A Graded Approach to Database Repair by Context-Aware Distance Semantics

Ofer Arieli
School of Computer Science
The Academic College of Tel-Aviv, Israel
oarieli@mta.ac.il

Anna Zamansky
Department of Information Systems
University of Haifa, Israel
annazam@is.haifa.ac.il

May 22, 2015

Abstract

The problem of inconsistent information in databases often arises in the context of data integration and data exchange. In these areas the common assumption is that the real world is consistent, thus an inconsistent database does not correspond to any reliable state and it needs to be "repaired" according to a chosen policy. Many of these policies have to deal with the problem of an exponential blowup in the number of possible repairs. For this reason, recent approaches advocate more flexible and fine-grained policies based on the user's preference. In this paper we take a further step towards more personalized inconsistency management by incorporating ideas from context-aware systems. The idea is to employ grades of different repairs according to their relevance for a particular user. The outcome is a graded framework for inconsistency maintenance in database systems, controlled by context-aware and distance-based considerations.

1 Introduction

Inconsistency handling in constrained databases is a primary issue in the context of consistent query answering, data integration, and data exchange. The general view in such cases is that the inconsistent database does not provide a faithful description of its domain of discourse and therefore it should be "repaired" so that its consistency will be restored. The standard approaches to this issue are usually based on the principle of minimal change, aspiring to achieve consistency via a minimal amount of data modifications (see, e.g., [4, 12, 13, 22]). A key question in this respect is how to *choose* among the different possibilities of restoring the consistency of a database (i.e., 'repairing' it).

Earlier approaches to inconsistency management were based on the assumption that there should be some fixed, pre-determined way of repairing a database. Recently, there has been a paradigm shift towards user-controlled inconsistency management policies. Works taking this approach provide a possibility for the user to express some *preference* over all possible database repairs, preferring certain repairs to others (Some examples are [35, 41, 46, 60]; See [56] for a survey of other related works). While such approaches provide the user with flexibility and control over inconsistency management, in reality they entail a considerable technical burden on the user's shoulders of calibrating, updating and maintaining preferences or policies. Moreover, in many cases these preferences should be *dynamic*, changing quickly on the go (e.g., depending on the user's geographical location). In the era of ubiquitous computing, with database systems practically everywhere, database users have little technical background, and even less time to dwell on the technical details of inconsistency management. As a consequence, there is a frequent demand for *easy* – and sometimes even *fully automatic* – inconsistency management solutions with little cognitive load, which are still expected to be *personalized* and tuned for particular needs. This leads to the idea of introducing *context-awareness* into inconsistency management.

Context-awareness is defined as the use of contexts to provide task-relevant information and services to the user (see [1]). We believe that inconsistency management has natural relations to the concept of context. Accordingly, the goal of this paper is to incorporate notions and techniques that have been studied by the context-aware computing community (see, e.g., [20] and [54]) into consistency management in database systems. For this, we use a logical approach for preferring repairs, which combines the following two grading ingredients:

- *Distance-based semantics* for restoring the consistency of inconsistent databases according to the principle of minimal change, and
- *Context-awareness considerations*, based on graded ranking, for incorporating user preferences.

Example 1 Let us consider the following simple database instance:

empNum	name	address	salary
1	John	Tower Street 3, London, UK	70K$
1	John	Herminengasse 8, Wien, AT	80K$
2	Mary	42 Street 15, New York, US	90K$

A functional dependency that may be violated in this case is $\mathsf{empNum} \rightarrow \mathsf{salary}$, stating that the salary of an employee is uniquely determined by the employee's number. Thus, assuming that this database contains an information coming from several equally reliable sources, one has to resolve the inconsistency in it, although each source could have provided a completely consistent data. Minimal change considerations (which will be expressed in what follows by distance functions) imply in this case that it is enough to delete either the first or the second tuple for restoring consistency. Now, the decision which tuple to delete may be *context-dependent*. For instance, for tax assessments tuples with higher salaries may be preferred, while tuples with lower salaries may have higher priority when loans or grants are considered.

The choice between the first two tuples may also be determined by other, more dynamic considerations. For instance, it is quite possible that two different employees called John were assigned the same employee number by mistake. Alternatively, the same employee (John) may have two different addresses in two different countries, but the salary information associated with at least one of them is erroneous. In either cases, a user located in Austria is most probably interested in the Austrian address (or the Austrian employee), while a user located in the UK will make the most out of the other address (or employee).

The rest of this paper is organized as follows. In Section 2 we review some of the basic definitions of database concepts and distance-based semantics. In Section 3 we show how context-awareness can be modeled in our framework using a graded approach, and incorporate context-aware considerations into distance-based semantics. In Section 4 we consider some applications of our approach and in Section 5 we discuss some future work and conclude.[1]

[1]This paper is an extension of the work presented in the 35th Linz Seminar on Fuzzy Set Theory, dedicated to Graded Logical Approaches and their Applications (Linz, Austria, February 2014). A short version of this paper was also published in [64].

