# Inconsistency Measures for Repair Semantics in OBDA

Bruno Yun, Srdjan Vesic, Madalina Croitoru, Pierre Bisquert

## HAL Id: lirmm-01892704
## https://hal-lirmm.ccsd.cnrs.fr/lirmm-01892704v1

Submitted on 10 Oct 2018

# Inconsistency Measures for Repair Semantics in OBDA

**Bruno Yun[1], Srdjan Vesic[2], Madalina Croitoru[1], Pierre Bisquert[3]**

[1] Univ. Montpellier, LIRMM

[2] CRIL - CNRS, Univ. Artois

[3] INRA, LIRMM

yun@lirmm.fr, vesic@cril.fr, croitoru@lirmm.fr pierre.bisquert@inra.fr

## Abstract

In this paper, we place ourselves in the Ontology Based Data Access (OBDA) setting and investigate reasoning with inconsistent existential rules knowledge bases. We use the notion of inconsistency measures on sets of facts to rank and filter repairs. We propose a generic framework to answer queries by using the best repairs and study productivity and properties of such a framework.

## 1 Introduction

In this paper we place ourselves in the OBDA setting and investigate query answering over a set of fact bases enriched by the ontology [Poggi *et al.*, 2008]. One of the main challenges of reasoning in OBDA applications is handling the inherent inconsistency that might occur amongst independently built data sources partially describing the same knowledge of interest [Benferhat *et al.*, 1997; Lukasiewicz *et al.*, 2015; Lembo *et al.*, 2015; Hecham *et al.*, 2017b]. Classically inconsistent tolerant semantics consider all maximally consistent subsets of a fact base (called repairs) that they manipulate using a modifier (expansion, splitting, etc.) and an inference strategy (intersection, universality, etc.) [Baget *et al.*, 2016a].

Using all repairs might be inappropriate for certain applications that would rather focus on particular sources of knowledge. For instance, when considering more reliable knowledge (i.e. sensor information, provenance data etc.) one could only consider repairs using mostly facts from such sources. Preferences on facts have been used for inconsistency tolerant reasoning in [Staworko *et al.*, 2012]. In that setting, the authors suppose that the preference order on the facts is given but unfortunately, this is not always the case. In such cases we propose to use the inconsistency of the elements of the knowledge base (KB) as an intrinsic preference on the facts. Such inherent preference on the facts (i.e. facts that are more or less guilty for the inconsistencies) generates a preference on the repairs that are containing these facts (i.e. repairs that contain more or less controversial facts). In this paper, we propose a framework that takes into consideration the inconsistency on the facts when using the repairs for query answering and restricts the set of repairs to the "best" with respect to inconsistency values. Since we consider a subset of repairs

we obtain more answers than classical inconsistency tolerant query answering.

In this paper, we characterise desirable properties of such frameworks like free facts entailment, syntax independence and reliability preservation. We also provide an implementation of our approach and discuss its performance. The salient point of this paper lies in it being the *first approach in the literature capable of ranking repairs using only the inherent structure of the KB*. This is a significant result as our approach is applicable on a large variety of domains without requiring additional preference information. Furthermore, we show the significance and the practical interest of our approach using the real data collected in the framework of the Pack4Fresh project for reducing food wastes. During this project, we collected data using an online poll from a set of professionals of the food industry, including wholesalers, quality managers, floorwalkers and warehouse managers, about food packagings and their characteristics. The framework was able to rank the repairs efficiently and the results were then analysed and evaluated by experts from the packaging industry.

## 2 Background Notions

The existential rules language [Calì *et al.*, 2009] is composed of formulae built with the usual quantifiers ($\exists, \forall$) and *only two connectors*: implication ($\rightarrow$) and conjunction ($\wedge$) and is composed of facts, rules and negative constraints. A *fact* is a ground atom of the form $p(t_1, \ldots, t_k)$ where $p$ is a predicate of arity $k$ and $t_i$, with $i \in [1, \ldots, k]$, constants. The set of all possible facts is denoted by $\mathcal{A}$. An existential *rule* is of the form $\forall \overrightarrow{X}, \overrightarrow{Y} \ H[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow \exists \overrightarrow{Z} C[\overrightarrow{Z}, \overrightarrow{X}]$ where $H$ and $C$ are existentially closed atoms or conjunctions of existentially closed atoms and $\overrightarrow{X}, \overrightarrow{Y}, \overrightarrow{Z}$ their respective vectors of variables. A *rule is applicable* on a set of facts $\mathcal{F}$ if and only if there exists a homomorphism [Baget *et al.*, 2016b] from $H$ to $\mathcal{F}$. Applying a rule to a set of facts (also called *chase*) consists of adding the set of atoms of $C$ to the facts according to the application homomorphism. Different *chase* mechanisms use different restrictions that prevent infinite redundancies [Baget *et al.*, 2011]. Here, we use recognisable classes of existential rules where the chase is guaranteed to stop [Baget *et al.*, 2011]. A *negative constraint* is a rule of the form $\forall \overrightarrow{X}, \overrightarrow{Y} \ H[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow \bot$ where $H$ is an existentially closed atom or conjunctions of existentially closed

atoms, $\overrightarrow{X}, \overrightarrow{Y}$, their respective vectors of variables and $\bot$ is *absurdum*. Please note that the number of atoms in $H$ is not bounded and that negative constraints generalise simple binary conflicts that can easily be translated between the two representations: $\neg p(\vec{X})$ is transformed into $np(\vec{X})$ and the negative constraint $p(\vec{X}) \wedge np(\vec{X}) \rightarrow \bot$ is added to the rules.

We say that $F_1$ *entails* $F_2$ denoted by $F_1 \models F_2$ if and only if there is a homomorphism from the set of atoms in $F_2$ to the set of atoms in $F_1$ where $F_1$ and $F_2$ are two existentially closed conjunctions of atoms. A conjunctive query is an existentially quantified conjunction of atoms. For readability, we restrict ourselves to boolean conjunctive queries, which are closed formulas (the framework and the obtained results can be extended to general conjunctive queries). The set of all possible boolean conjunctive queries is denoted by $\mathcal{Q}$.

