. We recall that computing the conflict set $\mathsf{Conf}(\mathcal{A})$ is done in polynomial time w.r.t. the size of $\mathcal{A}$ (e.g. [28]). This corresponds to Step 1 of our algorithm. Steps 3–10 concern the computation of $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ which is also done in polynomial time. These steps state that checking if some assertion $f \in \mathcal{A}$ is elected boils down to parsing all the assertional conflicts (Step 3) in $\mathsf{Conf}(\mathcal{A})$. This is done in linear time w.r.t. the size of $\mathsf{Conf}(\mathcal{A})$ (the size is itself bounded by $\mathcal{O}(|\mathcal{A}|^2)$).

Proof of Proposition 2: Let us show that $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ is consistent w.r.t. $\mathcal{T}$. Assume that this is not the case. This means that there is $f \in \mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ and there is $g \in \mathsf{Elect}(\mathcal{A}, \trianglerighteq), g \neq f$, such that $\langle \mathcal{T}, \{f, g\} \rangle$ are conflicting. Since f and g are in $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$, then using Definition 7, this means that $f \triangleright g$ and $g \triangleright f$ which is impossible.

Proof of Proposition 3: Lastly, by construction of $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$, it is easy to check that when $\trianglerighteq$ is a total preorder, then $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ collapses with the non-defeated repair of $\trianglerighteq$. And if $\trianglerighteq$ is flat (namely for every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, $f \trianglerighteq g$ and $g \trianglerighteq f$), then $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \mathsf{IAR}(\mathcal{A}) = \{f \in \mathcal{A} \mid \text{there is no } g \in \mathcal{A} \text{ s.t. } \{f, g\} \text{ are conflicting}\}$.

□

□

We now apply the algorithm $\mathsf{ComputeElected}$ to our running example and obtain the same set of elected assertions as in Example 5.

Example 6. *We recall that* $\mathsf{Conf}(\mathcal{A})$ *contains the following three conflicts:*

- $\mathcal{C}_1 = \{\mathsf{MDance}(d_2), \mathsf{TDance}(d_2)\}$. *Since* $\mathsf{TDance}(d_2) \triangleright \mathsf{MDance}(d_2)$*, the assertion* $\mathsf{MDance}(d_2)$ *is not elected.*
- $\mathcal{C}_2 = \{\mathsf{MDance}(d_2), \mathsf{hasProp}(d_2, u)\}$. *Since* $\mathsf{MDance}(d_2) \bowtie \mathsf{hasProp}(d_2, u)$*, the assertion* $\mathsf{hasProp}(d_2, u)$ *is also not elected.*
- $\mathcal{C}_3 = \{\mathsf{WProp}(d_5), \mathsf{WoProp}(d_5)\}$. *Since* $\mathsf{WProp}(d_5) \triangleq \mathsf{WoProp}(d_5)$*, neither of the two assertions is elected.*

All the remaining assertions are elected. Namely:

$$\begin{aligned}\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \{ \quad & \mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \mathsf{WProp}(d_3), \\ & \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}.\end{aligned}$$

We obtained the same result when we considered all the total extensions of $\trianglerighteq$.

□

In the rest of this paper, we briefly investigate three possible ways to go beyond the Elect method while still maintaining the safety of the results. First, we redefine Elect in terms of a preferred semantics from the literature. Second, we go beyond DL-Lite$_{\mathcal{R}}$ and adapt Elect to more expressive DLs. Third, we introduce a method for producing larger repairs.

7 A Preference-Based Semantics for Elect

The non-defeated semantics [13] underlies the definition of the Elect method when a partial preorder $\trianglerighteq$ is applied to an ABox $\mathcal{A}$. Indeed, the partial preorder is seen as a family of total preorders for which non-defeated repairs can be computed. Then a single consistent sub-base (denoted by $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$) is obtained from the intersection of the non-defeated repairs. The question addressed in this section is whether using a strategy that is more productive than the non-defeated semantics yields a single consistent sub-base that is larger than $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$. A candidate strategy is the preference-based semantics introduced in [23] and which can be used as a basis to redefine the Elect method. Let us first recall the notion of preferred repairs defined for totally preordered ABoxes.

Definition 8. *Let $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be a prioritized ABox. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two consistent sub-bases of $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$.*

- *$\mathcal{R}_1$ is* equally *preferred to $\mathcal{R}_2$ if $(\mathcal{R}_1 \cap \mathcal{S}_i = \mathcal{R}_2 \cap \mathcal{S}_i)$ for $i = 1, \ldots, n$.*
- *$\mathcal{R}_1$ is* strictly *preferred to $\mathcal{R}_2$ if $(\mathcal{R}_2 \cap \mathcal{S}_i \subsetneq \mathcal{R}_1 \cap \mathcal{S}_i)$ for some $i \in \{1, \ldots, n\}$, and $(\mathcal{R}_1 \cap \mathcal{S}_j = \mathcal{R}_2 \cap \mathcal{S}_j)$ for all $j, 1 \leq j < i$.*

Then $\mathcal{R} \subseteq \mathcal{A}$ is a preferred repair of $\mathcal{A}$ if there is no $\mathcal{R}' \subseteq \mathcal{A}$ s.t. $\mathcal{R}'$ is strictly preferred to $\mathcal{R}$.

The notion of preferred repairs has been first introduced in the context of propositional logic [29] (see also [16]).

Example 7. *Let us consider the partial preorder $\trianglerighteq$ of Example 4 and one of its total extensions, namely $\geq_1$. Recall that the stratification associated with the relation $\geq_1$ is: $\mathcal{A} = (\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4)$. We compute the preferred repairs of $(\mathcal{A}, \geq_1)$ in a constructive manner from the strata $\mathcal{P}_1$ to $\mathcal{P}_4$ as follows:*

- *The set $\mathcal{P}_1$ is conflict-free so all its elements can belong to the same repair.*
- *The set $\mathcal{P}_2$ contains $\mathsf{WoProp}(d_5)$ and $\mathsf{WProp}(d_5)$ which are conflicting and should not belong to the same repair. So there are two repairs at this level.*
- *The only element of the set $\mathcal{P}_3$, namely $\mathsf{MDance}(d_2)$, conflicts with $\mathsf{TDance}(d_2)$ of the set $\mathcal{P}_1$, hence it should not be included in any repair.*
- *The set $\mathcal{P}_4$ is conflict-free so all its elements can belong to the same repair.*

Thus $(\mathcal{A}, \geq_1)$ has exactly two preferred repairs, $\mathcal{R}_1$ and $\mathcal{R}_2$, as follows:

$$\begin{aligned}
\mathcal{R}_1 &= \mathcal{P}_1 \cup \{\mathsf{hasProp}(d_2, u), \mathsf{WProp}(d_3), \mathsf{WProp}(d_5)\} \cup \mathcal{P}_4 \\
&= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{WProp}(d_3), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \\
&\quad \mathsf{WProp}(d_5), \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}. \\
\mathcal{R}_2 &= \mathcal{P}_1 \cup \{\mathsf{hasProp}(d_2, u), \mathsf{WProp}(d_3), \mathsf{WoProp}(d_5)\} \cup \mathcal{P}_4 \\
&= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{WProp}(d_3), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \\
&\quad \mathsf{WoProp}(d_5), \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}.
\end{aligned}$$

Consider $(\mathcal{A}, \geq_2)$ and $(\mathcal{A}, \geq_3)$ from Example 4, where the associated stratified ABoxes are $\mathcal{A} = (\mathcal{P}_1, \mathcal{P}_2 \cup \mathcal{P}_3, \mathcal{P}_4)$ and $\mathcal{A} = (\mathcal{P}_1, \mathcal{P}_3, \mathcal{P}_2, \mathcal{P}_4)$, respectively. Following a similar reasoning, the preferred repairs associated with $(\mathcal{A}, \geq_2)$, resp. $(\mathcal{A}, \geq_3)$, are by coincidence the same as above, namely $\mathcal{R}_1$ and $\mathcal{R}_2$.

