Characterizing the Possibilistic Repair for Inconsistent Partially Ordered Assertions

Sihem Belabbes[1 [0000−0002−8159−7122]] and Salem Benferhat[2 [0000−0002−4853−3637]]

[1] Laboratoire d'Intelligence Artificielle et Sémantique des Données (LIASD), IUT de Montreuil, Université Paris 8, Saint-Denis, France
belabbes@iut.univ-paris8.fr

[2] Centre de Recherche en Informatique de Lens (CRIL), Université d'Artois & CNRS, Lens, France
benferhat@cril.fr

Abstract. Ontologies specified in DL-Lite are commonly used to facilitate query answering. Formally, an ontology is a knowledge base composed of a TBox (a set of axioms) and an ABox (a set of assertions). The assertions may be conflicting with respect to the axioms, so the inconsistency in the ABox should be resolved before querying it. This is usually achieved by computing the set of all the conflicts of the ABox. We have recently proposed a method for handling inconsistency in ontologies where the assertions are partially preordered and uncertain. We have defined π-accepted assertions as those assertions that are more certain than at least one assertion of each conflict in the ABox. In DL-Lite ontologies, a conflict is a subset of two assertions, and the set of all the conflicts can be computed in polynomial time. Thus our method is also polynomial in the ABox's size in DL-Lite. We propose here a new equivalent characterization of π-accepted assertions that is also tractable, but without exhibiting the conflicts beforehand. Instead, it is based on a consistency check, such that an assertion is π-accepted if it is consistent with all the assertions that are at least as certain or that are incomparable to it in terms of certainty degrees. This new characterization allows to generalise the method to description logic languages that are more expressive than DL-Lite and where the conflicts may not be computable efficiently.

Keywords: Inconsistency management · Formal ontologies · DL-Lite.

1 Introduction

An ontology is a Description Logic [1] knowledge base with two components: a TBox and an ABox. The TBox contains terminological knowledge designed by domain experts and encoded in the form of axioms. The ABox is a dataset composed of ground facts about particular entities. Its elements are called assertions and they are usually obtained from various information sources.

Answering queries posed over data pieces that are semantically enriched with domain knowledge has the advantage of deriving new facts from the knowledge base. Nonetheless, the drawback is a potential increase in computational complexity, except for the DL-Lite fragments of Description Logics [12] in which query answering is carried out in polynomial time in the ABox's size.

Query answering should be performed over a consistent knowledge base in order to ensure the validity of the derived conclusions. The TBox's axioms can be safely considered as correct and unquestionable. However, the ABox assertions may be prone to errors, incomplete and potentially contradictory when assessed against the axioms. Therefore, the whole knowledge base may be inconsistent and classical Description Logic semantics cannot be used to compute query answers.

Restoring the consistency of the ABox with respect to the TBox can be addressed using the notion of a repair, defined as a maximally-consistent subset of the ABox, and over which query answers can be computed. Since an inconsistent ABox may admit several repairs, a significant body of work has been devoted to designing strategies (a.k.a. inconsistency-tolerant semantics) for choosing which repair(s) should be queried in lieu of the initial ABox [2–4, 8–11, 14, 16, 17].

Arguably, one of the most well-known strategies is the "Intersection of ABox Repair" (IAR) semantics [13]. Basically, a query answer is a valid conclusion of the knowledge base, called an IAR-consequence, if it can be derived from a single repair obtained from the intersection of all the repairs of the ABox. Equivalently, an IAR-consequence is an assertion that is not involved in any conflict [5], which is defined as a minimally-inconsistent subset of the ABox.

Another imperfection in the data is uncertainty. Possibility theory has been used as the underlying framework to define a formal setting for standard possibilistic DL-Lite [6]. Each assertion is assigned a real number in the unit interval $]0, 1]$ to represent its certainty degree, such that the highest weight where inconsistency is met is called the inconsistency degree of the ABox. The possibilistic repair is defined as a consistent subset of the ABox containing all the assertions that are strictly more certain than the inconsistency degree.

A framework has been recently proposed for the case where the certainty degrees of multi-source data may not be comparable on the same scale [3]. It computes a single repair for the ABox, in the spirit of the IAR semantics. It assumes that the TBox's axioms are fully certain, but that the ABox assertions may be uncertain and are equipped with partially ordered symbolic weights. It proposes a characterization based on the notion of π-accepted assertions, which are the assertions that are more certain than at least one assertion of each conflict. The repair is then the set of all π-accepted assertions. Most notably, it can be computed in polynomial time in the ABox's size in DL-Lite$_{\mathcal{R}}$ [3].

This follows directly from the fact that each conflict in DL-Lite involves (at most) two assertions, and that the conflict set can be computed in polynomial time in the ABox's size [11]. However, in more expressive Description Logic languages, conflicts may involve any number of assertions, and the number of conflicts may be exponential. Hence, the favourable computational properties of this method cannot be guaranteed beyond DL-Lite.

[3] The fragment DL-Lite$_{\mathcal{R}}$ is a dialect of DL-Lite that provides the logical underpinnings for the OWL 2 QL profile [15], which is devoted to query answering.

Exhibiting all the conflicts of the ABox is often a prerequisite for computing repairs. In this research, we introduce a new equivalent characterization of π-accepted assertions that is not based on conflicts. It rather performs a consistency check whereby an assertion is π-accepted if it is consistent with all the assertions that are at least as certain or that are incomparable to it in terms of certainty degrees. This way, the method can be generalized to other Description Logic languages, regardless of the computational complexity of computing the conflicts.

This paper is structured as follows. Section 2 presents some preliminaries. Section 3 recalls the method for computing a partially preordered possibilistic repair. Section 4 introduces our new characterization, before concluding.