**Definition 2.1.** A *knowledge base* $\mathcal{K}$ is a tuple $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}$ is a finite set of facts, $\mathcal{R}$ a set of existential rules and $\mathcal{N}$ a set of negative constraints.

The set of all KBs is denoted by $\mathcal{K}_{bs}$. The *saturation* of $\mathcal{F}$ by $\mathcal{R}$ is the set of all possible atoms and conjunctions of atoms that are entailed, after using all possible rule applications from $\mathcal{R}$ over $\mathcal{F}$ until a fixed point. The output of this process is called the closure and is denoted by $\mathcal{S}at_{\mathcal{R}}(\mathcal{F})$. A set $\mathcal{F}$ is said to be $\mathcal{R}$-*consistent* if no negative constraint hypothesis can be entailed, i.e. $\mathcal{S}at_{\mathcal{R}}(\mathcal{F}) \not\models \bot$. Otherwise, $\mathcal{F}$ is said to be $\mathcal{R}$-*inconsistent*. The *ground closure* of a set $\mathcal{F}$ by $\mathcal{R}$, denoted $\mathcal{C}\ell_{\mathcal{R}}(\mathcal{F})$, is different from the closure of the saturation since it only keeps ground atoms. Namely, $\mathcal{C}\ell_{\mathcal{R}}(\mathcal{F}) = \{\text{ground atoms } a \mid a \in \mathcal{S}at_{\mathcal{R}}(\mathcal{F})\}$. The notion of ground closure is widely used in the literature and especially when dealing with inconsistency tolerant inference since computing the intersection of closures is ambiguous. Indeed, atoms resulting from the application of rules with existential variables contain *null variables* which makes the intersection of two closures undefined.

We assume that the existential rules are skolemized [Marnette, 2009], namely $\forall \overrightarrow{X}, \overrightarrow{Y} \; H[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow \exists \overrightarrow{Z} C[\overrightarrow{Z}, \overrightarrow{X}]$ will be replaced with $\forall \overrightarrow{X}, \overrightarrow{Y} \; H[\overrightarrow{X}, \overrightarrow{Y}] \rightarrow C[f(X), \overrightarrow{X}]$ where $f$ is a new symbol function. Using the above mentioned methodology in our setting enables us to have the closure equal to the ground closure and guarantees a finite saturation.

We recall the notion of minimal inconsistent sets and maximal consistent sets of a KB [Reiter, 1987].

**Definition 2.2.** Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB, we say that:

- Minimal $\mathcal{R}$-*inconsistent* sets of $\mathcal{K}$ are defined as $MI(\mathcal{K}) = \{X \subseteq \mathcal{F} \mid \mathcal{S}at_{\mathcal{R}}(X) \models \bot \text{ and } \forall X' \subset X, \mathcal{S}at_{\mathcal{R}}(X') \not\models \bot\}$. A fact $f \in \mathcal{F}$ is called a *free fact* if and only if there is no $X \in MI(\mathcal{K})$ s.t. $f \in X$.

- Maximal for set inclusion $\mathcal{R}$-*consistent* sets (repairs) of $\mathcal{K}$ are defined as $\mathcal{C}_{omax}(\mathcal{K}) = \{X \subseteq \mathcal{F} \mid \mathcal{S}at_{\mathcal{R}}(X) \not\models \bot \text{ and for every } X \subset X', \mathcal{S}at_{\mathcal{R}}(X) \models \bot\}$

In the OBDA setting rules and constraints act as an ontology used to "access" different data sources. These sources are prone to inconsistencies. As it is common in the literature, we suppose that the rules are compatible with the negative constraints, i.e. the union of those two sets is satisfiable [Lembo

*et al.*, 2010]. Indeed, the ontology is believed to be reliable as it is the result of a robust construction by domain experts. However, as data can be heterogeneous due to merging and fusion, the data is assumed to be the source of inconsistency.

# 3 The Ranking-Based Inference Framework

In this section, we introduce the ranking-based inference framework (RIF) and its three main components: the inconsistency value, the lifting function and the inconsistency tolerant inference. It is similar to the work in argumentation by [Konieczny *et al.*, 2015], where only the best extensions are used for reasoning. The section is organised as follows: in Section 3.1, we recall the notion of Drastic and MI Shapley inconsistency values, in Section 3.2, we give examples of lifting functions and in Section 3.3, we show how inconsistency tolerant inferences are modified in order to be used in the framework.

An *inconsistency measure* according to [Grant and Hunter, 2011] is a function that, given a KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, associates a number to each set of facts.

**Definition 3.1.** An inconsistency measure is a function $I : \mathcal{K}_{bs} \times 2^{\mathcal{A}} \rightarrow \mathbb{R}$ such that for every $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N}) \in \mathcal{K}_{bs}$ and $C \in 2^{\mathcal{A}}$, $I(\mathcal{K}, C) = 0$ if and only if $C$ is $\mathcal{R}$-consistent, $I(\mathcal{K}, C \cup C') \geq I(C)$ and if $\alpha$ is a free fact of $\mathcal{K}$, then $I(\mathcal{K}, \mathcal{F}) = I(\mathcal{K}, \mathcal{F} \backslash \{\alpha\})$. For readability purposes, we will use the notation $I^{\mathcal{K}}(C)$ instead of $I(\mathcal{K}, C)$ and $I(C)$ if the working KB $\mathcal{K}$ is obvious.

An *inconsistency value* is a function that associates a number to each fact of a KB $\mathcal{K}$. Many inconsistency values were defined by [Hunter and Konieczny, 2010] using existing inconsistency measures and the Shapley value from coalitional game theory. We introduce a framework that makes use of these inconsistency values together with a lifting function and an inconsistency tolerant inference relation to improve the productivity of query answering for an inconsistent KB.

Our framework is based on three layers. First, an *inconsistency value* is used to calculate the score of each fact of $\mathcal{K}$. We previously mentioned Shapley inconsistency values, but any function returning a score for each fact of $\mathcal{K}$ can be used.

