Non-Objection Inference for Inconsistency-Tolerant Query Answering

Salem Benferhat
Univ Artois, France
benferhat@cril.fr

Zied Bouraoui
Aix-Marseille Univ, France
zied.bouraoui@univ.amu.fr

Madalina Croitoru
Univ Montpellier, France
croitoru@lirmm.fr

Odile Papini
Aix-Marseille Univ, France
odile.papini@univ.amu.fr

Karim Tabia
Univ Artois, France		tabia@cril.fr

Abstract

Repair based techniques are a standard way of dealing with inconsistency in the context of ontology based data access. We propose a novel non-objection inference relation (along with its variants) where a query is considered as valid if it follows from at least one repair and it is consistent with all the repairs. These inferences are strictly more productive than universal inference while preserving the consistency of its set of conclusions. We study the productivity and properties of the new inferences. We also give experimental results of the proposed non-objection inference.

1 Introduction

Inconsistency-Tolerant Query Answering is one of the challenging problems that received a lot of attention in recent years, e.g. [Lukasiewicz et al., 2015; Benferhat et al., 2015; Bienvenu et al., 2016; Wan et al., 2016; Baget et al., 2016]. Inconsistency may arise due to several reasons: merging, integration, revision, etc. In this paper, we place ourselves in the context of Ontology-Based Data Access (OBDA), where the ontological knowledge is assumed to be satisfiable and fully reliable. In such a setting, inconsistency comes from the data, i.e. occurs when some assertional facts contradict some constraints imposed by the ontological knowledge. As ontology language we use DL-Lite, a lightweight family of description logics, well suited for OBDA thanks to the so-called first-order rewritability property that ensures the efficient handling of queries [Calvanese et al., 2007].

Existing works in this area, e.g. [Lukasiewicz et al., 2012; Lembo et al., 2015], have studied different inconsistency-tolerant query answering (well-known as semantics) closely related to works on consistent query answering from inconsistent Databases (e.g. [Chomicki, 2007; Bertossi, 2011]) or inference from inconsistent propositional logic knowledge bases, e.g. [Baral et al., 1992; Benferhat et al., 1993; Nebel, 1994; Benferhat et al., 1997]. Ontology-based consistent query answering (AR-semantics) [Lembo et al., 2010] comes down first to compute the set of repairs (i.e. all maximally consistent subsets of facts consistent with the ontology) and then checking if a query can be entailed using these repairs. As shown in [Lembo et al., 2010; Bienvenu, 2012], the AR entailment (also called universal entailment) is a hard task (co-NP complete) for DL-Lite and also for other lightweight DLs [Rosati, 2011]. Several approximations of AR entailment have been proposed, e.g. IAR, ICAR, ICR [Bienvenu, 2012], k-sup and k-def [Bienvenu and Rosati, 2013], etc.

We propose a new inconsistency-tolerant inference relation, called non-objection inference, where a query is considered as valid if it is entailed by at least one repair and it is consistent with all the other repairs. The intuition behind is that no repair has an objection veto to the acceptance of the query. If query entailment from repairs is seen as posing a vote for, against or abstaining to a query then, in this semantics, some repairs are “voting” for a query (i.e. the query is entailed) and the rest of the repairs are not against (i.e the query body atoms together with the atoms in the repair are consistent with the terminology) then the query is accepted without any objections. In addition, two variants of non-objection inference based on a selection of repairs (that can be against a query) are also considered. Consider the axiom stating that the teachers cannot be taught. Consider people getting some courses as statements. Statement 1 (“Alice is teaching Bob”) and statement 2 (“Bob is teaching Celine”) are two inconsistent statements since Bob is both a teacher and being taught. Let us consider Q (“Is Celine a student?”). Statement 1 is not inconsistent with Q (it has no objection to the acceptance of Q). As statement 2 entails Q we can say that asking if Celine is a student is a non-objection conclusion that follows from the two inconsistent statements. This makes intuitively sense as the main inconsistency arose from Bob. Please note that existing works (excepting for existential semantics) return the empty set. However, existential semantics are returning non intuitive results as even the query Q (“Is Bob a student?”) is existentially accepted (with Bob being the cause of the inconsistency). With no-objection inference this is not accepted.

The main salient points of the newly introduced semantics is its efficiency (more efficient than universal entailment) and the fact that the set of conclusions it yields are consistent (unlike credulous entailment). Interestingly enough, we show that query answering with non-objection inference is achieved in polynomial time. The inferences are strictly more productive than universal inference while preserving the consistency of its set of conclusions. We study the productivity, properties and complexity of the new inferences.
2 Preliminaries

In this section, we recall DL-Lite logic (namely DL-LiteR), which underlies OWL2-QL ontology language designed for applications that use huge volumes of data) and we review the main existing inconsistency-tolerant inference relations proposed to deal with inconsistency.

**DL-Lite syntax and semantics.** The starting points are N_C, N_R and N_T, three pairwise disjoint sets of atomic concepts, atomic roles and individuals respectively. Let A ∈ N_C, P ∈ N_R, three connectors ‘*’, ‘@’ and ‘~’ are used to define complex concepts and roles. Basic concepts (resp. roles) B (resp. R) and complex concepts (resp. roles) C (resp. E) are defined in DL-Lite as follows:

\[
\begin{align*}
B & \rightarrow A \mid \exists R C \rightarrow B \mid \neg B \\
R & \rightarrow P \mid P^- E \rightarrow R \mid \neg R
\end{align*}
\]

where P^- represents the inverse of P. A DL-Lite knowledge base (KB) is a pair K = (T, A) where T is called the TBox and A is called the ABox. A TBox includes a finite set of inclusion axioms on concepts and on roles respectively of the form: B ⊑ C and R ⊑ E. The ABox contains a finite set of assertions (facts) of the form A(a) and P(a, b) where A ∈ N_C, P ∈ N_R and a, b ∈ N_T.