2 Preliminaries

2.1 Overview of DL-Lite$_\mathcal{R}$

The DL-Lite$_\mathcal{R}$ language is built upon three countably infinite and mutually disjoint sets. These are: a set $\mathsf{C_N}$ of *concept names*, a set $\mathsf{R_N}$ of *role names* and a set $\mathsf{I_N}$ of *individual names*. The syntax is recursively defined as follows:

- $R := P \mid P^-$ is a *basic role*, with $P \in \mathsf{R_N}$ and its *inverse* $P^- \in \mathsf{R_N}$.
- $E := R \mid \neg R$ denotes a *complex role*.
- $B := A \mid \exists R$, with $A \in \mathsf{C_N}$, stands for a *basic concept*.
- $C := B \mid \neg B$ represents a *complex concept*.

In terms of semantics, an interpretation is a tuple $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$, where $\Delta^\mathcal{I} \neq \emptyset$ and $\cdot^\mathcal{I}$ is an interpretation function mapping concept names A to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, role names P to $P^\mathcal{I} \subseteq (\Delta^\mathcal{I} \times \Delta^\mathcal{I})$, and individual names a to $a^\mathcal{I} \in \Delta^\mathcal{I}$. We extend the function $\cdot^\mathcal{I}$ to interpret complex concepts and roles of DL-Lite$_\mathcal{R}$ as follows:

$$(P^-)^\mathcal{I} = \{(y,x) \in (\Delta^\mathcal{I} \times \Delta^\mathcal{I}) \mid (x,y) \in P^\mathcal{I}\};$$
$$(\exists R)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I} \text{ s.t. } (x,y) \in R^\mathcal{I}\};$$
$$(\neg B)^\mathcal{I} = \Delta^\mathcal{I} \setminus B^\mathcal{I};$$
$$(\neg R)^\mathcal{I} = (\Delta^\mathcal{I} \times \Delta^\mathcal{I}) \setminus R^\mathcal{I}.$$

In this paper, we consider the following vocabulary. Let:

- $\mathsf{C_N} = \{\mathsf{Electric}, \mathsf{Thermal}, \mathsf{Plugin}, \mathsf{Manual}, \mathsf{Energy}\}$, representing resp.: electric vehicle, thermal car, rechargeable plug-in car, manual gearbox and energy type.
- $\mathsf{R_N} = \{\mathsf{useFuel}\}$, which links a thermal car to an energy type. Hence, the inverse $\mathsf{useFuel}^-$ links an energy type to a thermal car.
- $\mathsf{I_N} = \{v_1, v_2, v_3, p\}$, where v_i represents a particular vehicle, and "p" stands for the energy type petrol.

An *inclusion axiom* on concepts (resp. on roles) is a statement of the form $B \sqsubseteq C$ (resp. $R \sqsubseteq E$). Concept inclusions with (resp. without) the negation symbol "$\neg$" on the right of the inclusion symbol are called *negative* (resp. *positive*) inclusion axioms. A TBox $\mathcal{T}$ is a finite set of inclusion axioms. An *assertion* is a statement of the form $A(a)$ or $P(a,b)$, where $a, b \in \mathsf{I_N}$. An ABox $\mathcal{A}$ is a finite set of assertions. A Knowledge Base (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

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. $R^\mathcal{I} \subseteq E^\mathcal{I}$).

Similarly, $\mathcal{I}$ *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. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$). An interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ (resp. $\mathcal{A}$), 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 KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, if $\mathcal{I} \Vdash \mathcal{T}$ and $\mathcal{I} \Vdash \mathcal{A}$.

A KB $\mathcal{K}$ is *consistent* if it admits at least one model, otherwise it is *inconsistent*. A TBox $\mathcal{T}$ is *incoherent* if there is $A \in \mathsf{C_N}$ that is empty in every model of $\mathcal{T}$, otherwise it is *coherent*.
We use the following running example.

Example 1. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ KB, where the TBox is given by:

$$\mathcal{T} = \left\{ \begin{array}{llll} 1.\ \mathsf{Electric} & \sqsubseteq \neg\mathsf{Manual} & 2.\ \mathsf{Thermal} & \sqsubseteq \neg\mathsf{Plugin} \\ 3.\ \exists\mathsf{useFuel} & \sqsubseteq \mathsf{Thermal} & 4.\ \exists\mathsf{useFuel}^- & \sqsubseteq \mathsf{Energy} \end{array} \right\}$$

Axiom 1 indicates that the set of electric vehicles is disjoint from the set of manual transmission vehicles. Axiom 2 indicates that the set of thermal vehicles is disjoint from the set of rechargeable plug-in vehicles. Axiom 3 states that any element using fuel is a thermal vehicle. Axiom 4 specifies that the fuel used by a vehicle is an energy type. Axioms 1 and 2 are negative inclusions on concepts. Consider the flat ABox (the assertions are equally certain):

$$\mathcal{A} = \left\{ \begin{array}{l} \mathsf{Manual}(v_1), \mathsf{Electric}(v_1), \mathsf{Plugin}(v_1), \mathsf{Thermal}(v_2), \\ \mathsf{Plugin}(v_2), \mathsf{Electric}(v_3), \mathsf{useFuel}(v_2, p), \mathsf{Energy}(p) \end{array} \right\}$$

One can see that $\mathcal{K}$ is inconsistent. For example, the assertions $\mathsf{Manual}(v_1)$ and $\mathsf{Electric}(v_1)$ violate Axiom 1.

□

2.2 The IAR semantics

Restoring the consistency of the ABox relies on the notion of ABox repair, inspired from data repair in relational databases to ensure consistent query answering (e.g. see [7]). In the following definitions, we assume $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an inconsistent DL-Lite$_{\mathcal{R}}$ KB.