**Definition 3.2.** An inconsistency value is a function $\mathsf{S} : \mathcal{K}_{bs} \times \mathcal{A} \rightarrow \mathbb{R}$. Let $\succeq_{\mathcal{A}}^{\mathsf{S}}$ be the total, reflexive and transitive binary order on $\mathcal{A}$ with respect to $\mathcal{K}$ and $\mathsf{S}$ defined as: for every $a, b \in \mathcal{A}, a \succeq_{\mathcal{A}}^{\mathsf{S}} b$ if and only if $\mathsf{S}(\mathcal{K}, a) \leq \mathsf{S}(\mathcal{K}, b)$. For readability purposes, we write $\mathsf{S}_a(\mathcal{K})$ instead of $\mathsf{S}(\mathcal{K}, a)$.

Second, we need a *lifting function*, i.e. a function that compares the set of repairs, based on the individual scores of facts with respect to an inconsistency value. A criterion of comparison would be to evaluate the "strongest" fact of each set. A generalisation of this criterion is the so-called *leximax* which, in the case where the best facts are equally strong, proceeds to compare the next best fact of each set. Please note that the set of all total, reflexive and transitive binary orders on $X$ is denoted by $\succeq_X$.

**Definition 3.3.** A lifting function is a function $\mathsf{L} : 2^{\mathcal{A}} \times \succeq_{\mathcal{A}} \rightarrow \succeq_{2^{\mathcal{A}}}$. For readability purposes, we use the notation $\mathsf{L}^{\succeq}(X)$ for $\mathsf{L}(X, \succeq)$. Furthermore, $(E, E') \in \mathsf{L}^{\succeq}(X)$ means that $E$ is better than or equal to $E'$.

Third, we use an *inconsistency tolerant inference relation* restricted to the best repairs sets ranked by the lifting function to answer the query. At this step, one can use the usual inconsistency tolerant inference relations such as AR, IAR, ICR or any of the modifier-based semantics of [Baget *et al.*, 2016a]

**Definition 3.4.** An inconsistency tolerant inference relation is a function $\models: \mathcal{K}_{bs} \times \mathcal{Q} \to \{True, False\}$.

Based on the previous notions, we define our framework.

**Definition 3.5.** A *ranking-based inference framework* (RIF) is a tuple $\mathcal{RIF} = (\mathtt{S}, \mathtt{L}, \models)$ where $\mathtt{S}$ is an inconsistency value, $\mathtt{L}$ is a lifting function and $\models$ is an inconsistency tolerant inference. The top result of $\mathcal{RIF} = (\mathtt{S}, \mathtt{L}, \models)$ on a KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is $\mathcal{O}(\mathcal{RIF}, \mathcal{K}) = \{E \in \mathcal{C}_{omax}(\mathcal{K}) \mid$ for all $E' \in \mathcal{C}_{omax}(\mathcal{K}), (E, E') \in \mathtt{L}^{\succeq^{\mathtt{s}}_{\mathcal{F}}}(\mathcal{F})\}$.

## 3.1 $\mathcal{RIF}$ Inconsistency Value

An inconsistency value is a function that associates a value to each fact of the KB. This value is supposed to be higher the more a fact is conflicting with the other facts. In this paper, we make the choice to focus on the Shapley inconsistency value introduced by [Hunter and Konieczny, 2010] because it possesses many desirable properties as will be shown in Proposition 3.1 below. The Shapley inconsistency value uses notions from game theory to measure the responsibility of each fact to the overall inconsistency of the KB.

**Definition 3.6.** Let $I$ be an inconsistency measure, $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ a KB and $f \in \mathcal{F}$, the Shapley inconsistency value corresponding to $I$, noted $\mathtt{S}^I$ is defined as:

$$\mathtt{S}^I_f(\mathcal{K}) = \sum_{C \subseteq \mathcal{F}} \frac{(|C| - 1)!(|\mathcal{F}| - |C|)!}{|\mathcal{F}|!}(I^{\mathcal{K}}(C) - I^{\mathcal{K}}(C \setminus \{f\}))$$

Note that if one considers $\mathcal{F}$ as the vector $(f_1, f_2, \ldots, f_n)$, then $\mathtt{S}^I(\mathcal{K})$ is the vector of corresponding Shapley inconsistency values, i.e. $\mathtt{S}^I(\mathcal{K}) = (\mathtt{S}^I_{f_1}(\mathcal{K}), \mathtt{S}^I_{f_2}(\mathcal{K}), \ldots, \mathtt{S}^I_{f_n}(\mathcal{K}))$.

Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB, the inconsistency values investigated by this paper are the following:

- *The drastic Shapley inconsistency value* is computed by using the following inconsistency measure:

$$I^{\mathcal{K}}_d(X) = \begin{cases} 0 & \text{if } X \text{ is } \mathcal{R}\text{-consistent with respect to } \mathcal{K} \\ 1 & \text{otherwise} \end{cases}$$

- *The MI Shapley Inconsistency value* is computed by using the following inconsistency measure:

$$I^{\mathcal{K}}_{MI}(X) = |MI((X, \mathcal{R}, \mathcal{N}))|$$

We now show that every Shapley inconsistency value satisfies Distribution, Symmetry and Minimality. The result and its proof are similar to that of [Hunter and Konieczny, 2010].

**Proposition 3.1.** Let $I$ be an arbitrary inconsistency measure and $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ a KB, the Shapley inconsistency value corresponding to $I$ satisfies:

- **(Distribution)** $\sum\limits_{a \in \mathcal{F}} \mathtt{S}^I_a(\mathcal{K}) = I^{\mathcal{K}}(\mathcal{F})$

- **(Symmetry)** If $a, b \in \mathcal{F}$ such that for all $X' \subseteq \mathcal{F}, a, b \notin X'$ we have $I^{\mathcal{K}}(X' \cup \{a\}) = I^{\mathcal{K}}(X' \cup \{b\})$ then it holds that $\mathtt{S}^I_a(\mathcal{K}) = \mathtt{S}^I_b(\mathcal{K})$

- **(Minimality)** If $a$ is a free fact of $\mathcal{K}$ then $\mathtt{S}^I_a(\mathcal{K}) = 0$

In Example 3.1, we show how the MI and drastic inconsistent values are computed from a simple KB.