The semantics is given in terms of interpretations I = (Δ_I, τ_I) which consist of an non-empty domain Δ_I and an interpretation function τ_I that assigns to each a ∈ N_I an element a^I ∈ Δ_I, to each A ∈ N_C a subset A^I ⊆ Δ_I and to each P ∈ N_R an P^I ⊆ Δ_I × Δ_I. The function τ_I is extended in a straightforward way for complex concepts and roles, e.g (¬B)^I = Δ_I \ B^I, (B^I)^I = \{(y, x) ∈ Δ_I × Δ_I | (x, y) ∈ P^I\}. The interpretation I is said to be a model of a concept (resp. role) inclusion axiom, denoted by I |= B ⊑ C (resp. I |= R ⊑ E), if B^I ⊆ C^I (resp. R^I ⊆ E^I). Similarly, we say that I satisfies a concept (resp. role) assertion, denoted by I |= A(a) (resp. I |= P(a, b)), if a^I ∈ A^I (resp. (a^I, b^I) ∈ P^I).

An interpretation I is said to satisfy a KB K = (T, A), denoted I |= K, if I |= T and I |= A. Such interpretation is said to be a model of K. Lastly, a TBox T is said to be inconsistent if there exists a concept C.s.t \( \forall I : I |= T \), we have C^I = \emptyset. A DL-Lite KB K is said to be inconsistent if it does not admit any model.

**Query answering.** A query is a first-order logic formula, denoted q = \{\vec{x} | \phi(\vec{x})\}, where \( \vec{x} = (x_1, ..., x_n) \) are free variables, n is the arity of \( \phi \) and atoms of \( \phi(\vec{x}) \) are of the form A(t_i) or P(t_i, t_j) with A ∈ N_C and P ∈ N_R and t_i, t_j are terms, i.e. constants of N_T or variables. When \( \phi(\vec{x}) \) is of the form \( \exists \vec{y}. \phi(\vec{x}, \vec{y}) \) where \( \vec{y} \) are bound variables called existentially quantified variables, and \( \forall \vec{y}. \phi(\vec{x}, \vec{y}) \) is a conjunction of atoms of the form A(t_i) or P(t_i, t_j) with A ∈ N_C and P ∈ N_R and t_i, t_j are terms, then \( \phi(\vec{x}) \) is said to be a conjunctive query (CQ).

When n = 0, then q is called a boolean query (BQ). A BQ with no bound variables is called a ground query (GQ). Lastly, when q only contains one atom with no free variables, then it is called an instance query (IQ) (i.e. instance checking). For a BQ q, we have I |= q iff (\phi)^I = true and K |= q iff \( \forall I : I |= K \Rightarrow I |= q \). For a CQ q with free variables \( \vec{x} = (x_1, ..., x_n) \), a tuple of constants \( \bar{a} = (a_1, ..., a_n) \) is said to be the certain answer for q over K if the BQ (q|\bar{a}) obtained by replacing each variable \( x_i \) by \( a_i \) in q(\vec{x}), evaluates to true for every model of K. Hence CQ answering can be reduced to BQ answering. For more details, see [Artale et al., 2009].

**Inconsistency-tolerant consequence relations.** Coping with inconsistency can be done by first computing the set of inclusion-maximal consistent subsets, called repairs, then using them to perform inference. A repair is defined as follows:

**Definition 1.** Let K = (T, A) be a DL-Lite KB. A subset R \( \subseteq A \) is said to be a repair iff (i) (T, R) is consistent, and (ii) \( \forall R' : R' \subseteq R', (T, R') \) is inconsistent.

Let us denote by R(K) the set of repairs of K. The following definition formally introduces the most common inconsistency-tolerant inferences.

**Definition 2.** Let K = (T, A) be a DL-Lite KB. Let R(K) be the set of repairs of K and q be a query. Then:

- q is said to be a universal (or AR) consequence of K, denoted by K \( \models_{AR} q \) if \( \forall R \in R(K), (T, R) \models q \).
- q is said to be a credulous (or existential) consequence of K, denoted by K \( \models_{C} q \) if \( \exists R \in R(K), (T, R) \models q \).
- q is said to be a safe (or IAR) consequence of K, denoted by K \( \models_{IAR} q \) if \( \exists R, (T, R) \models q \).

The inference consequence relations given in Definition 2 are based on repairs computed using the initial ABox. However, one can define these entailments using closed ABox instead of the initial ABox (like the ICAR-entailment and the CAR-entailment [Lembo et al., 2010]) or using closed repairs instead of repairs themselves (like the IAR-entailment [Bienvenu, 2012]). The IAR-entailment is the most cautious one and the credulous entailment is the most productive one. The universal entailment (or AR-semantics) can be considered as safe since a query is accepted if it can be deduced from each repair using standard DL-Lite semantics. Query answering within the AR-semantics is co-NP-complete even for simple DL-Lite languages such as DL-Litecore [Lembo et al., 2010]. The credulous entailment is often considered as adventurous. It is so adventurous that the set of conclusions may be inconsistent w.r.t the TBox. If one views each repair as a consistent set of assertions provided by a distinct source of information, then credulous entailment just says whether there is a source or a reason, that supports a given query.