□

The notion of preferred repairs, initially defined for total preorders, can then be used as a basis for defining a repair associated with a partial preorder $\trianglerighteq$. We call the new setting $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$ (where PR stands for preferred repairs). Like $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$, $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$ considers all the total extensions $\geq$ of $\trianglerighteq$. However, instead of intersecting the non-defeated repairs like in Elect, we consider the intersection of the preferred repairs, denoted by $\mathrm{IPR}(\mathcal{A}, \geq)$, like so:

Definition 9. *Let $(\mathcal{A}, \trianglerighteq)$ be a partially preordered ABox. Let $\geq$ be some total extension of $\trianglerighteq$ and $(\mathcal{A}, \geq)$ be the corresponding totally preordered ABox.*
The intersection of the preferred repairs associated with $(\mathcal{A}, \geq)$ is:

$$\mathsf{IPR}(\mathcal{A}, \geq) = \bigcap \{\mathcal{R} \mid \mathcal{R} \textit{ is a preferred repair of } \geq\}.$$

The preferred repair associated with $(\mathcal{A}, \trianglerighteq)$ is:

$$\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq} \{\mathsf{IPR}(\mathcal{A}, \geq) \mid \geq \textit{ is a total extension of } \trianglerighteq\}.$$

Example 8. *We continue Example 7 and compute the intersection of the preferred repairs associated with $\geq_1$, $\geq_2$ and $\geq_3$.*

$$\begin{aligned} \mathsf{IPR}(\mathcal{A}, \geq_1) &= \mathsf{IPR}(\mathcal{A}, \geq_2) = \mathsf{IPR}(\mathcal{A}, \geq_3) = \mathcal{R}_1 \cap \mathcal{R}_2 \\ &= \mathcal{P}_1 \cup \{\mathsf{hasProp}(d_2, u), \mathsf{WProp}(d_3)\} \cup \mathcal{P}_4. \end{aligned}$$

It follows that the preferred repair associated with $(\mathcal{A}, \trianglerighteq)$ is:

$$\begin{aligned} \mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) &= \mathcal{P}_1 \cup \{\mathsf{hasProp}(d_2, u), \mathsf{WProp}(d_3)\} \cup \mathcal{P}_4 \\ &= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{WProp}(d_3), \\ &\quad \mathsf{TDance}(d_4), \mathsf{hasProp}(d_2, u), \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}. \end{aligned}$$

Notice that: $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) = \mathsf{Elect}(\mathcal{A}, \trianglerighteq) \cup \{\mathsf{hasProp}(d_2, u)\}$. *Hence:*

$$\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subsetneq \mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq).$$

□

Obviously, $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$ is consistent, since it is the intersection of some repairs which are by definition consistent.
Next, we show that the base $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$ is larger than the base $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$.

Proposition 5. *Let $(\mathcal{A}, \trianglerighteq)$ be a partially preordered ABox.*
Then $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subseteq \mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$. The converse is false.

Proof. Consider an assertion $f \in \mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ but $f \notin \mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)$. This means that there is some extension $\geq$ of $\trianglerighteq$ and some preferred repair $\mathcal{R}$ of $\geq$ s.t. $f \notin \mathcal{R}$. Let $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be the well-ordered partition associated with $\geq$.
Assume that $f \in \mathcal{S}_i$ for some $i \in \{1, \ldots, n\}$. Since f is elected, it follows that the set $[(\mathcal{R} \cap \mathcal{S}_1) \cup \ldots \cup (\mathcal{R} \cap \mathcal{S}_i)] \cup \{f\}$ is consistent. Indeed:

i) by the definition of a repair: $[(\mathcal{R} \cap \mathcal{S}_1) \cup \ldots \cup (\mathcal{R} \cap \mathcal{S}_i)]$ is consistent,

ii) since f is elected, there is no $g \in [(\mathcal{R} \cap \mathcal{S}_1) \cup \ldots \cup (\mathcal{R} \cap \mathcal{S}_i)]$ s.t. $\{f, g\}$ are conflicting (since all the elements conflicting with f are in $\mathcal{S}_j$ with $j > i$).

Hence one can construct a repair $\mathcal{R}'$ that contains:
$[(\mathcal{R} \cap \mathcal{S}_1) \cup \ldots \cup (\mathcal{R} \cap \mathcal{S}_i)] \cup \{f\}$. This, by definition, means that $\mathcal{R}'$ is strictly preferred to $\mathcal{R}$, which contradicts the fact that $\mathcal{R}$ is a preferred repair.

The converse of Proposition 5 does not hold, thus: $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) \not\subset \mathsf{Elect}(\mathcal{A}, \trianglerighteq)$. A counterexample is given in Example 8.

□ □

It is worth mentioning that the method $\mathrm{Partial}_{\mathrm{PR}}$ for computing a preferred repair for a partially preordered ABox $(\mathcal{A}, \trianglerighteq)$ does not compete with Elect in terms of computational criteria. Indeed, when $\trianglerighteq$ is simply a total preorder, $\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) = \mathsf{IPR}(\mathcal{A}, \trianglerighteq)$, and it has been shown that the complexity of $\mathsf{IPR}(\mathcal{A}, \trianglerighteq)$ w.r.t. the size of the ABox is coNP-hard [23].

8 Elect Beyond DL-Lite$_{\mathcal{R}}$

Up until now, we have restricted ourselves to DL-Lite$_{\mathcal{R}}$ for its good trade-off between expressive power and computational complexity. In this section, we take a step further and propose to generalize the Elect method to partially preordered ABoxes that are expressed in languages that are more expressive than DL-Lite$_{\mathcal{R}}$. In particular, we may consider description languages where the assertional conflicts $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ need not be binary (i.e., they involve more than two assertions unlike DL-Lite$_{\mathcal{R}}$).

From a semantic point of view, we see no limitation and the obtained results also collapse with the IAR-repair (for flat ABoxes) and the non-defeated repair (for totally preordered ABoxes). Indeed, both the IAR and the non-defeated semantics are defined independently of the size of the conflicts. In fact, the IAR-repair is simply the intersection of all the maximal repairs, and the non-defeated repair is expressed in terms of the IAR-repair. As a result, the definitions of the IAR and non-defeated semantics need no adaptation. However, the situation may be different regarding computational issues in the presence of conflicts of arbitrary size. Let us first introduce the DL-Elect method then discuss its computational properties.

8.1 The DL-Elect method

The DL-Elect method proceeds like the Elect method. The main difference lies in the notion of elected assertions which needs to be redefined to cater for non-binary conflicts. Intuitively, an assertion is elected if it is strictly preferred to at least one of its opponents in every conflict where it is involved. For the sake of readability, we simple re-use the term ‘elected’ even for non-binary conflicts.

Definition 10. *Let $\mathcal{K}$ be a DL KB with a partially preordered ABox $(\mathcal{A}, \trianglerighteq)$. An assertion $f \in \mathcal{A}$ is elected if for every $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ where $f \in \mathcal{C}$, there is $g \in \mathcal{C}, g \neq f$, s.t. $f \triangleright g$.*

Note that when $\mathcal{C}$ is a binary conflict, Definition 10 amounts to Definition 7.

Similarly to the definition of the set $\mathsf{Elect}(\mathcal{A}, \trianglerighteq)$ in DL-Lite$_{\mathcal{R}}$, we formally define the set $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ in the case of more expressive DL languages.

Definition 11. *Let $\mathcal{K}$ be a DL KB with a partially preordered ABox $(\mathcal{A}, \trianglerighteq)$.*

$$\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq} \{\mathsf{nd}(\mathcal{A}, \geq) \mid \geq \textit{ is a total extension of } \trianglerighteq\}.$$

Let us consider the following modified version of our running example. It is based on the vocabulary introduced in Example 1. In addition, the symbol $\sqcap$ denotes the conjunction of (complex) concepts and $\perp$ denotes the bottom concept, with their well-known DL semantics.

Example 9. *Let $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ be a DL KB, where the TBox is:*

$$\begin{aligned} \mathcal{T}_1 = \{ \quad & 1.\ \mathsf{MDance} \sqsubseteq \mathsf{Dance},\ 2.\ \mathsf{TDance} \sqsubseteq \mathsf{Dance},\ 3.\ \exists \mathsf{hasInst} \sqsubseteq \mathsf{Dance}, \\ & 4.\ \mathsf{MDance} \sqsubseteq \neg \exists \mathsf{hasInst},\ 5.\ \mathsf{FDance} \sqcap \exists \mathsf{hasInst} \sqcap \exists \mathsf{hasProp} \sqsubseteq \perp \}. \end{aligned}$$

Axioms 3 to 5 state, in order, that: anything having an instrument is a dance, the sets of modern dances and those of elements having instruments are disjoint, and an element may not be at the same time a festival dance, something that has a prop and something that has an instrument. Notice that Axiom 5 represents a ternary conflict, hence it cannot be expressed in DL-Lite$_{\mathcal{R}}$. (Recall that disjointness in DL-Lite$_{\mathcal{R}}$ concerns only pairs of concepts or pairs of roles.)