**Example 3.1.** Let us consider the KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where $\mathcal{F} = \{d(M), a(M), c(M), b(M, S)\}$, $\mathcal{R} = \{\forall x(c(x) \land b(x, S) \to u(x))\}$ and $\mathcal{N} = \{\forall x (d(x) \land a(x) \to \bot), \forall x (u(x) \land d(x) \to \bot), \forall x (u(x) \land a(x) \to \bot)\}$. We have that $\mathtt{S}^{I_d}_{d(M)}(\mathcal{K}) = 4 \times \frac{1}{12} = \frac{1}{3}$ and $\mathtt{S}^{I_{MI}}_{d(M)}(\mathcal{K}) = 4 \times \frac{1}{12} + \frac{1}{4} \times 2 = \frac{5}{6}$. Thus, here we have that $\mathcal{F} = \{d(M), a(M), c(M), b(M, S)\}$ and $\mathtt{S}^{I_d}(\mathcal{K}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6})$ and $\mathtt{S}^{I_{MI}}(\mathcal{K}) = (\frac{5}{6}, \frac{5}{6}, \frac{2}{3}, \frac{2}{3})$. Since the higher score means being more inconsistent, the resulting ranking on facts, for both inconsistency values, is $c(M) \sim b(M, S) \succ d(M) \sim a(M)$.

We recall that we work with the total, reflexive and transitive ranking $\succeq^{\mathtt{s}}_{\mathcal{F}}$ on $\mathcal{F}$ extracted from the inconsistency value.

## 3.2 $\mathcal{RIF}$ Lifting

A *lifting function* $\mathtt{L}^{\succeq}$ compares sets of elements with respect to the ranking $\succeq$ and returns a total order on the sets.

Let us first introduce the $\mathtt{sort}$ relation that will be used in order to define the $\mathtt{L}^{\succeq}_{leximax}$ notion below. Given a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and a total, reflexive and transitive binary relation $\succeq$ on $X$, $\mathtt{sort}(X, \succeq)$ returns a sorted vector $(x_1, x_2, \ldots, x_n)$ such that for every $x_i, x_j$, we have that $x_i \succeq x_j$ if and only if $i \leq j$. The element at position $i$ in the vector $\mathtt{sort}(X, \succeq)$ is denoted by $\mathtt{sort}_i(X, \succeq)$. Note that the returned vector is not necessarily unique due to the fact that some elements might be equivalent, i.e. $x_i \sim x_j$.

In this paper, we consider two possible instantiations of the lifting function $\mathtt{L}$. The $\mathtt{L}^{\succeq}_{max}$ lifting function compares the subsets with respect to their maximal elements and $\mathtt{L}^{\succeq}_{leximax}$ compares the elements after sorting them in decreasing order.

Let $Y$ be a set of elements, $\succeq$ a ranking on Y, $E, E' \in 2^Y$, $\mathtt{sort}(E, \succeq) = (x_1, x_2, \ldots, x_n)$ and $\mathtt{sort}(E', \succeq) = (x'_1, x'_2, \ldots, x'_m)$. We say that:

- $(E, E') \in \mathtt{L}^{\succeq}_{max}(Y)$ if and only if $max(E) \succeq max(E')$, where $max(X) = \mathtt{sort}_1(X, \succeq)$.

- $(E, E') \in \mathtt{L}^{\succeq}_{leximax}(Y)$ if and only if one of the following holds: (1) $m = n$ and for every $i \in \{1, \ldots, n\}, x_i \sim x'_i$, (2) there exists $i \in \{1, \ldots, min(m, n)\}$ s.t. $x_i \succ x'_i$ and for every $j \in \{1, \ldots, i - 1\}, x_j \sim x'_j$ or (3) $n > m$ and for every $i \in \{1, \ldots, m\}, x_i \sim x'_i$.

**Example 3.2** (Ex 3.1 cont.)**.** Let us consider the RIF $\mathcal{RIF} = (\mathtt{S}^{I_d}, \mathtt{L}^{\succeq}_{leximax}, \models)$. We have for every $R \in \mathcal{C}_{omax}(\mathcal{K}) \setminus \{c(M), b(M, S)\}$, it holds that $(\{c(M), b(M, S)\}, R) \in \mathtt{L}^{\succeq}_{leximax}(\mathcal{F})$ but $(R, \{c(M), b(M, S)\}) \notin \mathtt{L}^{\succeq}_{leximax}(\mathcal{F})$ and thus, $\mathcal{O}(\mathcal{RIF}, \mathcal{K}) = \{\{c(M), b(M, S)\}\}$

## 3.3 $\mathcal{RIF}$ Inference

Inconsistency-Tolerant Query Answering is a challenging problem that received a lot of attention recently. We recall that we place ourselves in the context of OBDA, where the ontology is assumed to be satisfiable and fully reliable. In the following, we recall some of the most well-known inconsistency tolerant inferences that have been proposed in the

literature. Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB and $q$ be a boolean conjunctive query. Then:

- $q$ is said to be AR entailed by $\mathcal{K}$ denoted by $\mathcal{K} \models_{AR} q$ if and only if for every $R \in \mathcal{C}_{omax}(\mathcal{K}), \mathcal{Cl}_{\mathcal{R}}(R) \models q$

- $q$ is said to be IAR entailed by $\mathcal{K}$ denoted by $\mathcal{K} \models_{IAR} q$ if and only if $\mathcal{Cl}_{\mathcal{R}} \left( \bigcap_{R \in \mathcal{C}_{omax}(\mathcal{K})} R \right) \models q$

- $q$ is said to be ICR entailed by $\mathcal{K}$ denoted by $\mathcal{K} \models_{ICR} q$ if and only if $\bigcap_{R \in \mathcal{C}_{omax}(\mathcal{K})} \mathcal{Cl}_{\mathcal{R}}(R) \models q$

**Example 3.3** (Ex 3.1 cont.). A query $q = \exists x(c(x))$ is not AR, IAR nor ICR entailed. Indeed, we cannot entail $q$ from the closure of all the repairs, the intersection of the closure of all the repairs nor the closure of the intersection of all repairs.