3 Non-Objection Inference and its Variants

In this section, we propose three inference relations more productive than the universal entailment but less adventurous than the credulous one: non-objection inference (no), cardinality-based non-objection (cno) and limited-based non-objection inference (lno).

3.1 Non-Objection Inference: no

Let us first define the statement of a query being consistent with a repair. Note that within DLs q is often restricted to a conjunctive query (CQ).

**Definition 3.** Let K = (T, A) be a DL-Lite KB, q be a CQ, and \( \bar{c} = (c_1, ..., c_n) \) be a tuple of constants. A repair R of K is said to be consistent with q(\bar{c}) w.r.t a TBox T iff there exists an interpretation I that satisfies both \( q(\bar{c}), R \) and T.
We can now introduce the non-objection (no for short) inference as follows:

**Definition 4.** Let \( K=(T,A) \) be a DL-Lite KB and let \( R(K) \) be the set of repairs of \( K \). Let \( q \) be a CQ and \( e=(e_1,\ldots,e_n) \) be a tuple of constants. A query \( q(e) \) is said to be a non-objection consequence of \( K \), denoted by \( K\models_{\text{no}} q(e) \), if (1) there exists at least one repair \( R \) of \( K \) such that \( \langle T,R\rangle \models q(e) \), (2) for each repair \( R' \in R(K) \), \( R' \) is consistent with \( q(e) \).

The term non-objection is understood in the sense that none of the repairs is against accepting query. If a repair is compared to an expert/source/voter, then a conclusion is accepted if at least one expert/source/voter supports the conclusion and none is against or has an objection.

**Example 1.** Let Alice, Bob, Celine be three individuals. Let TeachTo be a role, where TeachTo(x,y) means that the individual \( x \) is the teacher of the individual \( y \). Assume that we have the following inconsistent DL-Lite KB \( K=(T,A) \) where \( T=\{\exists \text{TeachTo}\land \neg \exists \text{TeachTo}\} \) and \( A=\{\text{TeachTo}(Alice,Bob),\text{TeachTo}(Bob,Celine)\} \). We have \( R_1=\{\text{TeachTo}(Alice,Bob)\} \) and \( R_2=\{\text{TeachTo}(Bob,Celine)\} \) two repairs. Consider the following query \( q_1=\exists y.\text{TeachTo}(y,Celine) \). Clearly, \( K\models_{\text{no}} q_1 \) since \( q_1 \) follows from \( R_2 \) and \( q_1 \) is consistent with \( R_1 \) and \( T \).

### 3.2 Adding Cardinality to no Inference: cno

We now investigate a new inference obtained by adding a cardinality criterion to no. We basically restrict the application of the no-inference to a selection of repairs. Here we consider the largest (in the sense of cardinality) repairs. This might make sense in certain practical settings. Consider a retail use case and two products, one liked by women and not by men and the other liked by men and not by women. If the number of women is greater than the number of men, then we might decide selling the first product. Let \( CR(K)=\{R:R \in R(K) \text{ and } |R'|>|R| \} \) be the set of largest repairs of \( K \) where \( |R| \) is the size of \( R \) (i.e. the number of facts in \( R \)). Then, the cardinality-based non-objection (cno for short) inference relation, is simply obtained from Definition 4 by only considering the largest repairs. More formally:

**Definition 5.** Let \( K=(T,A) \) be a DL-Lite KB, \( q \) be a CQ and \( e=(e_1,\ldots,e_n) \) be a tuple of constants. A query \( q(e) \) is said to be a cardinality-based non-objection consequence of \( K \), denoted by \( K\models_{\text{cno}} q(e) \), if (1) \( \exists R \in CR(K) \) s.t. \( \langle T,R\rangle \models q(e) \), and (2) \( \forall R' \in CR(K),\langle T,R'\rangle \) is consistent with \( q(e) \).

**Example 2.** Let \( K=(T,A) \) be a DL-Lite KB where \( T=\{B \subseteq -D, B \subseteq C, D \subseteq E, A \subseteq -B, A \subseteq -C, A \subseteq -D\} \) and \( A=\{A(a),B(a),C(a),D(a)\} \). We have \( R(K)=\{\{A(a)\},\{B(a),C(a)\},\{D(a),C(a)\}\} \) and \( CR(K)=\{R_1=\{A(a)\},R_2=\{D(a)\}\} \). Clearly \( K\models_{\text{cno}} E(a) \) since \( \langle T,R_2\rangle \models E(a) \) and \( R_1 \cup \{E(a)\} \) is consistent with \( T \).

The cno-inference relation is a completely new inconsistency-tolerant inference which has neither been proposed in a propositional logic setting nor in the description logics setting. Restricting the set of repairs aims to increase the productivity of the inference and the cardinality criterion is a natural way to select the set of the repairs used in inference. If rejecting an assertion from a repair can be seen as “the price to pay” then the cardinality criterion aims to reduce the price of the rejected assertions.