Definition 1. *A repair, denoted by $\mathcal{R}$, is an inclusion-maximal subset of $\mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{R} \rangle$ is consistent.*

One of the most well-known inconsistency-tolerant semantics is the IAR (Intersection of ABox Repair) semantics [13]. It evaluates queries over a single repair obtained from the intersection of all the repairs of the ABox.

Definition 2. *A query answer is an IAR-consequence of $\mathcal{K}$ if it can be derived from the subset:* $\mathsf{IAR}(\mathcal{A}) = \bigcap \{\mathcal{R} \mid \mathcal{R}$ *is a repair of* $\mathcal{A}\}$.

Answers are returned in polynomial time in the ABox's size in DL-Lite$_{\mathcal{R}}$ [12,13].

Negative inclusion axioms in the TBox allow to specify the disjointness of assertions in the ABox. This is captured by the notion of conflict.

Definition 3. *A conflict, denoted by $\mathcal{C}$, is an inclusion-minimal subset of $\mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{C} \rangle$ is inconsistent.*

We denote the set of all the conflicts in the ABox $\mathcal{A}$ by $\mathsf{Cf}(\mathcal{A})$. We assume that $\mathcal{A}$ does not contain any assertion φ such that $\langle \mathcal{T}, \{\varphi\} \rangle$ is inconsistent. Thus any conflict $\mathcal{C}$ in $\mathsf{Cf}(\mathcal{A})$ is binary, in other words, $|\mathcal{C}| = 2$ [11].

By definition, ABox repairs are conflict-free, so the assertions of the same conflict cannot belong to the same repair. An equivalent characterization for the IAR semantics computes $\mathsf{IAR}(\mathcal{A})$ as the set of free assertions [5, 13], i.e., the assertions of $\mathcal{A}$ that are not involved in any conflict.

Example 2. The KB $\mathcal{K}$ contains three conflicts:

- $\{\mathsf{Manual}(v_1), \mathsf{Electric}(v_1)\}$, which contradicts axiom 1.
- $\{\mathsf{Thermal}(v_2), \mathsf{Plugin}(v_2)\}$, which contradicts axiom 2.
- $\{\mathsf{useFuel}(v_2, p), \mathsf{Plugin}(v_2)\}$, which contradicts axioms 2 and 3.

The ABox $\mathcal{A}$ admits the following four repairs:

- $\mathcal{R}_1 = \{\mathsf{Manual}(v_1), \mathsf{Plugin}(v_1), \mathsf{useFuel}(v_2, p), \mathsf{Thermal}(v_2), \mathsf{Electric}(v_3), \mathsf{Energy}(p)\}$.
- $\mathcal{R}_2 = \{\mathsf{Electric}(v_1), \mathsf{Plugin}(v_1), \mathsf{useFuel}(v_2, p), \mathsf{Thermal}(v_2), \mathsf{Electric}(v_3), \mathsf{Energy}(p)\}$.
- $\mathcal{R}_3 = \{\mathsf{Manual}(v_1), \mathsf{Plugin}(v_1), \mathsf{Plugin}(v_2), \mathsf{Electric}(v_3), \mathsf{Energy}(p)\}$.
- $\mathcal{R}_4 = \{\mathsf{Electric}(v_1), \mathsf{Plugin}(v_1), \mathsf{Plugin}(v_2), \mathsf{Electric}(v_3), \mathsf{Energy}(p)\}$.

The intersection of these repairs yields the set:

$$\mathsf{IAR}(\mathcal{A}) = \{\mathsf{Plugin}(v_1), \mathsf{Electric}(v_3), \mathsf{Energy}(p)\}.$$

□

In the rest of the paper, we present the characterization recently proposed in [3]. We then discuss its shortcomings and introduce a characterization that is more efficient computationally.

3 Partially preordered possibilistic repair

Let us recall the method defined in [3] for computing a possibilistic repair for partially preordered DL-Lite$_\mathcal{R}$ ontologies. Consider a partially ordered uncertainty scale $\mathbb{L} = (\mathsf{U}, \rhd)$, defined over:

- a partially ordered set (POS) $\mathsf{U} = \{u_1, \ldots, u_n, \mathbb{1}\}$, and
- a strict partial order $\rhd$ (i.e., an irreflexive and transitive relation).

The element $\mathbb{1}$ represents full certainty, such that: $\forall u_i \in \mathsf{U} \setminus \{\mathbb{1}\}, \mathbb{1} \rhd u_i$.
Intuitively, the elements of a POS denoted by U represent certainty degrees applied to the ABox assertions. When $u_i \not\rhd u_j$ and $u_j \not\rhd u_i$, we say that u_i and u_j are incomparable and we denote it by $u_i \bowtie u_j$.
A partially preordered DL-Lite$_\mathcal{R}$ KB is a triple $\mathcal{K}_\rhd = \langle \mathcal{T}, \mathcal{A}_\rhd, \mathbb{L} \rangle$ with:

$$\mathcal{A}_\rhd = \{(\varphi_i, u_i) \mid \varphi_i \text{ is a DL-Lite}_\mathcal{R} \text{ assertion}, u_i \in \mathsf{U}\},$$

where a single weight u_i is assigned to each assertion φ_i.
Given two assertions $(\varphi_i, u_i), (\varphi_j, u_j) \in \mathcal{A}_\rhd$, we write $\varphi_i \rhd \varphi_j$ to mean $u_i \rhd u_j$ (i.e., φ_i is strictly preferred to φ_j), and write $\varphi_i \bowtie \varphi_j$ to mean $u_i \bowtie u_j$ (i.e., φ_i and φ_j are incomparable). Note that the relation $\rhd$ on U is a strict partial

order [4]. However, the ABox $\mathcal{A}_{\rhd}$ is partially preordered because the same weight can be assigned to more than one assertion.
Computing the partially preordered possibilistic repair proceeds as follows [3]:

(a) Compute the compatible bases of $\mathcal{A}_{\rhd}$, i.e., consider all the total preorder extensions of $\rhd$ over U.
(b) Compute the possibilistic repair associated with each compatible base.
(c) Intersect all the repairs to obtain a single repair denoted by $\pi(\mathcal{A}_{\rhd})$.