We propose here to reuse AR, IAR, ICR by restricting them to the output of a $\mathcal{RIF}$ instead of the whole set of repairs.

**Definition 3.7.** Let $x \in \{AR, IAR, ICR\}$. $\models_x^{\mathcal{RIF}}$ denote the restriction of $\models_x$ to the output of $\mathcal{RIF}$ instead of the whole set of repairs.

For instance, the restricted version of AR will be denoted by $\models_{AR}^{\mathcal{RIF}}$ and defined as $\mathcal{K} \models_{AR}^{\mathcal{RIF}} q$ if and only if for every $R \in \mathcal{O}(\mathcal{RIF}, \mathcal{K}), \mathcal{Cl}_{\mathcal{R}}(R) \models q$.

**Example 3.4** (Ex 3.2 cont.). Let $q$ be the query $\exists x(c(x))$. The query $q$ is AR, IAR and ICR entailed with respect to $\mathcal{RIF}$ since $\mathcal{O}(\mathcal{RIF}, \mathcal{K}) = \{\{c(M), b(M,S)\}\}$.

## 4 $\mathcal{RIF}$ Results

This section presents a characterisation of the framework in terms of properties and general productivity results. In Section 4.1, we show some desirable properties of the framework and how such component properties relate to framework properties. In Section 4.2, we show an algorithm for computing the output of the framework and its performance on a given set of data. In Section 4.3, we explicit the use of our framework on a real life scenario.

### 4.1 $\mathcal{RIF}$ Properties

In this section, we show that the desirable properties on the components can lead to desirable properties on the entire framework. First, we introduce desirable properties for **general inconsistency values**. The *Minimality* property states that a free fact should have the lowest score. The *Flawed* property conveys the idea that a non-free fact should have a strictly positive score. Lastly, the *Bottom Facts* property states that an $\mathcal{R}$-inconsistent fact should have the score 1.

> **Minimality** For any KB $\mathcal{K}$ and every free fact $a$ of $\mathcal{K}$, it holds that $\mathsf{S}_a^I(\mathcal{K}) = 0$.
>
> **Flawed** For any KB $\mathcal{K}$ and every non free fact $a$ of $\mathcal{K}$, it holds that $\mathsf{S}_f(\mathcal{K}) > 0$.
>
> **Bottom Facts** For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $f \in \mathcal{F}$ such that $\{f\}$ is $\mathcal{R}$-inconsistent, it holds that $\mathsf{S}_f(\mathcal{K}) = 1$.

The *R-Append* and *N-Append* properties are satisfied if the addition of a rule or a negative constraint to a KB cannot decrease the score of any fact.

> **R-Append** For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $r \notin \mathcal{R}$, if $\mathcal{K}' = (\mathcal{F}, \mathcal{R} \cup \{r\}, \mathcal{N})$ then for every $f \in \mathcal{F}, \mathsf{S}_f(\mathcal{K}) \leq \mathsf{S}_f(\mathcal{K}')$.
>
> **N-Append** For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $n \notin \mathcal{N}$, if $\mathcal{K}' = (\mathcal{F}, \mathcal{R}, \mathcal{N} \cup \{n\})$ then for every $f \in \mathcal{F}, \mathsf{S}_f(\mathcal{K}) \leq \mathsf{S}_f(\mathcal{K}')$.

The *Abstraction-I* property states that an inconsistency value should not rely on the names of constants or predicates.

> **Abstraction-I** For any KB $\mathcal{K}$ and any isomorphism[1] $\gamma$ such that $\gamma(\mathcal{K}) = \mathcal{K}'$, we have that for every $f \in \mathcal{F}$, $\mathsf{S}_f(\mathcal{K}) = \mathsf{S}_{\gamma(f)}(\mathcal{K}')$.

The *Cardinality-MI* property says that the score is based on minimal inconsistent subsets. Namely, if the score of a fact is strictly inferior to the score of an other fact, it means that the number of minimal inconsistent sets the first fact belongs to is strictly lower than the number of minimal inconsistent sets the second facts belongs to.

> **Cardinality-MI** For any KB $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $f, f' \in \mathcal{F}$ such that $\mathsf{S}_f(\mathcal{K}) < \mathsf{S}_{f'}(\mathcal{K})$ then $|\{X \in MI(\mathcal{K}) \mid f \in X\}| < |\{X \in MI(\mathcal{K}) \mid f' \in X\}|$

**Proposition 4.1.** It holds that:

- $\mathsf{S}^{I_{MI}}$ **satisfies** Minimality, Flawed, Abstraction-I, Bottom Facts and **does not satisfy** Cardinality-MI, R-Append and N-Append.

- $\mathsf{S}^{I_d}$ **satisfies** Minimality, Flawed, Abstraction-I and **does not satisfy** Cardinality-MI, R-Append, N-Append and Bottom Facts.

The lifting function uses a ranking on elements in order to provide a ranking on sets. We introduce some basic desirable **properties for lifting functions** below.

The *Data Sensitive* property ensures that element ranking is taken into account by the lifting function. The *Abstraction-L* states that the ranking returned by the lifting function should not be concerned with the names of the elements.

> **Data Sensitive** For any set of elements $Y$ such that $|Y| > 1$ and $E, E' \in 2^Y$, there exist two different total, reflexive and transitive binary relations $\succeq, \succeq'$ on Y such that $(E, E') \in \mathsf{L}^{\succeq}(Y)$ and $(E, E') \notin \mathsf{L}^{\succeq'}(Y)$.
>
> **Abstraction-L** For any set of elements $Y$, any total, reflexive and transitive binary relation $\succeq$ on $Y$, $E, E' \in 2^Y$ and isomorphism $\gamma$ such that $\gamma(Y) = Y'$ and $\gamma(\succeq) = \succeq'$, we have that $(E, E') \in \mathsf{L}^{\succeq}(Y)$ if and only if $(\gamma(E), \gamma(E')) \in \mathsf{L}^{\succeq'}(Y')$.

**Proposition 4.2.** $\mathsf{L}_{max}^{\succeq}, \mathsf{L}_{leximax}^{\succeq}$ satisfy Data Sensitive and Abstraction-L.