### 3.3 Limited Non-Objection Inference: lno

Last, we introduce an alternative inference more productive than the no and cno inference relations. The intuition is that when some answers to a query are obtained from a repair, then only repairs from \( CR(K) \) can make objections against the query answers. To comment on the practical relevance, recall the Obama administration that did not manage to pass certain motions (queries) as the elders (the repairs with the biggest cardinality) voted against.

**Definition 6.** Let \( K=(T,A) \) be a DL-Lite KB. Let \( q \) be a CQ and \( e=(e_1,\ldots,e_n) \) be a tuple of constants. Then \( q(e) \) is a limited non-objection consequence relation, denoted \( K\models_{\text{lno}} q(e) \), if (1) \( \exists R \in CR(K) \) s.t. \( \langle T,R\rangle \models q(e) \) and (2) \( \forall R' \in CR(K),\langle T,R'\rangle \) is consistent with \( q(e) \).

**Example 3.** Let \( K=(T,A) \) be a DL-Lite KB where \( T=\{A \subseteq E, A \subseteq \neg B, A \subseteq \neg D, C \subseteq B, D \subseteq \neg B, D \subseteq \neg E\} \) and \( A=\{A(a),B(a),C(a),D(a)\} \). We have \( R(K)=\{R_1=\{A(a)\},R_2=\{D(a)\}\} \). Clearly \( K\models_{\text{lno}} E(a) \) since \( \langle T,R_1\rangle \models E(a) \) and \( R_2 \cup \{E(a)\} \) is consistent with \( T \).

The difference between \( \models_{\text{no}} \) and \( \models_{\text{lno}} \) concerns item 2 of Definitions 4 and 6, where in \( \models_{\text{no}} \) all the repairs are considered while in \( \models_{\text{lno}} \) only the largest ones are taken into account. Similarly, the difference between \( \models_{\text{cno}} \) and \( \models_{\text{lno}} \) concerns item 1 of Definitions 5 and 6, where in \( \models_{\text{cno}} \) all the repairs are used while in \( \models_{\text{cno}} \) only the largest ones are used. Again if we view repairs as experts/sources/voters then a limited non-objection conclusion is considered as accepted if there is an expert/source/voter that accepts the conclusion and none of the oldest/wise expert/source/voter is against.

### 3.4 Productivity of no, cno and lno Inferences

From a productivity point of view, no-entailment is between the universal entailment (i.e. \( \models_{\text{AR}} \)) and the credulous entailment. Figure 1 gives productivity relations between different inference relations described in this paper. The arrow “\( A \to B \)” means that each query which is considered as valid by the inference relation \( A \) is also considered as valid by \( B \). Note that the no-inference is different from the AR, IAR, ICR and CAR entailments. There is also a difference between no and the k-support and k-defeater inferences given in [Bien-
venu and Rosati, 2013). In particular, our approach does not depend on a parameter $k$.

## 4 Complexity Analysis

In this section we study the data complexity of query answering under the inference relations proposed in this paper. Recall that we assumed that the TBox $T$ is stable. Besides, we also recall that computing the set of conflicts can be done in polynomial time and that each conflict is a pair of assertions (a single fact if $T$ is incoherent) [Calvanese et al., 2010].

### 4.1 Data Complexity of no Inference

Let $K=(T, A)$ be a DL-Lite KB, $q(x)$ be a CQ and $\vec{c}$ be a tuple of constants of $K$. To establish the data complexity of no-inference, we focus on the following question. Given a KB $K$ and a CQ $q(x)$ does there exist a tuple of constants $\vec{c}'$ of $A$ such that $K \models_{\text{no}} q(\vec{c}')$? We show that the data complexity of the query answering under no semantics is polynomial. Firstly, let us consider ground query (GQ) of the form $q(\vec{x}) = \bigwedge_{i=1}^{n} X_i(\alpha_i)$ where $X_i(\alpha_i)$ is either of the form $A(t_i)$ or $P(t_i, t_j)$ with $A$ is a concept, $P$ is a role and $\alpha_i$ is either a term (if $X_i$ is a concept) or a pair of terms (if $X_i$ is a role). For each $X_i(\alpha_i)$ we define:

\[ Label_{X_i(\alpha_i)} = \{ e : e \in A \land (T \cup \{ D_i \leq \neg X_i \}, \{ e, D_i(\alpha_i) \}) \text{ is inconsistent} \]

where $D_i$ is a new concept (resp. a new role if $X_i$ is a role) associated with $X_i$. Intuitively, $Label_{X_i(\alpha_i)}$ represents the set of all supports for the instance $X_i(\alpha_i)$. The following proposition concerns the case where the query just concerns one instance (instance query).

**Proposition 1.** Let $K=(T, A)$ be a DL-Lite KB. Then $X_i(\alpha_i)$ is entailed by at least one repair $R$ of $K$ if and only if $Label_{X_i(\alpha_i)}$ is not empty.

**Proof.** Assume that from some repair $R$, we have $(T, R) \models X_i(\alpha_i)$. This means $(T \cup \{ D_i \leq \neg X_i \}, R \cup \{ D_i(\alpha_i) \})$ is inconsistent. Hence there exists a conflict in $(T \cup \{ D_i \leq \neg X_i \}, R \cup \{ D_i(\alpha_i) \})$. This conflict necessarily contains $D_i(\alpha_i)$ and an element $e$ of $R$ (since $(T, R)$ is consistent by definition of a repair). Hence $Label_{X_i(\alpha_i)}$ is not empty. For the converse, assume that $\exists R \models (T, R)\models X_i(\alpha_i)$. This means that $\forall R, (T \cup \{ D_i \leq \neg X_i \}, R \cup \{ D_i(\alpha_i) \})$ is consistent. Hence $\forall R, \exists e \in R$ s.t. $(T \cup \{ D_i \leq \neg X_i \}, \{ e, D_i(\alpha_i) \})$ is inconsistent. Since $\bigcup_{R \in \mathcal{R}(K)} R = A$, then $Label_{X_i(\alpha_i)} \neq \emptyset$.