The ABox is:

$$\mathcal{A}_1 = \{\mathsf{TDance}(d_6), \mathsf{MDance}(d_6), \mathsf{FDance}(d_6), \mathsf{hasProp}(d_6, r), \mathsf{hasInst}(d_6, c)\}.$$

The conflict set is given by $\mathsf{Conf}(\mathcal{A}_1) = \{\mathcal{C}_{11}, \mathcal{C}_{12}\}$, where:

$$\begin{aligned} \mathcal{C}_{11} &= \{\mathsf{MDance}(d_6), \mathsf{hasInst}(d_6, c)\}, \\ \mathcal{C}_{12} &= \{\mathsf{FDance}(d_6), \mathsf{hasInst}(d_6, c), \mathsf{hasProp}(d_6, r)\}. \end{aligned}$$

Assume a partial preorder $\trianglerighteq_1$ over $\mathcal{A}_1$ such that:

$$\begin{aligned} & \mathsf{MDance}(d_6) \trianglerighteq_1 \mathsf{hasProp}(d_6, r) \triangleright_1 \mathsf{hasInst}(d_6, c), \quad \textit{and} \\ & \mathsf{TDance}(d_6) \trianglerighteq_1 \mathsf{MDance}(d_6) \triangleright_1 \mathsf{FDance}(d_6) \triangleright_1 \mathsf{hasInst}(d_6, c). \end{aligned}$$

Using Definition 11, one can check that:

$$\mathsf{DL\text{-}Elect}(\mathcal{A}_1, \trianglerighteq_1) = \{\mathsf{TDance}(d_6), \mathsf{MDance}(d_6), \mathsf{FDance}(d_6), \mathsf{hasProp}(d_6, r)\}.$$

$\square$

We now provide a characterization (without computing all the total preorders) for the set $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ and show that it contains all the elected assertions.

Proposition 6.

1. *An assertion $f \in \mathcal{A}$ is elected in $(\mathcal{A}, \trianglerighteq)$ if and only if $f \in \mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$.*

   (The notions of elected assertion and $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ are given by Definitions 10 and 11 respectively).

2. *Furthermore,* $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ *is consistent w.r.t.* $\mathcal{T}$*.*

Proof. Let $(\mathcal{A}, \trianglerighteq)$ be a partially preordered assertional base.

(1.i) Let $f \in \mathcal{A}$ be an elected assertion. Let us show that for each total extension $\geq$ of $\trianglerighteq$, we have $f \in \mathsf{nd}(\mathcal{A}, \geq)$. Let $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be the well-ordered partition associated with a total extension $\geq$. Assume that $f \in \mathcal{S}_i$ for some $i \in \{1, \ldots, n\}$.
Since f is elected in $(\mathcal{A}, \trianglerighteq)$ and $\geq$ is a total extension of $\trianglerighteq$, applying Definition 10 entails that for every $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ where $f \in \mathcal{C}$, there is $g \in \mathcal{C}, g \neq f$, s.t. $f \triangleright g$ and also $f > g$. This also means that for every $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ s.t. $f \in \mathcal{C}$, there is $g \in \mathcal{C}$ s.t. $g \neq f$ and $g \in \mathcal{S}_j$ with $j > i$. Hence, there is no conflict $\mathcal{C}$ in $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$ where $f \in \mathcal{C}$. (Recall that a conflict is a minimal inconsistent set of assertions w.r.t. $\mathcal{T}$. Hence, removing any element from $\mathcal{C}$ leads to a consistent set of assertions w.r.t. $\mathcal{T}$.) This means that $f \in \mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$. Therefore $f \in \mathsf{nd}(\mathcal{A}, \geq)$. Hence $f \in \mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$.

(1.ii) Let us show the converse. Assume that $f \in \mathcal{A}$ is not elected and let us build a total extension $\geq$ of $\trianglerighteq$ such that $f \notin \mathsf{nd}(\mathcal{A}, \geq)$.
The fact that f is not elected means that there is $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ s.t. $f \in \mathcal{C}$ and for every $g \in \mathcal{C}$, $g \neq f$, $f \triangleright g$ does not hold. This means that there is a total extension $\geq$ of $\trianglerighteq$ where for every $g \in \mathcal{C}$, $g \neq f$, $g \geq f$. Indeed, it is enough to set the ordering between elements of $\mathcal{C}$ w.r.t. f as follows: for every $g \in \mathcal{C}, g \neq f, g \geq f$, and then complete the remaining relation in such a way to extend $\trianglerighteq$. Let $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be the well-ordered partition associated with $\geq$, and let $f \in \mathcal{S}_i$ for some $i \in \{1, \ldots, n\}$. Since for every $g \in \mathcal{C}, g \geq f$, then for every $g \in \mathcal{C}$, if $f \in \mathcal{S}_i$, $g \in \mathcal{S}_j$ for some $j \leq i$. Hence, for $k = 1, \ldots, n$, $f \notin \mathsf{IAR}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k)$ which means that $f \notin \mathsf{nd}(\mathcal{A}, \geq)$.

(2) To show the consistency of $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ w.r.t. $\mathcal{T}$, let us assume the opposite, so there is a conflict $\mathcal{C} \subseteq \mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$. Since each element in $\mathcal{C}$ is elected, then for every $f \in \mathcal{C}$, there is $g \in \mathcal{C}, g \neq f$, s.t. $f \triangleright g$, which is impossible (the strict order $\triangleright$ is irreflexive and transitive).

□ □

Next, we apply Proposition 6 to Example 9 and reproduce the same result.

Example 9. *(continued) Recall that* $\mathsf{Conf}(\mathcal{A}_1) = \{C_{11}, C_{12}\}$*.* $\mathsf{hasInst}(d_6, c)$ *belongs both to* $\mathcal{C}_{11}$ *and* $\mathcal{C}_{12}$*, but it is strictly less preferred than any of its opponents in both conflict, thus it is not elected. The remaining assertions are elected. Indeed,* $\mathsf{TDance}(d_6)$ *is not involved in any conflict.* $\mathsf{FDance}(d_6)$ *and* $\mathsf{hasProp}(d_6, r)$ *(resp.* $\mathsf{MDance}(d_6)$*) are strictly preferred to (at least) one of their opponents in* $\mathcal{C}_{12}$ *(resp.* $\mathcal{C}_{11}$*). Thus:*

$$\mathsf{DL\text{-}Elect}(\mathcal{A}_1, \trianglerighteq_1) = \{\mathsf{TDance}(d_6), \mathsf{MDance}(d_6), \mathsf{FDance}(d_6), \mathsf{hasProp}(d_6, r)\}.$$

□

In the next section, we discuss the properties of the DL-Elect method from a computational point of view.

8.2 Discussion of computational properties

We have defined the DL-Elect method in a similar way to the Elect method. We have shown that the latter is tractable in DL-Lite$_{\mathcal{R}}$ where the conflicts are binary. Thus maintaining the tractability property for the DL-Elect method may require that the conflicts are handled efficiently. Indeed, the time complexity for computing $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ depends on the time complexity for computing and generating/enumerating the conflict set $\mathsf{Conf}(\mathcal{A})$. If the latter complexity is polynomial and the size of the conflicts is polynomial with respect to the size of the ABox, then the whole process is polynomial. By a polynomial algorithm for computing/generating conflicts, we refer again to the existence of an algorithm that takes as input the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and generates/produces the set of all the assertional conflicts associated with $\mathcal{K}$. The space and time complexity of such algorithm is polynomial w.r.t. the size of the ABox. Notice that checking if some assertion $f \in \mathcal{A}$ is elected simply comes down to: (i) parsing all the assertional conflicts $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$, (ii) for each $\mathcal{C}$, checking if there is some assertion $g \in \mathcal{C}$ that is strictly less preferred than f. This is done in polynomial time w.r.t. the size of $\mathsf{Conf}(\mathcal{A})$ (i.e., the number of conflicts). However, it may happen that the number of conflicts may be exponential. We consider an example where the ABox is highly conflicting with the TBox to aid the discussion.

Example 10. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB in which the TBox has only one negative axiom:*

$$\mathcal{T} = \{\exists R_1 \sqcap \exists R_2 \sqcap \ldots \sqcap \exists R_n \sqsubseteq \bot\},$$

where $R_i, i = 1, \ldots, n$, are roles.
The unique axiom of the TBox excludes situations in which the same individual belongs simultaneously to the domain of each role R_i.
Let $\mathsf{I} = \{a\} \cup \{b_1, b_2, \ldots, b_m\}$ be a set of $(m + 1)$ individuals.
Let us assume that the ABox is defined as follow:

$$\mathcal{A} = \{R_i(a, b_{j_i}) \mid i = 1, \ldots, n\,; j_i = 1, \ldots, m\},$$

where a, b_{j_i} are individuals. Namely, the assertions are obtained by instantiating all the roles by the same m pairs of individuals, where the first element of each pair is the individual a, and the second element of each pair is a different individual within the set $\{b_1, b_2, \ldots, b_m\}$.