Please note that $\models_x^{\mathcal{RIF}}$ with $x \in \{AR, IAR, ICR\}$ satisfies the QCE, QCI, Cons, ConsS, ConsC properties from [Baget *et al.*, 2016b]. The next additional property states that if there is an isomorphism on KBs such that the output from the RIF is the same up to an isomorphism, then the answer to the query should remain the same too.

---

[1] The isomorphism $\gamma$ renames the predicates and the constants. We use an abuse of notation and apply $\gamma$ to sets of facts, sets of rules, negative constraints and to KBs (meaning we apply to all three).

**Abstraction-Q** For any KB $\mathcal{K}$, any RIF $\mathcal{RIF}$, any query $q$ and any isomorphism $\gamma$ such that $\gamma(\mathcal{K}) = \mathcal{K}'$ and $E \in \mathcal{O}(\mathcal{RIF}, \mathcal{K})$ if and only if $\gamma(E) \in \mathcal{O}(\mathcal{RIF}, \mathcal{K}')$, it holds that $\mathcal{K} \models^{\mathcal{RIF}} q$ if and only if $\gamma(\mathcal{K}) \models^{\mathcal{RIF}} \gamma(q)$.

We now introduce some **properties on the whole framework**. A desirable property is the entailment of free facts (the *Free* property). The *Supremacy* property states that if a fact is strictly less controversial than any other fact, then it should be entailed. *Non Entailment Ejection* states that if a fact is only entailed by the closure of a repair which is not ranked amongst the best repairs, then it will not be entailed by the KB. The *Abstraction* property states that if there is a renaming of the constants and predicates in a KB, the answers of our framework should remain unchanged.

**Free** For any KB $\mathcal{K}$ and any free fact $a$ of $\mathcal{K}$, it holds that $\mathcal{K} \models^{\mathcal{RIF}} a$.

**Supremacy** For any KB $\mathcal{K}$, if there exists $f \in \mathcal{F}$ such that $\mathtt{S}_f(\mathcal{K}) \neq 1$ and for every $f' \in \mathcal{F} \setminus \{f\}$, we have $\mathtt{S}_f(\mathcal{K}) < \mathtt{S}_{f'}(\mathcal{K})$ then $\mathcal{K} \models f$.

**Non Entailment Ejection** For any KB $\mathcal{K}$, $R \in \mathcal{C}_{omax}(\mathcal{K}) \setminus \mathcal{O}(\mathcal{RIF}, \mathcal{K})$ and $f \in \mathcal{Cl}_{\mathcal{R}}(R)$, if there is no $R' \in \mathcal{C}_{omax}(\mathcal{K}) \setminus \{R\}$ with $f \in \mathcal{Cl}_{\mathcal{R}}(R')$ then $\mathcal{K} \not\models f$.

**Abstraction** For any KB $\mathcal{K}$, any RIF $\mathcal{RIF}$, any isomorphism $\gamma$ such that $\gamma(\mathcal{K}) = \mathcal{K}'$ and any query $q$, it holds that $\mathcal{K} \models^{\mathcal{RIF}} q$ if and only if $\gamma(\mathcal{K}) \models^{\mathcal{RIF}} \gamma(q)$.

**Proposition 4.3.** It holds that:

- Abstraction-I, Abstraction-L and Abstraction-Q imply Abstraction.

- Let $\mathcal{RIF} = (\mathtt{S}, \mathtt{L}^{\succeq}, \models_x^{\mathcal{RIF}})$ where $x \in \{AR, IAR, ICR\}$ then $\mathcal{RIF}$ satisfies Free and Non Entailment Ejection.

- Let $\mathcal{RIF} = (\mathtt{S}, \mathtt{L}_y^{\succeq}, \models_x^{\mathcal{RIF}})$ where $x \in \{AR, IAR, ICR\}$, $y \in \{leximax, max\}$ and $\mathtt{S}$ satisfies Bottom Facts then $\mathcal{RIF}$ satisfies Supremacy.

Although it is not always better to be more productive for all applications, the following result shows that the RIF is more productive than the usual IAR, AR and ICR semantics.

**Proposition 4.4.** Let $\mathcal{RIF}$ be a RIF and $q$ a query, then:

- If $\mathcal{K} \models_x q$ then $\mathcal{K} \models_x^{\mathcal{RIF}} q$ with $x \in \{AR, IAR, ICR\}$

- If $\mathcal{K} \models_{IAR}^{\mathcal{RIF}} q$ then $\mathcal{K} \models_{ICR}^{\mathcal{RIF}} q$

- If $\mathcal{K} \models_{ICR}^{\mathcal{RIF}} q$ then $\mathcal{K} \models_{AR}^{\mathcal{RIF}} q$

However, it is worth noting that our framework does not make every inconsistency tolerant inference relation more productive. For instance, that is not the case with the non objection semantics from [Benferhat *et al.*, 2016].

## 4.2 Algorithmic Results

In this section, we show a simple recursive algorithm that uses minimal inconsistent sets to compute the output of our framework and we study the behaviour of this algorithm thanks to an experiment. Since finding each minimal inconsistent set has been proven to be practically feasible as it is polynomial

for data complexity and exponential for combined complexity ([Lembo *et al.*, 2010; Hecham *et al.*, 2017a]), we make the assumption that this set is given. The output of our framework is obtained by calling Algorithm 1 with $X$ and $Z$ initialised to $\emptyset$ and $M$ to the set of minimal inconsistent sets of $\mathcal{K}$. The parameter $X$ is the set considered for building the result and $Z$ is the set used for avoiding redundancies by memorising encountered sets. In Algorithm 1, we begin by checking if the set considered was already encountered (see line $l_1$). Then, we proceed by finding facts with a minimal score that can be added to $X$ without triggering a minimal inconsistent set (see line $l_2$). If such facts cannot be found, it means that the considered set $X$ is indeed a repair and should be returned (see line $l_3$). Otherwise, the above mentioned process is repeated by augmenting the set of the considered set $X$ with the facts found in line $l_2$ (see line $l_4$). The set $E$ contains repairs with maximal elements with respect to $\mathtt{S}^{I_{MI}2}$. However, although these repairs contain maximal elements, they are not equivalent with respect to leximax and this is why we have to refine the set $E$ (see line $l_5$). Please note that an alternate definition of $\mathtt{S}^{I_{MI}}$ found in the work of [Hunter and Konieczny, 2010] can be used for a faster computation.