Proposition 1 can be easily generalized for GQs as follows.

**Proposition 2.** Let $K=(T, A)$ be a DL-Lite KB and $q$ be a GQ. Then $q$ is entailed by at least one repair $R$, namely $(T, R)\models q$, if and only if there exists a tuple $(e_1, \ldots, e_n)$ of $A$ such that $\forall i, e_i \in Label_{X_i(\alpha_i)}$ and $(e_1, \ldots, e_n)$ is consistent with $T$.

**Proof.** If there is no $(e_1, \ldots, e_n) \in Label_{X_1(\alpha_1)} \times \ldots \times Label_{X_n(\alpha_n)}$ which is consistent, this means that whatever is the considered repair $R$, there is some $X_i(\alpha_i)$ such that $(T, R)$ is consistent with $\neg X_i(\alpha_i)$. Hence, there is no repair $R$ where $(T, R)\models q$. For the converse, assume that $(e_1, \ldots, e_n)$ is consistent with $T$. Then it is enough to build a repair $R$ which contains $(e_1, \ldots, e_n)$. Since $(T, R)$ is consistent but inconsistent with each $\neg X_i(\alpha_i)$, then $(T, R)\models q$.

If $q$ is a CQ, we first explicit the FOL-rewritability of DL-Lite. Indeed, $q$ can be expressed as a disjunction of queries: $Rw(q) = q_1 \lor \ldots \lor q_n$. $Rw(q)$ can be computed using PERFECTREF algorithm proposed in [Calvanese et al., 2007]). Each query $q_i$ in $Rw(q)$ implicitly represents different possible supports of $q$ where the size of these supports is equal to the size of $q_i$. Now, computing the set of all the supports of $q$ comes down to compute for each query $q_i$ in $Rw(q) = q_1 \lor \ldots \lor q_n$ the set of facts of $A$ that implies it. Then to check if there exists a repair that entails $q$ comes down to verify whether there is a consistent maximal support of $q$. More precisely, we first compute minimal supports for $q$ denoted by $S(q)$. Then $q$ is entailed from a repair iff $\exists S \in S(q)$ such that $(T, S)$ is consistent. Moreover the following result holds.

**Proposition 3.** Let $K=(T, A)$ be a DL-Lite KB and $q$ be a GQ. Assume that $(T, \{ X_1(\alpha_1), \ldots, X_n(\alpha_n) \})$ is consistent. Then $\forall R, (T, R) \cup \{ X_1(\alpha_1), \ldots, X_n(\alpha_n) \}$ is consistent if and only if there is no $e \in A$ and no $X_i(\alpha_i)$ s.t. $(T, \{ e, X_i(\alpha_i) \})$ is inconsistent.

**Proof.** Assume for some $X_i(\alpha_i)$ of the query, there exists an assertion fact $e$ from $A$, such that $(T, \{ e, X_i(\alpha_i) \})$ is inconsistent. Then let $R$ be a repair that contains $e$. Such $R$ is not necessarily unique but always exists. It is enough to start with $R = \{ e \}$ and add as much as possible elements from $A$ while preserving consistency. Then $(T, R) \cup \{ X_i(\alpha_i) \}$ is inconsistent, and hence $\forall R, (T, R) \cup \{ X_i(\alpha_i) \}$ is consistent” does not hold. For the converse, assume that there is no $e \in A$ and no $X_i(\alpha_i)$ such that $(T, \{ e, X_i(\alpha_i) \})$ is conflicting. Then whatever is the repair $R$, we have $(T, R) \cup \{ X_i(\alpha_i), \ldots, X_n(\alpha_n) \}$ is consistent (recall that by definition $(T, R)$ is consistent and $(T, \{ X_i(\alpha_i), \ldots, X_n(\alpha_n) \})$ is consistent by assumption).

Of course, in Proposition 3 if $(T, \{ X_1(\alpha_1), \ldots, X_n(\alpha_n) \})$ is inconsistent, then trivially $\forall R, (T, R) \cup \{ X_1(\alpha_1), \ldots, X_n(\alpha_n) \}$ is consistent” does not hold.

**Proposition 4.** Let $K=(T, A)$ be a DL-Lite KB and $q$ be a GQ. Then the data complexity of the query evaluation problem under non-extensional semantics is polynomial.