One can check that the conflict set is:

$$\mathsf{Conf}(\mathcal{A}) = \{R_1(a, b_{j_1}) \mid j_1 = 1, \ldots, m\} \times \ldots \times \{R_n(a, b_{j_n}) \mid j_n = 1, \ldots, m\},$$

where the operator $\times$ denotes the Cartesian product of sets. It follows that any conflict in $\mathsf{Conf}(\mathcal{A})$ is of the form

$$\mathcal{C} = \{R_1(a, b_{j_1}), R_2(a, b_{j_2}), \ldots, R_n(a, b_{j_n})\}.$$

Namely, each conflict contains exactly one element from each role. This is explained by the fact that the individual a is present in the first component of each couple in each role.
Assume now that the partial preorder $\trianglerighteq$ over $\mathcal{A}$ is such that:

$$R_1(a, b_{j_1}) \triangleright R_i(a, b_{j_i}), \text{ for } i > 1.$$

This partial preorder states that each assertion obtained from the role R_1 is preferred to any assertion obtained from another role R_i (with $i > 1$). The other assertions are considered by default incomparable.
One can easily check that the only assertions that are elected are those of R_1. Indeed, let $R_1(a, b_1)$ be an assertion of the role R_1. Recall that each conflict that involves $R_1(a, b_1)$ is necessarily of the form

$$\mathcal{C} = \{R_1(a, b_1), R_2(a, b_{j_2}), \ldots, R_n(a, b_{j_n})\}.$$

Now, by definition of the partial preorder, we have:

$$R_1(a, b_1) \rhd R_i(a, b_{j_i}), \forall i = 2, \ldots, n, \forall j_i = 1, \ldots, m.$$

This means that $R_1(a, b_1)$ is elected.
Using a similar reasoning, we can also state that none of the assertions obtained from the other roles (namely, different from R_1) is elected. Indeed, when such assertion is involved in a conflict, it is either strictly less preferred than an assertion obtained from R_1 or is incomparable with an assertion obtained from other roles. Therefore, applying Proposition 6 to $(\mathcal{A}, \trianglerighteq)$ and $\mathsf{Conf}(\mathcal{A})$ yields:

$$\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq) = \{R_1(a, b_{j_1}) \mid j_1 = 1, \ldots, m\}.$$

Namely, $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ is composed only of all the assertions obtained from the role R_1.
$\square$

Clearly, applying the DL-Elect method in Example 10 by first computing all the conflicts is not appropriate, due to their number (i.e., the size of the conflict set).

9 CElect: A Closure-Based Extension of Elect

In this section, we briefly discuss a method for enlarging the consistent sub-base computed by the Elect method while still maintaining the safety of the obtained result. A natural solution to achieve this aim is to use the notion of *positive deductive closure* in which the closure of the ABox is defined with respect to the positive inclusion axioms of the TBox. This way, the consistency of the derived elements is ensured. Let us first introduce the closure operator [13, 32]:

Definition 12. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB. Let $\mathcal{T}_p$ be the set of all the positive inclusion axioms of $\mathcal{T}$. The deductive closure of $\mathcal{A}$ w.r.t. $\mathcal{T}$ is given by:*
$cl(\mathcal{A}) = \{B(a) \mid \langle \mathcal{T}_p, \mathcal{A} \rangle \models B(a)\, s.t.\, B$ *is a concept in* $\mathcal{T}$, a *is an individual of* $\mathcal{A}\}$
$\cup \{R(a, b) \mid \langle \mathcal{T}_p, \mathcal{A} \rangle \models R(a, b)\, s.t.\, R$ *is a role in* $\mathcal{T}$, a *and* b *are individuals of* $\mathcal{A}\}$.
Here $\models$ is a standard DL inference relation.

There are two possible ways for applying the positive closure, namely, either on the initial ABox, or on the non-defeated repairs computed for all the total extensions of the partial preorder (as per Definition 6).

In the first option, applying the positive closure to the initial ABox (in the spirit of the ICAR semantics for non-prioritized ABoxes [51]) raises two concerns. Indeed, applying the closure to the initial ABox may be semantically debatable since this may entail consequences that are derived from questionable assertions. For instance, assume that we have:

the TBox $\{B_1 \sqsubseteq \neg B_2, B_2 \sqsubseteq B_4, B_1 \sqsubseteq B_3\}$, and

the ABox $\{B_1(a), B_2(a), B_3(a)\}$, such that the assertion $B_1(a)$ is strictly preferred to the assertion $B_2(a)$.
Here, $B_1(a)$ is elected but it contradicts $B_2(a)$ from which the assertion $B_4(a)$ can be derived. It follows that adding $B_4(a)$ to the closed ABox is debatable since it is entailed from a questionable assertion in the initial ABox.
Furthermore, the reliability of the derived elements can be defined in various ways. For instance, assume that we have:

the TBox: $\{B_1 \sqsubseteq B_2, B_3 \sqsubseteq B_2\}$, and

the ABox: $\{B_1(x), B_3(x)\}$, such that the preference levels of $B_1(x)$ and $B_3(x)$ are incomparable.

The assertion $B_2(x)$ can be derived from $B_1(x)$ but also from $B_3(x)$. The question is then where to place $B_2(x)$ in terms of preference level. The intuition is to consider that $B_2(x)$ is *at least as plausible as* $B_1(x)$ and $B_3(x)$, but this is not straightforward to define in a general way.

The second option, using the closure operator $cl(\cdot)$, consists in applying the closure on all non-defeated repairs, which is safer. Indeed, the added conclusions are obtained only from assertions that are in non-defeated repairs, which are known to only containing safe assertions. Thus a repair associated with the partial preorder is computed as the intersection of the closed non-defeated repairs [13] associated with the total extensions of the partial preorder. We call this method CElect. Formally:

Definition 13. *Let $(\mathcal{A}, \trianglerighteq)$ be a partially preordered ABox, $cl(\cdot)$ be as in Definition 12 and $\mathsf{nd}(\mathcal{A}, \geq)$ be as in Definition 6.*

$$\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq}\{cl(\mathsf{nd}(\mathcal{A}, \geq)) \mid \geq \text{ is a total extension of } \trianglerighteq\}.$$

Example 11. *We recall that the non-defeated repairs of Example 5 are:*

- $\mathsf{nd}(\mathcal{A}, \geq_1) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3), \mathsf{hasProp}(d_2, u)\} \cup \mathcal{P}_4$
- $\mathsf{nd}(\mathcal{A}, \geq_2) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4$
- $\mathsf{nd}(\mathcal{A}, \geq_3) = \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4$

The positive deductive closure of each of the above non-defeated repairs is:

- $cl(\mathsf{nd}(\mathcal{A}, \geq_1)) = \mathsf{nd}(\mathcal{A}, \geq_1) \cup \mathcal{N} \cup \{\mathsf{Prop}(u)\}$
- $cl(\mathsf{nd}(\mathcal{A}, \geq_2)) = \mathsf{nd}(\mathcal{A}, \geq_2) \cup \mathcal{N}$
- $cl(\mathsf{nd}(\mathcal{A}, \geq_3)) = \mathsf{nd}(\mathcal{A}, \geq_3) \cup \mathcal{N}$

with $\mathcal{N} = \{ \mathsf{Dance}(d_1), \mathsf{WoProp}(d_1), \mathsf{Dance}(d_2), \mathsf{WProp}(d_2), \mathsf{Dance}(d_3), \mathsf{Dance}(d_4), \mathsf{WProp}(d_4), \mathsf{Prop}(h), \mathsf{Prop}(r)\}$.

$$\begin{aligned}
\mathsf{CElect}(\mathcal{A}, \trianglerighteq) &= \mathcal{P}_1 \cup \{\mathsf{WProp}(d_3)\} \cup \mathcal{P}_4 \cup \mathcal{N} \\
&= \{\mathsf{MDance}(d_1), \mathsf{TDance}(d_2), \mathsf{TDance}(d_3), \mathsf{TDance}(d_4), \\
&\quad \mathsf{Dance}(d_1), \mathsf{Dance}(d_2), \mathsf{Dance}(d_3), \mathsf{Dance}(d_4), \mathsf{Prop}(h), \mathsf{Prop}(r), \\
&\quad \mathsf{WoProp}(d_1), \mathsf{WProp}(d_2), \mathsf{WProp}(d_3), \mathsf{WProp}(d_4), \\
&\quad \mathsf{hasProp}(d_3, h), \mathsf{hasProp}(d_4, r)\}.
\end{aligned}$$

Notice from Example 5 that: $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = \mathsf{Elect}(\mathcal{A}, \trianglerighteq) \cup \mathcal{N}$. *Therefore:*

$$\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subsetneq \mathsf{CElect}(\mathcal{A}, \trianglerighteq).$$

However, in this particular case: $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq))$.