Let $\mathcal{WA} = \{(\varphi_i, \alpha_i) \mid (\varphi_i, u_i) \in \mathcal{A}_{\rhd}, \alpha_i \in]0,1]\}$ be a weighted ABox obtained from $\mathcal{A}_{\rhd}$ by replacing each symbolic weight $u_i \in \mathsf{U}$ with some real number $\alpha_i \in]0,1]$. Then $\mathcal{WA}$ is compatible with $\mathcal{A}_{\rhd}$ if it preserves the strict ordering between the assertions. Formally:

$$\forall(\varphi_i, \alpha_i) \in \mathcal{WA}, \forall(\varphi_j, \alpha_j) \in \mathcal{WA}, \text{ if } \varphi_i \rhd \varphi_j \text{ then } \alpha_i > \alpha_j.$$

The set of real numbers that can be assigned to the assertions is infinite, so there are infinitely many compatible bases. However, it suffices to consider a finite number thereof, i.e., only those that express a distinct preference ordering. The possibilistic repair of a weighted ABox is defined as:

$$\mathcal{R}_\pi(\mathcal{WA}) = \{\varphi \mid (\varphi, \alpha) \in \mathcal{WA}, \alpha > \mathsf{Inc}(\mathcal{WA})\},$$

where $\mathsf{Inc}(\mathcal{WA})$ is the inconsistency degree of $\mathcal{WA}$, i.e., the highest weight attached to an assertion that makes the ABox is inconsistent.
Hence, the partially preordered possibilistic repair of $\mathcal{A}_{\rhd}$ is defined as:

$$\pi(\mathcal{A}_{\rhd}) = \bigcap\{\mathcal{R}_\pi(\mathcal{WA}) \mid \mathcal{WA} \text{ is compatible with } \mathcal{A}_{\rhd}\}.$$

An equivalent characterization [3] of this method produces the same result, without executing the steps (a), (b) and (c) described above. It is based on the notion of π-accepted assertion. Intuitively, this is an assertion that is strictly preferred to at least one assertion of each conflict of $\mathcal{A}_{\rhd}$. The conflict set $\mathsf{Cf}(\mathcal{A}_{\rhd})$ is obtained with a small tweak to Definition 3 to take into account the weights.

Definition 4. *An assertion $(\varphi_i, u_i) \in \mathcal{A}_{\rhd}$ is π-accepted if:*

$$\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\rhd}), \exists(\varphi_j, u_j) \in \mathcal{C}, \varphi_i \neq \varphi_j, \text{ s.t. } \varphi_i \rhd \varphi_j \text{ (i.e., } u_i \rhd u_j).$$

It follows that in DL-Lite$_{\mathcal{R}}$ knowledge bases:

Proposition 1. *The set of all π-accepted assertions (without the weights) is equal to $\pi(\mathcal{A}_{\rhd})$. It can be computed in polynomial time in the ABox's size [3].*

Next, we propose a new characterization that is more computationally efficient.

4 Revisiting π-acceptance

4.1 A new characterization of π-acceptance

In inconsistent knowledge bases that are specified in the lightweight fragment DL-Lite$_{\mathcal{R}}$, each conflict in the ABox is a non-empty subset composed of at most two assertions [11]. In this work, we assume that the ABox does not contain any

[4] Namely, $\forall u_i \in \mathsf{U}$, $\forall u_j \in \mathsf{U}$, if $u_i \rhd u_j$ holds, then $u_j \rhd u_i$ does not hold.

assertion contradicting itself with respect to the axioms of the TBox. It follows that the conflict set is composed only of pairs of conflicting assertions. Hence, the size of the conflict set (i.e., the number of conflicts) is a square polynomial in the ABox's size, in the worst case.

In order to determine whether some assertion of the ABox is π-accepted, the characterization given in Definition 4 parses all the pairs of assertions in the conflict set, and it does so in linear time in the ABox's size. This means that in the worst case, checking the π-acceptance of an assertion can be achieved in square polynomial time in the ABox's size for DL-Lite$_{\mathcal{R}}$ ontologies. However, despite the fact that a square polynomial time complexity is also polynomial, it may be impractical in applications where the main reasoning task consists in answering queries posed over ontologies with high-dimensional datasets, and especially when the answers need to be computed efficiently.

Furthermore, like most frameworks proposed in the literature for handling inconsistency in Description Logic knowledge bases, the characterization described in Defintion 4 starts from the assumption that the conflict set is computed and available beforehand. This does not constitute an issue in DL-Lite$_{\mathcal{R}}$ ontologies since the time complexity for computing the conflict set is polynomial in the ABox's size [10, 11]. However, when dealing with ontologies specified in Description Logic languages that are more expressive than DL-Lite$_{\mathcal{R}}$, there is no assurance that the conflict set can be enumerated in tractable time with respect to the ABox's size.

Two important aspects to take into consideration concern the size of the conflicts (i.e., the number of assertions that constitute each conflict) and the size of the conflict set (i.e., the number of conflicts in the ABox). One advantage of the characterization given in Definition 4 is that it does not place any restrictions on the number of assertions within the conflicts, so long as each conflict contains at least two assertions. Hence, the characterization remains valid in frameworks where the conflicts are not necessarily binary and may be composed of an arbitrary number of assertions. However, as illustrated in Example 4, considering conflicts that may involve any number of assertions entails that the size of the conflict set may be exponential in the ABox's size. This means that the cost of parsing all the elements of the conflict set in order to check the π-acceptance of some assertion can no longer be considered as negligible.