**Proof.** From Proposition 1, checking whether there exists a repair where an instance $X_i(\alpha_i)$ follows from $(T, R)$, can be achieved in polynomial time, since computing conflicts is done in polynomial time. A naive algorithm for implementing Proposition 2 is to progressively compute consistent elements of $Label_{X_1(\alpha_1)} \times \ldots \times Label_{X_n(\alpha_n)}$ by removing at each step each conflicting tuple $(e_1, \ldots, e_i)$ from $Label_{X_1(\alpha_1)} \times \ldots \times Label_{X_n(\alpha_n)}$. Since computing conflicts is done in polynomial time and since the size of the query is bounded, then checking whether $q$ follows from $(T, R)$, where $R$ is a repair of $K$, is achieved in polynomial time w.r.t the size of the ABox. Lastly, checking whether $q$ is consistent with each repair is also achieved in polynomial time. It is enough to first check whether

\footnote{the $e_i$'s are not required to be distinct.}
\(\{T, \{X_1(\alpha_1), \ldots, X_n(\alpha_n)\}\}\) is consistent (which is done in PTime). If it is the case, then one needs to compute the set of conflicts \(C\) of \(\langle T, A \cup \{X_1(\alpha_1), \ldots, X_n(\alpha_n)\}\rangle\) (which is done in polynomial time) and check whether there is a pair \((f, g)\) of \(C\) such that \(f \in A\) and \(g \in \{X_1(\alpha_1), \ldots, X_n(\alpha_n)\}\) \(\square\).

Note that considering coherent ontologies is not a restriction. In case where \(T\) is coherent then a preprocessing step is needed. If \(A\) is an empty concept (resp. \(P\) is an empty role), then one should remove from the ABox all assertions of the form \(A(\alpha)\) (resp. \(P(\alpha, \beta)\)), where \(a, b\) are individuals. Queries \(q\) where some \(X_i(\alpha_i)\) is an empty concept or role cannot be considered as no-consequences.

4.2 Data Complexity of no- and lno Inferences

Contrarily to no-inference, cno and lno inferences are hard. We first analyze the following complexity problem: CP1. Is the size of the largest repair of an inconsistent DL-Lite KB \(K\) is at least equal to \(k\)? To analyze the computational complexity of CP1, we will for instance use complexity results known in graph theory regarding the problem of Maximum Independent Sets (MIS) or stables sets. Let us recall \(k\)-MIS, the following decision problem: "Given a symmetric graph \(G\), is there an independent set (or stable set), denoted by \(IS\), of size at least equal to \(k\)?" The computational complexity of \(k\)-MIS is known to be \(NP\)-complete [Garey and Johnson, 1979]. We will denote by \(Prov-k\)-MIS a call to a prover that solves a \(k\)-MIS problem. The following gives transformations between graphs and DL-Lite KBs.

A transformation from an inconsistent DL-Lite KB to \(G\): Let \(\mathcal{K}\rangle=\langle T, A\rangle\) be an inconsistent KB. Let \(C(A)\) be the set of all conflicts in \(A\). Recall that when \(T\) is coherent, then all conflicts of \(C(A)\) are pairs of elements of \(A\) and are computed in PTime. We define a graph associated with \(K\) as follows:

1. The set of nodes is simply the set of assertions in \(A\) (one assertion = one different node), and (2) A non-oriented arc is drawn from \(f\) to \(g\) if there is \(f \in A, g \in A\) such that \((f, g)\) is a conflict of \((T, A)\). Then we have the following result:

**Proposition 5.** Let \(\mathcal{K}\rangle=\langle T, A\rangle\) be a DL-Lite KB, and \(G\) be its associated graph as it is defined above. Let \(R \subseteq A\) be a subset of \(A\) and \(G_R\) be the set of nodes associated to \(R\). Then \(R\) is a maximal consistent subset of \(A\) iff \(G_R\) is a MIS of \(G\).

**Proof.** Assume that \(R\) is a maximal consistent subset of \(A\) but \(G_R\) is not a MIS of \(G\). This means that there exists a node \(f\) (namely a fact of \(A\)) s.t. \(f \notin G_R\) and \(\forall g \in G_R\), there is no arc between \(f\) and \(g\). Said differently, there exists an assertion \(f \in A\) s.t. \(f \notin R\) and \(\forall g \in R\), there is no conflict \(C(A)\) of the form \((f, g)\). This means that \(R \cup \{g\}\) is consistent and this contradicts the fact that \(R\) is a maximally consistent subset of \(A\). Similarly, assume that \(G_R\) is a MIS of \(G\) and let us show that \(R\) (the subset of assertions present in \(G_R\)) is a maximally consistent subset of \(A\). Clearly, \(R\) is consistent, since \(\forall f \in R, \forall g \in R\), we have \((f, g) \notin C\) where \(C\) is a conflict (otherwise, there would be an arc between \(f\) and \(g\)). \(R\) is maximal, since \(\forall h \in R\) there is an arc between \(h\) and a node from \(G_R\). Hence there is a conflict between \(h\) and an element of \(R\), namely \(R \cup \{h\}\) is inconsistent. Hence \(R\) is maximal. \(\square\)

Let us now give the converse transformation: Let \(G\) be a non-oriented graph. The DL-Lite KB associated with \(G\) is defined as follows: (1) We associate to each node \(e\) a concept also denoted by \(e\) (two different nodes have two distinct associated concepts), (2) We use "\(\alpha\)" as the unique individual used in \(A\), (3) For each non-oriented arc \(e \rightarrow f\), we add \((e \subseteq f)\) to \(T\), namely the TBox associated with \(G\) is defined by: \(\langle T, A \cup \{e \in f\} \rangle \) is an arc of \(G\), and (4) The ABox is simply the set of nodes with the same individual "\(\alpha\)". namely \(A = \{e(\alpha): \alpha\} \) is an individual and \(e\) is a node of \(G\). The DL-Lite KB associated with a graph only involves one individual. It neither contains positive axioms nor relation symbols.