Furthermore, recall from Example 8 of Section 7 that:
$\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq) = \mathsf{Elect}(\mathcal{A}, \trianglerighteq) \cup \{\mathsf{hasProp}(d_2, u)\}$.
It follows that: $cl(\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)) = cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq)) \cup \{\mathsf{Prop}(u)\}$.
Therefore, we conclude that:

$$cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq)) \subsetneq cl(\mathsf{Partial}_{\mathsf{PR}}(\mathcal{A}, \trianglerighteq)).$$

□

In order to show that the closure of the repair computed by the Elect method is not always equal to the result of the CElect method, we consider this other modified version of our running example.

Example 12. *Let $\mathcal{K}_2 = \langle \mathcal{T}_2, \mathcal{A}_2 \rangle$ be a DL-Lite$_\mathcal{R}$ KB, where the TBox is:*

$$\mathcal{T}_2 = \{ \quad 1.\, \mathsf{MDance} \sqsubseteq \mathsf{Dance},\ 2.\, \mathsf{FDance} \sqsubseteq \mathsf{Dance},\ 3.\, \exists \mathsf{hasInst} \sqsubseteq \mathsf{Dance},\\ 4.\, \mathsf{FDance} \sqsubseteq \neg \mathsf{MDance},\ 5.\, \mathsf{hasInst} \sqsubseteq \neg \mathsf{hasProp} \}.$$

Axiom 5 means that the set of ordered pairs of individuals linked by the role hasInst *and the set of those linked by the role* hasProp *are disjoint.*

The ABox is: $\mathcal{A}_2 = \{\mathsf{MDance}(d_7),\ \mathsf{FDance}(d_7), \mathsf{hasProp}(d_7, m), \mathsf{hasInst}(d_7, m)\}$[2].

Figure 3(a) depicts the binary conflicts of $\mathsf{Conf}(\mathcal{A}_2)$ *(dotted lines) and a partial preorder* $\trianglerighteq_2$ *over* $\mathcal{A}_2$*, where the preferences between the assertions are either strict (dashed arrows) or incomparable (no arrow).*

Clearly, none of the assertions $\mathsf{MDance}(d_7)$*,* $\mathsf{FDance}(d_7)$ *and* $\mathsf{hasInst}(d_7, m)$ *belongs to* $\mathsf{Elect}(\mathcal{A}_2, \trianglerighteq_2)$*, because none of them is strictly preferred to its opponent (see Figure 3(a)). However, only those three assertions allow to derive* $\mathsf{Dance}(d_7)$ *from* $\langle \mathcal{T}_2, \mathcal{A}_2 \rangle$*. Therefore:* $\mathsf{Dance}(d_7) \notin cl(\mathsf{Elect}(\mathcal{A}_2, \trianglerighteq_2))$.

On the other hand, $\mathsf{FDance}(d_7) \bowtie_2 \mathsf{hasInst}(d_7, m)$*. Hence, for any total extension* $\geq$ *of* $\trianglerighteq_2$*, either the two assertions are equally preferred, or one assertion is strictly preferred to the other (see Figures 3(b),(c),(d)). Therefore, for any extension* $\geq$*, either* $\mathsf{FDance}(d_7) \in \mathsf{nd}(\mathcal{A}_2, \geq)$ *or* $\mathsf{hasInst}(d_7, m) \in \mathsf{nd}(\mathcal{A}_2, \geq)$*, or both. Hence,* $\mathsf{Dance}(d_7) \in cl(\mathsf{nd}(\mathcal{A}_2, \geq))$*. This means that* $\mathsf{Dance}(d_7) \in \mathsf{CElect}(\mathcal{A}_2, \trianglerighteq_2)$*. It follows that:*

$$cl(\mathsf{Elect}(\mathcal{A}_2, \trianglerighteq_2)) \subsetneq \mathsf{CElect}(\mathcal{A}_2, \trianglerighteq_2).$$

□

Interestingly, CElect is still consistent and is equivalent to the closure of IAR (which is different from ICAR) for non-prioritized ABoxes. And for totally preordered ABoxes, CElect is equivalent to the closure of the non-defeated repair.

Proposition 7.

1. $\langle \mathcal{T}, \mathsf{CElect}(\mathcal{A}, \trianglerighteq) \rangle$ *is consistent.*
2. $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = cl(\mathsf{IAR}(\mathcal{A}, \trianglerighteq))$ *when* $\trianglerighteq$ *is non-prioritized (i.e., flat).*
3. $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = cl(\mathsf{nd}(\mathcal{A}, \trianglerighteq))$ *when* $\trianglerighteq$ *is a total preorder.*
4. $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subseteq \mathsf{CElect}(\mathcal{A}, \trianglerighteq)$ *and* $cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq)) \subseteq \mathsf{CElect}(\mathcal{A}, \trianglerighteq)$.
   The converse is false for both inclusions.

Proof. The proofs are quite straightforward.

(1) For the consistency property, recall that if $\geq$ is a total preorder over the ABox $\mathcal{A}$, then $\mathsf{nd}(\mathcal{A}, \geq)$ is consistent. Hence for each total extension $\geq_i$ of $\trianglerighteq$, both $\mathsf{nd}(\mathcal{A}, \geq_i)$ and its closure $cl(\mathsf{nd}(\mathcal{A}, \geq_i))$ are consistent. It follows that $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq_i} \{cl(\mathsf{nd}(\mathcal{A}, \geq_i)) \mid \geq_i \text{ is a total extension of } \trianglerighteq\}$ is consistent.

[2]Here the dance d_7 is assumed to have drums both as props and instruments.

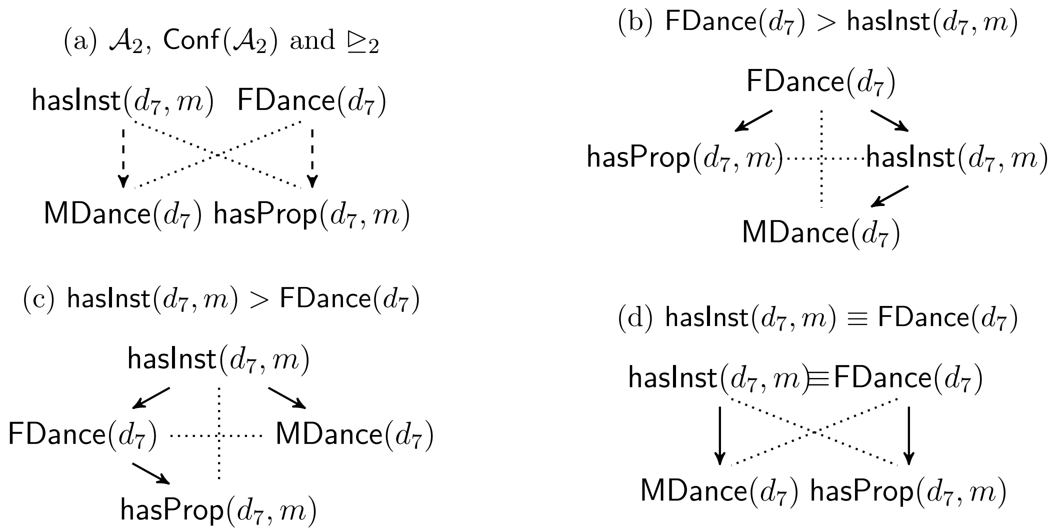


Figure 3: Dotted lines indicate the binary conflicts in $\mathsf{Conf}(\mathcal{A}_2)$. Dashed (resp. solid) arrows indicate strict preference according to $\trianglerighteq_2$ (resp. $\geq$) over $\mathcal{A}_2$.

(2) When $\trianglerighteq$ is flat (namely: for every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, $f \trianglerighteq g$ and $g \trianglerighteq f$), there is a single extension $\geq$ of $\trianglerighteq$ s.t. for every $f \in \mathcal{A}$, for every $g \in \mathcal{A}$, $f \geq g$ and $g \geq f$, which means that there is a single partition $\mathcal{A} = \mathcal{S}$. Hence $\mathsf{nd}(\mathcal{A}, \trianglerighteq) = \mathsf{IAR}(\mathcal{S}) = \mathsf{IAR}(\mathcal{A}, \trianglerighteq)$. Thus by definition, $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) = cl(\mathsf{IAR}(\mathcal{A}, \trianglerighteq))$.