Clearly, the complexity of exhibiting all the conflicts has a direct impact on the computational properties of checking the π-acceptance of an assertion and of determining the set of all π-accepted assertions, i.e., the partially preordered possibilistic repair of the ABox. In order to mitigate these limitations, we propose an equivalent characterization for π-acceptance that does not involve comparisons between an assertion and all the elements of the conflict set. The idea is to rather make use of the consistency checking mechanism that is associated with the Description Logic language in which the ontology is encoded.

The first step is to build, for each assertion (φ_i, u_i) in $\mathcal{A}_{\rhd}$, the set $\Delta(\varphi_i)$ composed of all the other assertions of $\mathcal{A}_{\rhd}$ that are either more certain than φ_i or incomparable to φ_i, in terms of the strict partial order $\rhd$. Said differently, this set $\Delta(\varphi_i)$ is composed of all assertions of $\mathcal{A}_{\triangleright}$ that are not strictly less preferred than φ_i. Formally:

$$\Delta(\varphi_i) = \mathcal{A}_{\triangleright} \backslash \{\varphi_j | (\varphi_j, u_j) \in \mathcal{A}_{\triangleright} \text{ s.t. } u_i \triangleright u_j\}.$$

Note that we omit the weights associated to the assertions in the set $\Delta(\varphi_i)$ in order to be able to use the standard consistency checking mechanism underlying the ontological language.

Then, the new characterization that we propose in this paper determines the π-acceptance status of any given assertion φ_i by simply checking whether φ_i together with the subset $\Delta(\varphi_i)$ is consistent with respect to the axioms of the TBox. Most importantly, this characterization is equivalent to Definition 4.

Proposition 2. *An assertion (φ_i, u_i) of $\mathcal{A}_{\triangleright}$ is π-accepted in terms of $\triangleright$ if and only if $\{\varphi_i\} \cup \Delta(\varphi_i)$ is consistent with respect to $\mathcal{T}$.*

Proof. Consider $(\varphi_i, u_i) \in \mathcal{A}_{\triangleright}$.
(i) Assume that $\{\varphi_i\} \cup \Delta(\varphi_i)$ is consistent w.r.t. $\mathcal{T}$ but that (φ_i, u_i) is not π-accepted. According to Definition 4, this means that there is a conflict $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\triangleright})$ such that for each assertion $(\varphi_j, u_j) \in \mathcal{C}$, $\varphi_i \neq \varphi_j$, we have $\varphi_i \not\triangleright \varphi_j$. This means that for each element $(\varphi_j, u_j) \in \mathcal{C}$, we have either $\varphi_j \triangleright \varphi_i$ or $\varphi_j \bowtie \varphi_i$ (recall that $\triangleright$ is a strict partial order). This means that $\{\varphi_j | (\varphi_j, u_j) \in \mathcal{C}\} \subseteq \Delta(\varphi_i)$. This contradicts the fact that $\{\varphi_i\} \cup \Delta(\varphi_i)$ is consistent.

(ii) Assume that (φ_i, u_i) is π-accepted but that $\{\varphi_i\} \cup \Delta(\varphi_i)$ is inconsistent w.r.t. $\mathcal{T}$. This means that there is a conflict $\mathcal{C}$ such that for each assertion $(\varphi_j, u_j) \in \mathcal{C}$, we have $\varphi_j \in \{\varphi_i\} \cup \Delta(\varphi_i)$. This also means that each element $(\varphi_j, u_j) \in \mathcal{C}$, $\varphi_i \neq \varphi_j$, is such that either $\varphi_j \triangleright \varphi_i$ or $\varphi_j \bowtie \varphi_i$. This contradicts the fact that (φ_i, u_i) is π-accepted.

□

An important property of the characterization introduced in Proposition 2 is that it runs in polynomial time in the ABox's size in any Description Logic language where the ABox's consistency can also be checked in polynomial time.

Proposition 3. *Checking whether an assertion (φ_i, u_i) of $\mathcal{A}_{\triangleright}$ is π-accepted can be achieved in polynomial time with respect to the size of $\mathcal{A}_{\triangleright}$ in DL-Lite$_{\mathcal{R}}$.*

Proof. The proof follows directly from the fact that consistency checking is tractable in DL-Lite$_{\mathcal{R}}$ ontologies.

□

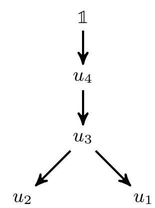
We illustrate both characterizations on our running example.

Example 3. Consider the KB $\mathcal{K}_{\triangleright} = \langle \mathcal{T}, \mathcal{A}_{\triangleright}, \mathbb{L} \rangle$ obtained from Example 1 by keeping the same TBox $\mathcal{T}$ and assigning symbolic weights to the assertions as depicted in Figure 1,(a)-(c). The compatible bases of $\mathcal{A}_{\triangleright}$ are given by Figure 1,(d)-(f), where the weights in the unit interval are shown on the left side of each subfigure. The inconsistency degrees are : $\mathsf{Inc}(\mathcal{WA}_1) = 0.4$, $\mathsf{Inc}(\mathcal{WA}_2) = 0.4$ and

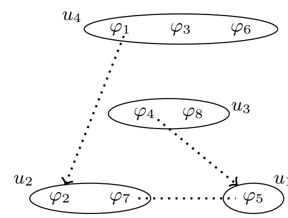
(a) The base $\mathcal{A}_{\rhd}$

Assertions	Weights
$\varphi_1 = \mathsf{Manual}(v_1)$	u_4
$\varphi_3 = \mathsf{Plugin}(v_1)$	u_4
$\varphi_6 = \mathsf{Electric}(v_3)$	u_4
$\varphi_4 = \mathsf{Thermal}(v_2)$	u_3
$\varphi_8 = \mathsf{Energy}(p)$	u_3
$\varphi_7 = \mathsf{useFuel}(v_2, p)$	u_2
$\varphi_2 = \mathsf{Electric}(v_1)$	u_2
$\varphi_5 = \mathsf{Plugin}(v_2)$	u_1