**Proposition 6.** Let \(G\) be a non-oriented graph, and \(\mathcal{K} = \langle T, A\rangle\) be the DL-Lite KB associated with \(G\), as it is defined above. Then, \(\forall e(\alpha) \in A, \forall f(\beta) \in A, \forall e(\alpha), f(\beta) \in C(A)\) iff there is a non-oriented arc between \(f\) and \(e\).

The proof is immediate. Since there is no relation symbols nor positive axioms, then the negative closure of \(T\) is simply \(T\). Besides, for each \(e \subseteq f\) of \(T\) (namely, an arc from \(G\) by construction), there exists exactly one conflict \((e(\alpha), f(\beta))\) from \(A\) since there is exactly one individual \(\alpha\). Using Propositions 5 and 6, the following proposition gives the complexity of computing the cardinality of the largest repair of \(A\).

**Proposition 7.** Let \(\mathcal{K} = \langle T, A\rangle\) be a DL-Lite KB. The complexity of computing the cardinality of the largest consistent subset of \(K\) is \(O(|\log_2(|A|)| + \text{Prov-k-MIS})\), namely there is an \(O(|\log_2(|A|)|)\) calls to a prover of a \(k\)-MIS problem.

It is enough to apply a dichotomic search between 1 and \(|A|\), and for each value \(1 < k < |A|\) we call a \(k\)-MIS prover. The following complexity problem will be helpful to analyze the complexity of \(|\text{I}_{\text{n}}|\) and \(|\text{I}_{\text{m}}|\).

**CP2.** Determine whether there exists a repair from \(\mathcal{CR}(K)\) which is consistent with a fact \(X(\alpha)\).

To implement CP2, we consider \(\mathcal{K}' = \langle T \cup \{D \sqsupseteq X\}, A \cup \{X(\alpha)\}\rangle\) where \(D\) is a new concept (or role) associated with \(X\). \(\mathcal{K}'\) is the result of adding to \(K\) the assumption that \(X(\alpha)\) is true. Let \(G_{K'}\) (resp. \(G_{K}\)) be the graph associated with \(K\) (resp. \(K'\)) using the transformation given in CP1 and \(IS_{K}\) be the size of the largest repair of \(\mathcal{CR}(K)\) (resp. the largest IS of \(G_{K}\) using CP1). Hence, we have:

**Proposition 8.** \(G_{K'}\) has an independent set of size \((|IS_{K}| + 1)\) iff \(\exists R \subseteq \mathcal{CR}(K)\) which is consistent with \(X(\alpha)\).

**Proof.** Note first that \(G_{K'}\) is obtained from \(G_{K}\) by first adding a new node \(D\) and drawing and arc between \(D\) and a node \(e\) of \(G_{K}\) iff \(\langle T \cup \{D \sqsupseteq X\}, \{D(\alpha), e\}\rangle\) is conflicting. If \(G_{K'}\) has an \(IS\) of size \((|IS_{K}| + 1)\) then \(IS\) necessarily contains the node \(D\) plus \(IS_{K}\) nodes from \(G_{K}\). Namely, \(IS\) contains \(D\) plus an \(IS\) of size \(|IS_{K}|\) from \(G_{K}\). Hence, there is a repair \(R'\) from \(K'\) which contains \(D(\alpha)\) and a repair \(R\) from \(K\). This means that \(\langle T \cup \{D \sqsupseteq X\}, R \cup \{D(\alpha)\}\rangle\) is consistent, hence \(\langle T \cup \{D \sqsupseteq X\}, R \cup \{D(\alpha), X(\alpha)\}\rangle\) is consistent and consequently \(\langle T \cup \{D \sqsupseteq X\}, R \cup \{X(\alpha)\}\rangle\) is consistent. \(\square\)

The above proposition can be easily generalized for ground queries. Now it remains to provide a procedure that checks if a GQ is consistent with some cardinality-based maximal
repair. This is quite straightforward. Let \( q \leftarrow \bigwedge_{i=1}^{n} X_i(\alpha_i) \) be a GQ. Let \( D \) be a new concept and \( R \) be a new role. Let \( T' = T \cup \{ D \subseteq X_i; \text{if } X_i \text{ is a concept} \} \cup \{ R \subseteq X_i; \text{if } X_i \text{ is a role} \} \) and \( A' = A \cup \{ \delta(\alpha_i); \text{if } X_i(\alpha_i) \in q \text{ and } X_i \text{ is a concept} \} \cup \{ R(\alpha_i); \text{if } X_i(\alpha_i) \in q \text{ and } X_i \text{ is a role} \} \). Then \( q \) is entailed from a repair \( R \in CR(K) \) iff there exists a MIS of size \( k_{\text{max}} + n \) in \( G_K \). Clearly, \( |cno| \) and \( |lno| \) are hard problems since their associated decision problems belong to \( \Delta^2_p \).

**Proposition 9.** Let \( K = (T, A) \) be a DL-Lite KB and \( q \) be a GQ. Then the data complexity of the query evaluation problem under cno and lno semantics is in \( \Delta^2_p \).