**Data**: Two sets of sets of facts $M, Z$ and a set of facts $X$
**Result**: A set of repairs $\mathcal{O}(\mathcal{RIF}, \mathcal{K})$ where
$$\mathcal{RIF} = (\mathtt{S}_{IMI}, \mathtt{L}_{leximax}^{\succeq}, \models^{\mathcal{RIF}})$$

**begin**
- $l_1$   **if** $X \notin Z$ **then**
  - $Z \longleftarrow Z \cup \{X\}$
  - $Y \longleftarrow \{f \in \mathcal{F} \setminus X \mid \text{for all}$ $m \subseteq X, m \cup \{f\} \notin M\}$
  - $l_2$   $Y' \longleftarrow \{y \in Y \mid \mathtt{S}_y^{I_{MI}} \text{ is minimal}\}$
  - **if** $Y' = \emptyset$ **then**
    - $l_3$   **return** $\{X\}$
  - **else**
    - $E \longleftarrow \emptyset$
    - **for** $y' \in Y'$ **do**
      - $l_4$   $E \longleftarrow E \cup OFRecc(M, X \cup \{y'\})$
    - **end**
    - $l_5$   $E' \longleftarrow$ top sets of $E$ with $\mathtt{L}_{leximax}^{\succeq}$ with respect to $\mathtt{S}^{I_{MI}}$
    - **return** $E'$
  - **end**
- **else**
  - **return** $\emptyset$
- **end**

**end**

**Algorithm 1:** The $OFRecc$ algorithm

We ran the algorithm on the KBs described by [Yun *et al.*, 2017] and compared its performance with a basic algorithm for computing the RIF, namely naively finding all repairs, computing the inconsistency values and refining them by using leximax. The KBs were split in two sets: A first set $A$ of 108 KBs with 2 to 7 facts, 0 to 6 rules and 1 to 4 binary

---

[2] The obtained set is similar to the notion of preferred subtheories of a stratification defined by [Brewka, 1989].

or ternary negative constraints and a second set $B$ of 26 KBs with 8 facts, 6 rules and between 1 and 2 binary or ternary negative constraints. For further details about the KBs, the reader is invited to consult the original paper of [Yun *et al.*, 2017]. The results were as follows:

- For the set $A$, the average number of repairs per KB was 2.89 and the average number of repairs in the output of the RIF was 2.11. It means that the average number of repair was reduced by 26.92%. Moreover, it takes an average of 543ms per KB to find the output of the RIF with the basic algorithm whereas it takes an average of 601ms per KB to find the same output with our algorithm.

- For the set $B$, the average number of repairs per KB was 4.5 and the average number of repairs in the output of the RIF was 3.62. It means that the average number of repair was reduced by of 19.66%. Moreover, it takes an average of 1.764s per KB to find the output of the RIF with the basic algorithm whereas it takes an average of 1.617s per KB to find the output with our algorithm.

Although finding the set $Y'$ and $E'$ (see $l_2$ and $l_5$) can be found in polynomial time, in the worst case scenario, we would still have to search throughout all the subsets of $\mathcal{F}$ which would be exponential with respect to $|\mathcal{F}|$. All experiments were performed on a Mac machine running on macOS High Sierra with an Intel core i5 2.8 GHz and 8GB of RAM and were reproduced multiple times.

### 4.3 Application Scenario

We now consider an application scenario constructed in the setting of the Pack4Fresh project ([Yun *et al.*, 2016]) which was aimed at choosing the best packaging for strawberries. In this project, an online poll consisting of 66 questions was submitted to an audience of 21 professionals from the food industry. We distinguished four kinds of professionals: the wholesalers, the floorwalkers, the quality managers and the warehouse managers. The questions were aimed at collecting the individual vision of each person about the characteristics of four packagings: the wooden packaging (WP), the plastic packaging with a plastic film (PPF), the plastic packaging with a rigid lid ($PRL$) and the opened plastic packaging without lid ($OPL$). The answers of this poll were formalised into a set of 50 facts and 160 rules. In our application scenario, the inconsistency of the KB comes from the fusion of the divergent visions of the several professionals about the four aforementioned packagings. These visions were explicitly expressed using the rules. For instance, the rule $\forall x(PPF(x, wholesaler0) \rightarrow cheapCost(x))$ conveys the idea that the entity $wholesaler0$ believes that the PPF is cheap. A group of packaging experts constructed another set of 18 rules constituting expert knowledge. For instance, the rule $\forall x(keepHumidity(x) \rightarrow badFridgeConservation(x))$ states that if $x$ is a packaging that keeps humidity then $x$ is a bad packaging for fridge conservation whereas the rule $\forall x(accelerateDecaying(x) \rightarrow badEffectOnFruits(X))$ states that if $x$ is a packaging that accelerates decay then it is considered as having a bad effect on fruits. Lastly, a set of 34 negative constraints representing conflicting atoms, such as $\forall x(notBadEffectOnFruits(x)$



| $WP_1$ | |
| --- | --- |
| $\vdots$ | |
| $WP_{16}$ | |
| $PPF_1, PRL_1$ | |
| $PPF_2, PRL_2$ | |
| $PPF_3$ | |
| $PPF_4$ | |
| $PPF_5, PRL_3, PRL_4$ | |
| $PPF_6$ | |
| $PRL_5$ | |
| $PRL_6$ | |
| $PPF_7$ | |
| $\vdots$ | |
| $PPF_9$ | |
| $OPL_1$ | |
| $OPL_2$ | |

From "worst" to "best" repairs

Table 1: Ranking on repairs. For simplicity, repairs are denoted by the packaging they are referencing

$\land badEffectOnFruits(x) \rightarrow \bot)$, and incompatibilities between packagings, such as $\forall x, y, z, t(OPL(x, y) \land PRL(z, t) \rightarrow \bot)$, was added.