(b) The relation $\rhd$ over U



(c) The conflicts of $\mathcal{A}_{\rhd}$



(d) The base $\mathcal{WA}_1$ (e) The base $\mathcal{WA}_2$ (f) The base $\mathcal{WA}_3$

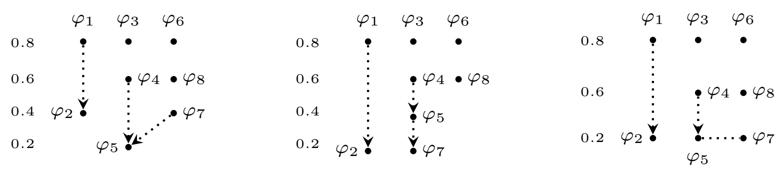


Fig. 1: The base $\mathcal{A}_{\rhd}$ and its compatible bases $\mathcal{WA}_1$, $\mathcal{WA}_2$ and $\mathcal{WA}_3$. The conflicts are represented with dotted lines. Arrow heads represent strict preference.

$\mathsf{Inc}(\mathcal{WA}_3) = 0.2$. The associated possibilistic repairs are by coincidence all the same and they are equal to their intersection. Hence the partially preordered repair corresponds to :

$$\pi(\mathcal{A}_{\rhd}) = \{\mathsf{Manual}(v_1), \mathsf{Plugin}(v_1), \mathsf{Electric}(v_3), \mathsf{Thermal}(v_2), \mathsf{Energy}(p)\}.$$

First characterization It is easy to see that the assertions $(\mathsf{Manual}(v_1), u_4)$, $(\mathsf{Plugin}(v_1), u_4)$, $(\mathsf{Electric}(v_3), u_4)$, $(\mathsf{Thermal}(v_2), u_3)$, $(\mathsf{Energy}(p), u_3)$ are strictly preferred to at least one assertion of each conflict (see Figures 1(a), 1(b) and 1(c)). Hence, according to Definition 4, these assertions are all π-accepted.

Second characterization For each assertion in $\mathcal{A}_{\rhd}$, we determine the corresponding Δ set (see Figures 1(a), 1(b) and 1(c)).

- $\Delta(\varphi_1) = \Delta(\varphi_3) = \Delta(\varphi_6) = \emptyset$.
  Since $\mathcal{A}_{\rhd}$ does not contain self-contradictory assertions, each of the three assertions together with the empty set is consistent w.r.t. $\mathcal{T}$. Hence, the assertions (φ_1, u_4), (φ_3, u_4) and (φ_6, u_4) are all π-accepted.
- $\Delta(\varphi_4) = \Delta(\varphi_8) = \{\varphi_1, \varphi_3, \varphi_6\}$.
  Each of the assertions (φ_4, u_3) and (φ_8, u_3) together with the Δ set is consistent w.r.t. $\mathcal{T}$. Hence, both assertions are π-accepted.
- $\Delta(\varphi_2) = \Delta(\varphi_7) = \{\varphi_1, \varphi_3, \varphi_6, \varphi_4, \varphi_8, \varphi_5\}$.
  None of (φ_2, u_2) or (φ_7, u_2) is π-accepted because both assertions are unsatisfiable with the corresponding Δ set.
- $\Delta(\varphi_5) = \{\varphi_1, \varphi_3, \varphi_6, \varphi_4, \varphi_8, \varphi_2, \varphi_7\}$. This Δ set is unsatisfiable, hence the assertion (φ_5, u_1) is not π-accepted.

Hence, the π-accepted assertions using the new characterization are given by the following five assertions: $(\mathsf{Manual}(v_1), u_4)$, $(\mathsf{Plugin}(v_1), u_4)$, $(\mathsf{Electric}(v_3), u_4)$, $(\mathsf{Thermal}(v_2), u_3)$ and $(\mathsf{Energy}(p), u_3)$.

□

This example illustrates that both characterizations return the same π-accepted assertions for $\mathcal{A}_{\triangleright}$, which also correspond to the assertions of the repair $\pi(\mathcal{A}_{\triangleright})$ where the symbolic weights are omitted.

In DL-Lite$_{\mathcal{R}}$ ontologies, $|\mathsf{Cf}(\mathcal{A}_{\triangleright})| = \mathcal{O}(n^2)$, where $n = |\mathcal{A}_{\triangleright}|$. Hence, computing the set of π-accepted assertions using the first characterization requires $\mathcal{O}(n^3)$ steps. Indeed, for each assertion of $\mathcal{A}_{\triangleright}$, the method parses all the conflict pairs in $\mathsf{Cf}(\mathcal{A}_{\triangleright})$ and compares it with both assertions of each pair (in the worst case). In contrast, using the second characterization requires n consistency checks to identify the π-accepted assertions of $\mathcal{A}_{\triangleright}$.

4.2 The case of non-binary conflicts

Our aim in this paper is to compute the set of π-accepted assertions. When dealing with an inconsistent knowledge base, it is desirable to have efficient procedures that allow to:

Task 1 check whether the knowledge base is consistent;
Task 2 compute the set of all the conflicts; and,
Task 3 check whether an assertion is satisfiable with the set of assertions that are either strictly more certain or incomparable.

It is clear that if there exists an efficient procedure that achieves Task 3 in polynomial time, then our method can be extended to Description Logic languages that are richer than DL-Lite$_{\mathcal{R}}$ and in which the conflicts can be of arbitrary size (i.e., not necessarily composed of two assertions like in DL-Lite$_{\mathcal{R}}$). So, checking whether an assertion is π-accepted can be done efficiently.