## 5 Rationality Properties and Experimental Evaluation

We now briefly analyze some properties of *no* inference and provide preliminary experimental evaluation results showing the scalability of no-inference. For logical properties, we focus on the situation where the considered conclusions are sets of assertions, which can also be seen as conjunctions of ground queries. Namely, given \( K = (T, A) \) a KB and \( A_1, A_2 \) two sets of facts s.t. \( (T, A_1) \) and \( (T, A_2) \) are consistent, we say that \( A_2 \) is a consequence of \( A_1 \) w.r.t. \( K \), denoted by \( A_1 \models_{s} A_2 \), if \( (T, A_1 \cup A_2) \models_{s} A_2 \). In the following, we recall the KLM rationality properties [Kraus et al., 1990] rephrased within our framework: *Reflexivity (R)* means that \( A \) has to be a consequence of \( A \). *Left Logical Equivalence (LLE)* expresses the fact that if \( A_1 \) and \( A_2 \) are equivalent then they have the same consequences. *Right Weakening (RW)* says that if a \( A \) is a consequence of \( A \), then \( A_2 \) is a consequence of \( A_2 \) then \( A_2 \) is a consequence of \( A_2 \). *Cut* expresses the fact that if \( A_2 \) is a consequence of \( A_2 \), then \( A_2 \) is a consequence of \( A_2 \). *Cut* expresses the fact that if \( A_2 \) is a consequence of \( A_2 \), then \( A_2 \) is a consequence of \( A_2 \). *Cautious Monotony (CM)* expresses that if \( A_2 \) and \( A_2 \) are consequences of \( A_2 \), then \( A_2 \) is a consequence of \( A_2 \). It can be shown that \( cno, lno \) and \( no \) inferences satisfy \( R, LLE, RW, CM \) but violate \( And \). Moreover, \( no \) and \( cno \) inferences satisfy CM while \( lno \) inference violates this property. Note that the difference between the language representing the KB and the one expressing the queries only impacts \( R, CM \) and \( Cut \) properties. In LLE \( A_2 \) can be replaced by a CQ. In RW \( A_2 \) (resp. \( A_2 \)) can be replaced by a CQ \( q_1 \) (resp. \( q_2 \)). In And \( A_2 \) (resp. \( A_2 \)) can be replaced by a CQ \( q_1 \) (resp. \( q_2 \)).

For experimental evaluation, we implemented a tool that checks whether a query \( q \) is a no-consequence of a DL-Lite KB \( K \). Note that we restrict the form of the queries to ground queries. As benchmark (available at https://code.google.com/p/combo-obda/), we considered the LUBM20 ontology (i.e. TBox), which corresponds to the DL-LiteR version of the original LUBM ontology [Lutz et al., 2013], and we used the Extended University Data Generator (EUDG) in order to generate the ABox assertions. This ontology only contains 208 positive inclusion axioms. We added 1296 negated axioms in order to allow for inconsistency (the list of these axioms can be found in [Bienvenu et al., 2014]). To efficiently compute conflicting elements (needed to check no-consequence), we evaluate over the ABox (stored as a DB using SQLite engine) queries expressed from the negated closure of the TBox to exhibit whether the ABox contains or not conflicting elements. In our case computing the negated closure of the ontology is done in 856906 milliseconds (ms). Table 1 gives for four ABoxes their size, and the time taken to check inconsistency. In each cell of column "IC", we first give the time taken to check inconsistency (the whole operation including the one of computing the negated closure) and then only the time taken to evaluate over each ABox queries from the negated closure. For instance, the time taken to check consistency for an ABox containing 2434 facts takes 857833ms with only 927ms for the queries evaluation operation. Note that without using query evaluation, looking for conflicting elements takes 563940ms. Similarly, Table 1 gives for four queries of different sizes, the time taken to check whether \( Q_i \) (with \( i \) is the size of the query) is a no-consequence (the whole operation) and then the time needed to compute \( Label_{X_i}(\alpha_i) \) and the time needed to check whether there exist elements of the ABox that conflict with the query. For example, the time taken to check whether \( Q_3 \) is a no-consequence of \( K = (T, A_1) \) is 1161829ms with 1226ms to compute \( Label_{X_i}(\alpha_i) \) and 1693ms to verify whether there exist elements of \( A_1 \) that conflict with \( Q_3 \). From Table 1, one can see that the most costly task is the one on computing the negated closure. Once done, checking inconsistency or no-consequence is efficiently done w.r.t the size of the ABox. One can also observe that checking no-inference slightly increases comparing to checking inconsistency. This is mainly due to the fact that we add new axioms in order to compute \( Label_{X_i}(\alpha_i) \). Of course this also depends on the size of the query (i.e. the number of added axioms). But once done, checking the existence of a repair that entails the query (by computing \( Label_{X_i}(\alpha_i) \)) and the existence of another repair against its entailment is done efficiently.

### 6 Conclusion

This paper introduced three new inconsistency-tolerant inferences. We gave productivity, properties and complexity results for the case where the ontology is expressed using DL-Lite. With respect to the state of the art, the salient point of our contribution lays in the fact that queries are handled, using no-inference, in polynomial time (contrarily to AR-entailment).
and the set of conclusions is always consistent (contrarily to $\exists$-entailment). This is not the case if one uses a propositional setting for instance. As a future work, one can investigate complexity results of no, ino and cno inferences for more expressive or other lightweight (for instance EL languages) ontology representation languages.

Acknowledgments

This work has been supported by the French National Research Agency, ASPIQ project ANR-12-BS02-0003. This work has also received support from the european project H2020 Marie Sklodowska-Curie Actions (MSCA) research and Innovation Staff Exchange (RISE): AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage.

References