The formalisation yielded a set of 33 repairs where each repair corresponds to the vision of a collection of individuals about a single packaging. For instance, the repair $OPL_2 = \{OPL(po, floorwalker0), OPL(po, wharehouse\_manager0)\}$ corresponds to the vision of $floorwalker0$ and $wharehouse\_manager0$ about the OPL. Amongst the 33 repairs, 16 concerned the WP, 6 concerned the PRL, 9 concerned the PPF and 2 concerned the OPL. The different number of repairs is explained by the diverse quantity of disagreements amongst individuals. For instance, only two wholesalers disagreed about the characteristics of the OPL whereas eight wholesalers disagreed about the characteristics of WP. In our model, the size of the repair corresponds to the number of individuals that agreed on all the characteristics of a specific packaging. Please note that the repairs are not ranked solely based on their cardinality.

Surprisingly, the ranking on repairs was extremely clear as it showed that $WP > PRL \sim PPF > OPL$ (see Table 1). Indeed, the repairs about WP were ranked above the other repairs. The repairs about the PRL were ranked roughly equally with the repairs about PPF and the repairs about OPL were last. The ranking was evaluated by a group of packaging experts which confirmed that the ranking on packagings was intuitive with respect to the data of the KB. Indeed, the experts acknowledged that WP was ranked first because its characteristics were slightly less contested by the experts.

The KB in DLGP format as well as a JAVA implementation of the tool for computing the output of our framework is accessible at: https://gite.lirmm.fr/yun/IJCAI2018.

## 5 Discussion

In this paper, we presented a framework that takes into consideration the inconsistency on the facts when using the re-

pairs for query answering and restricts the set of repairs to the best with respect to inconsistency values. Since we consider a subset of repairs, we obtain more answers than classical inconsistency tolerant query answering in most cases. We characterised the desirable properties of such framework with respect to the following properties: Abstraction, Free, Non Entailment Ejection and Supremacy. We also introduced an algorithm for computing the output of such a framework and showed its results on a real-life scenario about packagings.

Our framework is more abstract than the work of [Staworko *et al.*, 2012]: we can consider an inconsistency value returning the same preferences on facts, a lifting function that uses these preferences such that the completion optimal repairs are ranked first, Globally optimal repairs second, Pareto optimal repairs third and followed by the other repairs.

## Acknowledgments

## References

[Baget *et al.*, 2011] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.

[Baget *et al.*, 2016a] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. A General Modifier-Based Framework for Inconsistency-Tolerant Query Answering. In *KR 2016*, pages 513–516, 2016.

[Baget *et al.*, 2016b] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. Inconsistency-Tolerant Query Answering: Rationality Properties and Computational Complexity Analysis. In *JELIA 2016*, pages 64–80, 2016.

[Benferhat *et al.*, 1997] Salem Benferhat, Didier Dubois, and Henri Prade. Some Syntactic Approaches to the Handling of Inconsistent Knowledge Bases: A Comparative Study Part 1: The Flat Case. *Studia Logica*, 58(1):17–45, 1997.

[Benferhat *et al.*, 2016] Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Odile Papini, and Karim Tabia. Non-Objection Inference for Inconsistency-Tolerant Query Answering. In *IJCAI 2016*, pages 3684–3690, 2016.

[Brewka, 1989] Gerhard Brewka. Preferred Subtheories: An Extended Logical Framework for Default Reasoning. In *IJCAI 1989*, pages 1043–1048, 1989.

[Calì *et al.*, 2009] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *PODS 2009*, pages 77–86, 2009.

[Grant and Hunter, 2011] John Grant and Anthony Hunter. Measuring the Good and the Bad in Inconsistent Information. In *IJCAI 2011*, pages 2632–2637, 2011.

[Hecham *et al.*, 2017a] Abdelraouf Hecham, Pierre Bisquert, and Madalina Croitoru. On the Chase for All Provenance Paths with Existential Rules. In *RuleML+RR 2017*, pages 135–150, 2017.

[Hecham *et al.*, 2017b] Abdelraouf Hecham, Madalina Croitoru, and Pierre Bisquert. Argumentation-Based Defeasible Reasoning For Existential Rules. In *AAMAS 2017*, pages 1568–1569, 2017.

[Hunter and Konieczny, 2010] Anthony Hunter and Sébastien Konieczny. On the measure of conflicts: Shapley Inconsistency Values. *Artif. Intell.*, 174(14):1007–1026, 2010.

[Konieczny *et al.*, 2015] Sébastien Konieczny, Pierre Marquis, and Srdjan Vesic. On supported inference and extension selection in abstract argumentation frameworks. In *ECSQARU 2015*, pages 49–59, 2015.

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-Tolerant Semantics for Description Logics. In *RR 2010*, pages 103–117, 2010.

[Lembo *et al.*, 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.*, 33:3–29, 2015.

[Lukasiewicz *et al.*, 2015] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris, and Gerardo I. Simari. From Classical to Consistent Query Answering under Existential Rules. In *AAAI 2015*, pages 1546–1552, 2015.

[Marnette, 2009] Bruno Marnette. Generalized schema-mappings: from termination to tractability. In *PODS 2009*, pages 13–22, 2009.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking Data to Ontologies. *J. Data Semantics*, 10:133–173, 2008.

[Reiter, 1987] Raymond Reiter. A Theory of Diagnosis from First Principles. *Artif. Intell.*, 32(1):57–95, 1987.

[Staworko *et al.*, 2012] Slawek Staworko, Jan Chomicki, and Jerzy Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intell.*, 64(2-3):209–246, 2012.

[Yun *et al.*, 2016] Bruno Yun, Pierre Bisquert, Patrice Buche, and Madalina Croitoru. Arguing About End-of-Life of Packagings: Preferences to the Rescue. In *MTSR 2016*, pages 119–131, 2016.

[Yun *et al.*, 2017] Bruno Yun, Srdjan Vesic, Madalina Croitoru, Pierre Bisquert, and Rallou Thomopoulos. A Structural Benchmark for Logical Argumentation Frameworks. In *IDA 2017*, pages 334–346, 2017.