Note that this is not the case in the original method of calculating π-accepted assertions proposed in [3] for DL-Lite$_{\mathcal{R}}$ KBs, which is based on Task 2, i.e., it requires the preliminary computation of all the conflicts. However, even if an expressive language has an algorithm for exhibiting all the conflicts in polynomial time in the ABox's size, the time itself may be large. The following simple example illustrates this observation.

Example 4. We are interested in describing the integrity constraints restricting user access to machines (computers) in a large company. We first describe the vocabulary of the language, we then describe the knowledge base.

Suppose that a large company is made up of a number m of departments, simply numbered as $\{d_1, \dots, d_m\}$. Each department d_i has a number t of machines, denoted by $\{c_{i1}, \dots, c_{it}\}$ (we assume that all the departments have the same number of machines).

Suppose that we have a set of m role names (one per department) denoted by $\{\mathsf{Access}_1, \dots, \mathsf{Access}_m\}$. Intuitively, the role $\mathsf{Access}_i(x, c)$ means that in the department d_i, the user x has access to the machine c.

For the sake of simplicity, we are only interested in the permissions granted to a particular employee, for instance the head of the company, denoted by h. Our set of constants is therefore composed of:

$$\{h\} \cup (\bigcup_{i=1,\ldots,m} \{c_{i1}, \ldots, c_{it}\}).$$

We further assume that the TBox $\mathcal{T}$ is composed of a single negative axiom:

$$\mathcal{T} = \{\exists \mathsf{Access}_1 \sqcap \exists \mathsf{Access}_2 \sqcap \ldots \sqcap \exists \mathsf{Access}_m \sqsubseteq \bot\}.$$

This negative axiom means that there is no user that has access to at least one machine in each department.

The following ABox describes the access permissions granted to the user h, the head of the company. We assume that she has access to all the machines in the company, regardless of the department in which they are located.

$$\begin{aligned}\mathcal{A} = &\{\mathsf{Access}_1(h, c_{11}), \ldots, \mathsf{Access}_1(h, c_{1t})\}\\ &\cup \{\mathsf{Access}_2(h, c_{21}), \ldots, \mathsf{Access}_2(h, c_{2t})\}\\ &\cup \ldots\\ &\cup \{\mathsf{Access}_m(h, c_{m1}), \ldots, \mathsf{Access}_m(h, c_{mt})\}.\end{aligned}$$

We assume a partition of the partially preordered ABox $\mathcal{A} = \langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \rangle$, such that each sub-base $\mathcal{A}_i$ contains the assertions concerning access permissions to the machines of the department d_i:

$$\mathcal{A}_i = \{\mathsf{Access}_i(h, c_{i1}), \ldots, \mathsf{Access}_i(h, c_{it})\}.$$

The preference relation between the sub-bases $\mathcal{A}_i$, $i = 1, \ldots, n$, is defined as:

- for all j, $j = 2, \ldots, n$, we have: $\mathcal{A}_1 \rhd \mathcal{A}_j$, and
- for all k, $k = 2, \ldots, n$ such that $k \neq j$, we have: $\mathcal{A}_j \bowtie \mathcal{A}_k$.

Note that the assertions belonging to same sub-base $\mathcal{A}_i$ are equally certain.
One can easily check that the size of the ABox $\mathcal{A}$ is equal to $t * m$ assertions. Moreover, each m-uple:

$$\mathcal{C} = \{\mathsf{Access}_1(h, c_{j_1}), \mathsf{Access}_2(h, c_{j_2}), \ldots, \mathsf{Access}_m(h, c_{j_m})\}$$

obtained by taking exactly one assertion from each role is a conflict. Therefore, the conflict set is:

$$\begin{aligned}\mathsf{Cf}(\mathcal{A}) = &\{\mathsf{Access}_1(h, c_{11}), \ldots, \mathsf{Access}_1(h, c_{1t})\}\\ &\times \{\mathsf{Access}_2(h, c_{21}), \ldots, \mathsf{Access}_2(h, c_{2t})\}\\ &\times \vdots\\ &\{\mathsf{Access}_m(h, c_{m1}), \ldots, \mathsf{Access}_m(h, c_{mt}\}.\end{aligned}$$

where the operator $\times$ denotes the Cartesian product of sets. Hence, the size of the conflict set $\mathsf{Cf}(\mathcal{A})$ is:

$$|\mathsf{Cf}(\mathcal{A})| = \mathcal{O}(t^m).$$

The number of conflicts in the ABox is exponential. This implies that even with reasonable numbers, for instance $m = 10$ and $t = 200$, it is clearly not possible to exhibit all the conflicts in the knowledge base. Hence, the original method for determining π-acceptance [3] is impractical for such a scenario.

□

Through Example 4, we argue that even for a Description Logic language that allows to compute the conflict set in polynomial time, it is not sufficient to apply the method for computing π-accepted assertions given in [3]. Indeed, the size of the conflict set also needs to be polynomial in the ABox's size.

Note that if for a given Description Logic language, we have an algorithm that is polynomial in time and space to calculate the conflict set, then checking the consistency is also tractable. However, the converse does not hold. Indeed, assume that there is some language (or at least, special cases of knowledge bases) in which checking consistency is tractable but the number of conflicts in the ABox is not polynomial with respect to the ABox's size. Therefore, the characterization introduced in this paper (Proposition 2) is more efficient than the one introduced in [3] (Definition 4).

In the following example, we apply Proposition 2 in order to determine the π-acceptance of assertions, even in knowledge bases where the size of the conflict set is exponential.

Example 5. We continue Example 4 and illustrate the new characterization with two examples of queries in order to check whether the assertions $\mathsf{Access}_1(h, c_{11})$ and $\mathsf{Access}_3(h, c_{31})$ are π-accepted.