(3) By construction of $\mathsf{CElect}(\mathcal{A}, \trianglerighteq)$, it is easy to check that when $\trianglerighteq$ is a total preorder, then $\mathsf{CElect}(\mathcal{A}, \trianglerighteq)$ collapses with $cl(\mathsf{nd}(\mathcal{A}, \trianglerighteq))$.

(4) By definition, $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq_i} \{\mathsf{nd}(\mathcal{A}, \geq_i) \mid \geq_i \text{ is a total extension of } \trianglerighteq\}$. So for each total extension $\geq_i$ of $\trianglerighteq$, $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subseteq \mathsf{nd}(\mathcal{A}, \geq_i)$. Hence $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subseteq cl(\mathsf{nd}(\mathcal{A}, \geq_i))$ and consequently $cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq)) \subseteq cl(\mathsf{nd}(\mathcal{A}, \geq_i))$.
It follows that:
$\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subseteq \bigcap_{\geq_i} \{cl(\mathsf{nd}(\mathcal{A}, \geq_i)) \mid \geq_i \text{ is a total extension of } \trianglerighteq\} = \mathsf{CElect}(\mathcal{A}, \trianglerighteq)$. Example 11 serves as a counterexample to show that the converse is false. Indeed, since there is at least one case where $\mathsf{Elect}(\mathcal{A}, \trianglerighteq) \subsetneq \mathsf{CElect}(\mathcal{A}, \trianglerighteq)$, it follows that $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) \not\subseteq \mathsf{Elect}(\mathcal{A}, \trianglerighteq)$.
Similarly, $cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq)) \subseteq \mathsf{CElect}(\mathcal{A}, \trianglerighteq)$.
Example 12 serves as a counterexample to show that the converse is false. Namely, $\mathsf{CElect}(\mathcal{A}, \trianglerighteq) \not\subseteq cl(\mathsf{Elect}(\mathcal{A}, \trianglerighteq))$.

□ □

The CElect method produces a single repair for an inconsistent ABox that is, as expected, more productive than a repair computed with the Elect method.

10 Related work

Inconsistency management in knowledge bases is an important task because it is present in several application fields. Given an inconsistent set of beliefs (e.g. obtained by merging multiple-source coherent knowledge bases, or due to the presence of exceptional facts added to a rule base with exceptions), the goal is to answer queries and to provide

plausible conclusions.
Different attitudes can be followed in the presence of inconsistency in knowledge bases:

1. Replace first the inconsistent knowledge base by one or several of its consistent sub-bases, then apply the standard inference machinery of the underlying language (e.g. propositional logic, Description Logic, etc).
   The difficulty of this approach is to find, on the one hand, the relevant beliefs that must be kept and, on the other hand, the beliefs that can be removed. To restore the consistency, it is possible to select either a single coherent sub-base or several coherent sub-bases.
   The selection of a single sub-base is determined thanks to a unique choice of deletion of formulas. The selection of several sub-bases comes from the consideration of different possible restoration solutions.

2. Accept inconsistency and reason in its presence by modifying the standard inference relation of the considered language. This attitude towards inconsistency does not allow the application of standard inference relations (such as propositional logic) and requires to manage inconsistency at a higher level. Paraconsistent logics [2, 5, 33, 36, 61, 61] and argumentation frameworks [18, 22, 41, 70] are examples of such approaches. In the argumentation setting, the idea is to construct arguments (set of pieces of information in favour of a conclusion) and to choose the most relevant argument.

3. Modify the inconsistent knowledge base, in order to restore consistency, by rewriting some of its beliefs. This is especially true in the reasoning tolerant of exceptions, where the information is incomplete and the system must complement the knowledge provided to it.

Our work adopts the first attitude in the presence of contradictory information.

Within the framework of propositional logic, this attitude has been followed by a large body of work (e.g. [10, 15, 42, 44, 64]), where different strategies for restoring the coherence of the knowledge base have been proposed. They are based on the notions of maximal consistent sub-bases, in the context where the available pieces of information all have the same importance level. Associated with these strategies, two main inference mechanisms have been studied: i) the universal inference mechanism (also called inevitable [64] or skeptical in Reiter's default logic [62]) which use all the maximal consistent sub-bases, and ii) the existential inference mechanism (also called brave) that uses only a single maximal consistent sub-base. Other inference mechanisms such as the one based on the intersection of consistent maximal sub-bases [14] have been proposed.
Note that the notion of maximal consistent sub-bases has been used in other contexts such as in default reasoning (with the notion of extensions) or in model-based diagnosis (where a dual notion, called "hitting sets", is used [47, 49, 63]).

The problem of inconsistency studied in propositional logic frameworks was then extensively studied in database settings. In this context, databases may be inconsistent with respect to a set of integrity constraints. In the presence of inconsistent databases, the counterpart of inconsistent maximal sub-bases is called a repair (e.g. [3, 20, 43, 67]). When dealing with inconsistent databases, restoring consistency is not always possible. The effort is then mainly focused on processing queries [4, 21, 34, 35, 73]. In particular, the notion of consistent query answering (CQA) is proposed where only answers to a query that can be obtained from all the database repairs are taken into account.

The problem of managing inconsistencies has also been dealt with within the framework of ontologies and Description Logics, including within the framework of lightweight ontologies and lightweight Description Logics. There is a large body of work that studies both the semantics of inconsistency management approaches (e.g. [45, 46, 51, 66]), inconsistency-tolerant query answering (e.g. [52,53]) as well as the computational complexity impact induced by the presence of contradictory information (e.g. [25,65]).

One of the characteristics of databases and Description Logics compared to propositional logic is the natural separation between the data (databases and ABox) from the integrity constraints (in the context of databases) and the terminological base, i.e., the TBox (in the context of Description Logic). The methods of handling inconsistent bases written in propositional logic do not distinguish factual elements from generic elements. In Description Logics, a whole family of inconsistency management calls into question only the factual elements (that is to say the elements of the ABox). The notion of repairs (or extensions or even maximal consistent sub-bases) then becomes assertion repairs to highlight the character of questioning factual assertions and not generic information (considered stable). Our approach also follows this line of inconsistency management because our Elect method is also interested in questioning only elements of the ABox.

Among the two most famous existing semantics based on assertional repair, we find the AR semantics and the IAR semantics detailed in Section 3. In [24], an approximation of the AR semantics, in the framework of DL-Lite logics, has been proposed. This is done by introducing the notion of k-supporters (and its dual notion k-defeaters) for a query. Basically, a query is k-supported if there are k consistent subsets A_i ($i = 1, \ldots, k$) of the ABox, each of them infers the query and each repair $\mathcal{R}$ contains at least one of these consistent subsets A_i. For $k = 1$, the IAR semnantics is recovered and a more productive relation (than IAR semantics) is obtained for k greater than one.

Recently, other parameterized strategies have been proposed for the management of inconsistencies in knowledge bases. These strategies have been defined within frameworks that encompass DL-Lite logics, namely Datalog $\pm$ [30, 38, 55] (see also [56] for computational complexity analysis) and existential rules [8,9,54,59,60]. Parameterized inference defined in Datalog $\pm$, called k-lazy semantics, is in the spirit of the k-suppoter semantics. When k is equal to 0, we find the IAR semantics (just like 1-supporter), and when k is very large, we tend towards the AR semantics. However, there is still a difference between the k-lazy semantics and the k-supporter semantics which lies in the fact that the k-lazy semantics is not an approximation of the AR semantics. In the framework of existential rules, a general framework for capturing a large spectrum of inconsistency-tolerant semantics have been proposed. The parameterized semantics is defined with respect to a set of three basic modifiers of the set of ABoxes (called MBox) and a set of inference strategies. This parameterized semantics allows to recover the IAR semantics and many of its extensions.

It is often the case that dealing with inconsistency comes down to choosing among different possible contradictory options. Achieving this choice is not an easy task. This is why it is important to have a preference relation between different pieces of information of the knowledge base. Taking into account preferences between information facilitates the handling of inconsistency (e.g. [29]).

Preference between the sources can be of different natures. Oftentimes, the inconsistency-tolerant approaches try to determine a stratification on the knowledge base which induces a total preorder between the different pieces of information. A stratification allows a more easy management of inconsistencies. However, this also leads to compare certain pieces of information with others while they are incomparable and independent. This is why in this paper, partially ordered ABoxes are used. One of the common point with the parameterized strategies, summarized above, is to go beyond the IAR semantics and identify situations where this can be done efficiently. The Elect method proposed in this paper was designed in this spirit, as it provides tractable mechanisms for handling partially ordered inconsistent knowledge bases that also cover the IAR semantics.