We start with the assertion $\mathsf{Access}_1(h, c_{11})$ and determine $\Delta(\mathsf{Access}_1(h, c_{11}))$. One can determine that:

$$\Delta(\mathsf{Access}_1(h, c_{11})) = \mathcal{A}_1.$$

Indeed, the assertions of $\mathcal{A}_j$ with $j > 1$ are all strictly less certain than any assertion of $\mathcal{A}_1$.
One can easily check that $\Delta(\mathsf{Access}_1(h, c_{11}))$ is consistent. This means that the assertion $\mathsf{Access}_1(h, c_{11})$ is satisfiable with $\Delta(\mathsf{Access}_1(h, c_{11}))$. Therefore, we conclude that $\mathsf{Access}_1(h, c_{11})$ is π-accepted.

Regarding the assertion $\mathsf{Access}_3(h, c_{31})$, one can determine that:

$$\Delta(\mathsf{Access}_3(h, c_{31})) = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_m.$$

Indeed, for any assertion φ of $\mathcal{A}$, either $\mathsf{Access}_3(h, c_{31})$ is incomparable to φ (if $\varphi \in \mathcal{A}_j$ with $j > 1$), or $\mathsf{Access}_3(h, c_{31})$ is strictly less certain than φ (if $\varphi \in \mathcal{A}_1$). Then, one can also check that each assertion of the ABox $\mathcal{A}$ does not belong to at least one repair of the ABox. Therefore, the assertion $\mathsf{Access}_3(h, c_{31})$ is not satisfiable with the set $\Delta(\mathsf{Access}_3(h, c_{31}))$. Hence, we conclude that $\mathsf{Access}_3(h, c_{31})$ is not π-accepted.

□

5 Conclusion

Handling inconsistency in knowledge bases is an ongoing research topic meeting many applications, such as query answering from ontologies, where the focus is on defining efficient methods and procedures. In this paper, we address this issue by proposing a new characterization for checking (the so-called) π-acceptance in inconsistent and partially preordered ontologies. This characterization is an improvement over the original one. Indeed, the original characterization is based on the conflict set associated with the partially preordered knowledge base, so it assumes that the conflict set is readily available. The new characterization is rather based on a consistency check on a subset of the ABox, and does not require computing the conflict set.

Moreover, the new characterization can be applied to Description Logic languages that are more expressive than DL-Lite$_{\mathcal{R}}$, and remains efficient for any language where the consistency check can be performed efficiently.

In future work, we plan to investigate methods for producing more productive repairs and that are also tractable. One option is to consider the positive deductive closure but without incurring a computational explosion. We also plan to explore the case where the TBox's axioms may be uncertain and may be ignored or weakened as a means for resolving the inconsistency in the ABox.

Within the research project CROQUIS in collaboration with specialists in hydro-science, we plan to apply our methods of inconsistency handling to knowledge bases representing wastewater and stormwater networks in a large metropolis. The expert knowledge serves to complete the data, which is incomplete, imperfect, fragmented, outdated, multi-source, heterogeneous and uncertain. Data may consist of analog and digital maps of urban networks, geographical data, various types of images, intervention reports, and so on. It may be obtained from public organisations as well as private companies. Hence, it is virtually impossible for such a knowledge base to be consistent, making standard query answering tools inadequate. Moreover, given the sheer volume of data, a reduction in the computational complexity of repairing the inconsistency in the ABox is expected to have a direct positive impact on the experimental performance of our new characterization.

Acknowledgements: This research has received support from the ANR CROQUIS (Collecting, Representing, cOmpleting, merging, and Querying heterogeneous and UncertaIn waStewater and stormwater network data) project, grant ANR-21-CE23-0004 of the French research funding agency Agence Nationale de la Recherche (ANR).

References

1. Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation, and Applications (2007)
2. 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: Principles of Knowledge Representation and Reasoning (KR), Cape Town, South Africa. pp. 513–516 (2016)
3. Belabbes, S., Benferhat, S.: Computing a possibility theory repair for partially preordered inconsistent ontologies. IEEE Transactions on Fuzzy Systems pp. 1–10 (2021). https://doi.org/10.1109/TFUZZ.2021.3107776, https://doi.org/10.1109/TFUZZ.2021.3107776
4. Belabbes, S., Benferhat, S., Chomicki, J.: Handling inconsistency in partially preordered ontologies: the elect method. Journal of Logic and Computing **31**(5), 1356–1388 (2021)

5. Benferhat, S., Bouraoui, Z., Tabia, K.: How to select one preferred assertional-based repair from inconsistent and prioritized DL-Lite knowledge bases? In: International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina. pp. 1450–1456 (2015)
6. Benferhat, S., Bouraoui, Z.: Min-based possibilistic DL-Lite. Journal of Logic and Computation **27**(1), 261–297 (2017)
7. Bertossi, L.: Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
8. Bienvenu, M., Bourgaux, C.: Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR. pp. 141–151 (2020). https://doi.org/10.24963/kr.2020/15
9. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Query-driven repairing of inconsistent DL-Lite knowledge bases. In: IJCAI, New York, USA. pp. 957–964 (2016)
10. 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)
11. Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of DL-Lite knowledge bases. In: International Semantic Web Conference (1). pp. 112–128 (2010)
12. 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)
13. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, Bressanone/Brixen, Italy. pp. 103–117 (2010)
14. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Inconsistency handling in datalog+/- ontologies. In: ECAI, Montpellier, France. pp. 558–563 (2012)
15. 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/`
16. Trivela, D., Stoilos, G., Vassalos, V.: Querying expressive DL ontologies under the ICAR semantics. In: 31st DL workshop, Tempe, USA (2018)
17. 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)