11 Concluding discussions

We have tackled the problem of restoring consistency of a partially preordered ABox that may be inconsistent with respect to the TBox in DL-Lite ontologies. We have proposed a method, called Elect, which generalizes the IAR semantics (flat ABox) and the non-defeated semantics (totally preordered ABox). Basically, using Elect, a partial preorder is viewed as a family of total preorders to which non-defeated inference is applied, thus producing non-defeated repairs which are then intersected to obtain a single repair. We have introduced the notion of elected assertions that allows for an equivalent characterization of Elect. Most importantly, we have shown that the complexity of Elect is polynomial.

Furthermore, we have briefly analyzed three possible ways for going beyond the Elect method. The first way concerns improving the productivity (i.e., increasing the size of the repair) by using a preference-based semantics as the backbone of Elect. However this impacts negatively on the complexity, even in DL-Lite. The second way concerns increasing the size of the repair by producing only safe assertions. To this end, we have defined the method CElect that computes a larger repair compared to the result of the Elect method by using the notion of positive deductive closure.

The third way redefines Elect in the general context of DLs that are more expressive than DL-Lite to accommodate the fact that conflicts may involve more than two assertions. From a semantic point of view, the characterizations of Elect, that have been obtained in the context of DL-Lite, have been shown to remain valid for richer languages. Regarding the computational complexity when considering a richer language in which the conflicts are no longer binary. If the method for computing/generating those non-binary conflicts is done in polynomial time and the size of the conflict set is also polynomial (both are evaluated w.r.t. the size of the ABox), then the extension of Elect to the richer language can be done in polynomial time. In future work, we plan to study situations where Elect can be efficiently computed even when the size of the conflict set (i.e., the number of conflicts) is exponential. In what follows, we briefly discuss our intuition and reserve more detailed investigations for future work. Recall that in the DL-Lite family of languages, checking inconsistency and computing the conflict set of a KB starts by defining the negative closure of the TBox (i.e., the set of all the negative axioms that can be derived from the TBox). Then in order to check whether the KB is consistent, it is enough to check whether the ABox is consistent with each axiom in the negative closure of the TBox. Hence, consistency checking comes down to evaluating a set of Boolean queries (one per negative axiom) over the ABox.

Now, assume that a DL language follows the same consistency checking procedure as DL-Lite. So starting from the negative closure of the TBox, checking the consistency of the KB is reduced to answering a set of Boolean queries, associated with each negative axiom, and posed over the ABox. In this case, we argue that the calculation of $\mathsf{DL\text{-}Elect}(\mathcal{A}, \trianglerighteq)$ can be done in a tractable way.

Recall that an assertion f is said to be elected if for any conflict $\mathcal{C} \in \mathsf{Conf}(\mathcal{A})$ that involves f, there is an assertion $g \in \mathcal{C}$ such that f is strictly preferred to g. We suggest that instead of explicitly computing the whole conflict set, we would first simply need to exhibit the set of all the assertions to which f is not strictly preferred, namely: $Gt(\mathcal{A}, f) = \{g \mid g \in \mathcal{A}, f \not\triangleright g\}$. The idea is then to check whether there is an assertion in $Gt(\mathcal{A}, f)$ that conflicts with f. If it is the case, then f is not elected, otherwise it is elected. This can be achieved as follows: For each assertion $B(a)$ (resp. $R(a,b)$) in $\mathcal{A}$, it suffices to focus on the negative axioms that contain the concept B (resp. the role R). Then for each of these axioms, the associated query is augmented with the assertion $B(a)$ (resp. $R(a,b)$) and posed over the sub-ABox $Gt(\mathcal{A}, f) \cup \{R(a,b)\}$. For instance, let us consider our Example 10 where for sake of simplicity we consider that there are only three roles. The TBox $\mathcal{T}$ contains only the following negative axiom: $\mathcal{T} = \{\exists R_1 \sqcap \exists R_2 \sqcap \exists R_3 \sqsubseteq \bot\}$. The consistency test obtained from this axiom is the Boolean query:

$$q() = \exists x, \exists y_1, \exists y_2, \exists y_3, (R_1(x, y_1) \wedge R_2(x, y_2) \wedge R_3(x, y_3)).$$

Assume that we are interested in checking whether $R_1(a, b_1)$ is elected. Recall first that $Gt(\mathcal{A}, R_1(a, b_1))$ contains all the assertions g of the ABox $\mathcal{A}$ such that $R_1(a, b_1))$ is not strictly preferred to g. Namely, $Gt(\mathcal{A}, R_1(a, b_1)) = \{R_1(a, b_{j_1}) \mid j_1 = 1, \ldots, m\}$. Then it is enough to pose the new query:

$$q'() = \exists y_2, \exists y_3, (R_1(a, b_1) \wedge R_2(a, y_2) \wedge R_3(a, y_3))$$

over $Gt(\mathcal{A}, R_1(a, b_1))$. The answer is NO, hence $R_1(a, b_1)$ is elected. Following the same reasoning as in this small example, we are able to parse all the assertions of the initial ABox and determine whether they are elected, without having to exhibit all the conflicts.

In the framework of the AniAge project, an application of the present work is query answering from ontologies representing Southeast Asian traditional dances. Completion of a dance ontology is performed by domain experts by annotating dance videos with respect to the TBox to express the cultural knowledge behind particular dance postures, costumes, props, etc. Annotations of a given dance video can be translated into an ABox. Several experts may annotate the same video and they may attach confidence levels to their annotations. However different experts may not share the same meaning of confidence scales. This can be captured by applying a partial preorder on assertions of the ABox. Furthermore different experts may not share the same opinion on particular elements of a dance, so they may disagree in their annotations. This gives rise to inconsistencies (conflicts) in the ABox with respect to the TBox, making the whole KB inconsistent. Standard query answering tools are not appropriate in such a case as anything can be derived from an inconsistent KB. One way of addressing this issue is to define strategies for querying inconsistent KBs that also take into account priority levels to compute the answers.

Acknowledgements: This work has received support from the European Project

H2020 Marie Sklodowska- Curie Actions (MSCA), Research and Innovation Staff Exchange (RISE): Aniage project (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content), project number 691215. This paper has also been supported by University of Artois within the project AAP A2U QUID (QUeryIng heterogeneous Data). The authors would like to thank the reviewers for their useful comments. In particular, Sections 8.2 and 10 have been added upon request from the reviewers.

References

[1] Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic **50**, 510–530 (1985)

[2] Anderson, A.R., Belnap, N.D.: Entailment: The Logic of Relevance and Neccessity, Vol. I. Princeton University Press (1975)

[3] Arenas, M., Bertossi, L., Chomicki, J.: Consistent query answers in inconsistent databases. In: Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. pp. 68–79. PODS'99, Association for Computing Machinery, New York, NY, USA (1999)

[4] Arenas, M., Bertossi, L., Chomicki, J.: Answer sets for consistent query answering in inconsistent databases. Theory and Practice of Logic Programming **3**(4), 393–424 (2003)

[5] Arieli, O., Avron, A., Zamansky, A.: Maximal and premaximal paraconsistency in the framework of three-valued semantics. Studia Logica **97**(1), 31–60 (2011)

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

[7] Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation, and Applications (2007)

[8] Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artificial Intelligence **175**(9), 1620–1654 (2011)

[9] Baget, J., Benferhat, S., Bouraoui, Z., Croitoru, M., Mugnier, M., Papini, O., Rocher, S., Tabia, K.: A general modifier-based framework for inconsistency-tolerant query answering. In: KR, Cape Town, South Africa. pp. 513–516 (2016)

[10] Baral, C., Kraus, S., Minker, J., Subrahmanian, V.S.: Combining knowledge bases consisting of first-order analysis. Computational Intelligence **8**, 45–71 (1992)

[11] Belabbes, S., Benferhat, S.: Inconsistency handling for partially preordered ontologies: Going beyond elect. In: KSEM (1), Athens, Greece. pp. 15–23 (2019)

[12] Belabbes, S., Benferhat, S., Chomicki, J.: Elect: An inconsistency handling approach for partially preordered lightweight ontologies. In: LPNMR, Philadelphia, USA. pp. 210–223 (2019)

[13] Benferhat, S., Bouraoui, Z., Tabia, K.: How to select one preferred assertional-based repair from inconsistent and prioritized DL-Lite knowledge bases? In: IJCAI, Buenos Aires, Argentina. pp. 1450–1456 (2015)

[14] Benferhat, S., Dubois, D., Prade, H.: Representing default rules in possibilistic logic. In: Knowledge Representation and Reasoning. pp. 673–684 (1992)

[15] Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsistent beliefs without revising? In: IJCAI, Montréal, Canada. pp. 1449–1457 (1995)

[16] Benferhat, S., Dubois, D., Prade, H.: Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study. Part 2 : the prioritized case, vol. 24, pp. 473–511. Physica-Verlag, Heidelberg (1998)

[17] Benferhat, S., Bouraoui, Z., Chadhry, H., Shafry Bin Mohd Rahim Fc, M., Tabia, K., Telli, A.: Characterizing non-defeated repairs in inconsistent lightweight ontologies. In: SITIS, Naples, Italy. pp. 282–287 (2016)

[18] Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge bases. In: Heckerman, D., Mamdani, E.H. (eds.) UAI '93: Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence, The Catholic University of America, Providence, Washington DC, USA. pp. 411–419. Morgan Kaufmann (1993)

[19] Benferhat, S., Lagrue, S., Papini, O.: Reasoning with partially ordered information in a possibilistic logic framework. Fuzzy Sets and Systems **144**(1), 25–41 (2004)

[20] Bertossi, L.: Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)

[21] Bertossi, L.: Consistent query answering in databases. SIGMOD Record **35**(2), 68–76 (2006)

[22] Besnard, P., Hunter, A.: Elements of Argumentation. The MIT Press (2008)

[23] Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In: AAAI, Québec, Canada. pp. 996–1002 (2014)

[24] Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: IJCAI, Beijing, China. pp. 775–781 (2013)

[25] Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Canada (2012)

[26] Bienvenu, M., Bourgaux, C.: Inconsistency-tolerant querying of description logic knowledge bases. In: Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering. vol. 9885, pp. 156–202. LNCS. Springer (2016)

[27] Bienvenu, M., Bourgaux, C., Goasdoué, F.: Query-driven repairing of inconsistent DL-Lite knowledge bases. In: IJCAI, New York, USA. pp. 957–964 (2016)

[28] Bienvenu, M., Bourgaux, C., Goasdoué, F.: Computing and explaining query answers over inconsistent DL-Lite knowledge bases. Journal of Artificial Intelligence Research **64**, 563–644 (2019)

[29] Brewka, G.: Preferred subtheories: An extended logical framework for default reasoning. In: IJCAI, Detroit, USA. pp. 1043–1048 (1989)

[30] Calì, A., Gottlob, G., Lukasiewicz, T.: A general Datalog-based framework for tractable query answering over ontologies. Journal of Web Semantics **14**, 57–83 (2012), special Issue on Dealing with the Messiness of the Web of Data

[31] Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of DL-Lite knowledge bases. In: International Semantic Web Conference (1). pp. 112–128 (2010)

[32] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning **39**(3), 385–429 (2007)

[33] Carnielli, W., Coniglio, M., Marcos, J.: Logics of Formal Inconsistency, pp. 1–93 (2007)

[34] Chomicki, J.: Consistent query answering: Five easy pieces. In: Database Theory - ICDT 2007. pp. 1–17 (2007)

[35] Chomicki, J., Marcinkowski, J., Staworko, S.: Computing consistent query answers using conflict hypergraphs. pp. 417–426. CIKM'04, Association for Computing Machinery, New York, NY, USA (2004)

[36] da Costa, N.C.A.: On the theory of inconsistent formal systems. Notre Dame J. Formal Logic **15**(4), 497–510 (10 1974)

[37] Cozman, F.G.: Credal networks. Artificial Intelligence **120**, 199–233 (2000)

[38] Deagustini, C.A.D., Martinez, M.V., Falappa, M.A., Simari, G.R.: Datalog+/- ontology consolidation. Journal of Artificial Intelligence Research **56**, 613–656 (2016)

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

[40] Dubois, D., Fargier, H., Prade, H.: Ordinal and probabilistic representations of acceptance. Journal of Artificial Intelligence Research **22**, 23–56 (2004)

[41] Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321–358 (1995)

[42] Finkelstein, A., Gabbay, D.M., Hunter, A., Kramer, J., Nuseibeh, B.: Inconsistency handling in multi-perspective specifications. In: Sommerville, I., Paul, M. (eds.) Software Engineering - ESEC '93, 4th European Software Engineering Conference, Garmisch-Partenkirchen, Germany. Lecture Notes in Computer Science, vol. 717, pp. 84–99. Springer (1993)

[43] Fuxman, A., Miller, R.J.: First-order query rewriting for inconsistent databases. In: Database Theory - ICDT 2005, 10th International Conference, Edinburgh, UK. pp. 337–351 (2005)

[44] Gabbay, D., Hunter, A.: Making inconsistency respectable: A logical framework for inconsistency in reasoning, part I – a position paper. In: Jorrand, P., Kelemen, J. (eds.) Fundamentals of Artificial Intelligence Research. pp. 19–32. Springer Berlin Heidelberg (1991)

[45] Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings. pp. 353–367 (2005)

[46] Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, UK. pp. 454–459 (2005)

[47] Ignatiev, A., Morgado, A., Weissenbacher, G., Marques-Silva, J.: Model-based diagnosis with multiple observations. In: Kraus, S. (ed.) Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China. pp. 1108–1115 (2019)

[48] Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence **3**(52), 263–294 (1991)

[49] de Kleer, J.: Hitting set algorithms for model-based diagnosis. In: DX Workshop'11. pp. 100–105 (2011)

[50] Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to query answering in DL-Lite. In: 12th International Conference on Principles of Knowledge Representation and Reasoning (KR), Toronto, Canada. pp. 247–257 (2010)

[51] Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: International Conference on Web Reasoning and Rule Systems, Bressanone, Italy. LNCS, vol. 6333, pp. 103–117 (2010)

[52] Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Query rewriting for inconsistent DL-Lite ontologies. In: Web Reasoning and Rule Systems - 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings. pp. 155–169 (2011)

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

[54] Lukasiewicz, T., Martinez, M.V., Pieris, A., Simari, G.I.: From classical to consistent query answering under existential rules. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence 2015. pp. 1546–1552 (2015)

[55] Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Inconsistency handling in Datalog+/- ontologies. In: ECAI, Montpellier, France. pp. 558–563 (2012)

[56] Lukasiewicz, T., Malizia, E., Vaicenavičius, A.: Complexity of inconsistency-tolerant query answering in Datalog+/- under cardinality-based repairs. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 2962–2969 (2019)

[57] Martinez, M.V., Parisi, F., Pugliese, A., Simari, G.I., Subrahmanian, V.S.: Inconsistency management policies. In: KRR. pp. 367–377 (2008)

[58] Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles. W3C Recommendation. 11 December 2012. Available at https://www.w3.org/TR/owl2-profiles/

[59] Mugnier, M.: Data access with horn ontologies: Where description logics meet existential rules. Künstliche Intelligenz **34**(4), 475–489 (2020)

[60] Mugnier, M., Thomazo, M.: An introduction to ontology-based query answering with existential rules. In: Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School, Athens, Greece. Lecture Notes in Computer Science, vol. 8714, pp. 245–278. Springer (2014)

[61] Priest, G.: Paraconsistent Logic. In: Handbook of Philosophical Logic, pp. 287–393. Springer, Dordrecht (2002)

[62] Reiter, R.: A logic for default resoning. Artificial Intelligence **13**, 81–132 (1980)

[63] Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **32**(1), 57–95 (1987)

[64] Rescher, N., Manor, R.: On inference from inconsistent premisses. Theory and Decision **1**(2), 179–217 (1970)

[65] Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Spain. pp. 1057–1062 (2011)

[66] Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. pp. 355–362 (2003)

[67] Staworko, S., Chomicki, J., Marcinkowski, J.: Prioritized repairing and consistent query answering in relational databases. AMAI **64**(2-3), 209–246 (2012)

[68] Telli, A., Benferhat, S., Bourahla, M., Bouraoui, Z., Tabia, K.: Polynomial algorithms for computing a single preferred assertional-based repair. Künstliche Intelligenz **31**(1), 15–30 (2017)

[69] Touazi, F., Cayrol, C., Dubois, D.: Possibilistic reasoning with partially ordered beliefs. Journal of Applied Logic **13**(4), 770–798 (2015)

[70] Toulmin, S.: The uses of argument. Cambridge University Press (1956)

[71] Trivela, D., Stoilos, G., Vassalos, V.: Querying expressive DL ontologies under the ICAR semantics. In: Proceedings of the 31st DL workshop. Tempe, USA (2018)

[72] Tsalapati, E., Stoilos, G., Stamou, G., Koletsos, G.: Efficient query answering over expressive inconsistent description logics. In: IJCAI, New York, USA. pp. 1279–1285 (2016)

[73] Wijsen, J.: Foundations of query answering on inconsistent databases. SIGMOD Record **48**(3), 6–16 (2019)