2 Inconsistent Databases and Distance Semantics

To simplify of the presentation, in this paper we remain on the propositional level and reduce first-order databases to our framework by grounding them. In the sequel, $\mathcal{L}$ denotes a propositional language with a *finite* set of atomic formulas $\mathsf{Atoms}(\mathcal{L})$. An $\mathcal{L}$-*interpretation* I is an assignment of a truth value in $\{T, F\}$ to every element in $\mathsf{Atoms}(\mathcal{L})$. Interpretations are extended to complex formulas in $\mathcal{L}$ in the usual way, using the truth tables of the connectives in $\mathcal{L}$. The set of two-valued interpretations for $\mathcal{L}$ is denoted by $\Lambda_{\mathcal{L}}$. An interpretation I is a *model* of an $\mathcal{L}$-formula ψ, denoted by $I \models \psi$, if $I(\psi) = T$, and it is a model of a set Γ of $\mathcal{L}$-formulas, denoted by $I \models \Gamma$, if it is a model of every $\mathcal{L}$-formula in Γ. The set of models of Γ is denoted by $mod(\Gamma)$. We say that Γ is *satisfiable* if $mod(\Gamma)$ is not empty.

Definition 2 A *database* $\mathcal{DB}$ in $\mathcal{L}$ is a pair $\langle \mathcal{D}, \mathcal{IC} \rangle$, where $\mathcal{D}$ (the *database instance*) is a finite subset of $\mathsf{Atoms}(\mathcal{L})$, and $\mathcal{IC}$ (the *integrity constraints*) is a finite and consistent set of $\mathcal{L}$-formulas.

The meaning of $\mathcal{D}$ is determined by the conjunction of its facts, augmented with Reiter's *closed world assumption*, stating that each atomic formula that does not appear in $\mathcal{D}$ is false. This may be expressed by the set $\mathsf{CWA}(\mathcal{D}) = \{\neg p \mid p \notin \mathcal{D}\}$. Henceforth, a database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ will be associated with the theory $\Gamma_{\mathcal{DB}} = \mathcal{IC} \cup \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})$.

In the sequel we shall sometimes identify a database with its associated theory. Thus, for instance, a model of $\mathcal{DB}$ is any interpretation satisfying $\Gamma_{\mathcal{DB}}$.

Definition 3 A database $\mathcal{DB}$ is *consistent* iff $\Gamma_{\mathcal{DB}}$ is satisfiable.

When a database is not consistent at least one integrity constraint is violated, and so it is usually required to look for "repairs" of the database, that is, changes of the database instance so that its consistency will be restored. There are numerous approaches for doing so (see, e.g., [4, 12, 22] for some surveys on this subject). Here we follow the distance-based approach described in [5, 7], which we find suitable for our purposes as it provides a modular and flexible framework for a variety of methods of (cardinality-based) database repair and consistent query answering (see also [12, 13] and the references therein).

Distance-based reasoning is extensively studied in the context of, e.g., paraconsistent reasoning, belief revision, knowledge integration, and consistent query answering in database systems. It is based on the notion of *preferential semantics* [47, 55], where preferences are expressed in terms of distance functions. In the context of database systems this approach aims at addressing the problem that when $\mathcal{DB}$ is inconsistent $mod(\Gamma_{\mathcal{DB}})$ is empty, so reasoning with $\mathcal{DB}$ is trivialized. This may be handled by replacing $mod(\Gamma_{\mathcal{DB}})$ with the set $\Delta(\mathcal{DB})$ of interpretations that, intuitively, are 'as close as possible' to (satisfying) $\mathcal{DB}$, while still satisfying the integrity constraints. When $\mathcal{DB}$ is consistent, $\Delta(\mathcal{DB})$ and $mod(\Gamma_{\mathcal{DB}})$ coincide (see Proposition 15 below), which assures that distance-based semantics is a conservative generalization of standard semantics for consistent databases.

In what follows, we recall the relevant definitions for formalizing the intuition above (see also [5, 7]).

Definition 4 A *pseudo-distance* on a set U is a total function $d : U \times U \to \mathbb{R}^+$, which is

- symmetric: for all $\nu, \mu \in U$, $d(\nu, \mu) = d(\mu, \nu)$, and
- preserves identity: for all $\nu, \mu \in U$, $d(\nu, \mu) = 0$ if and only if $\nu = \mu$.

A pseudo-distance d is called a *distance* (*metric*) on U, if it satisfies the triangular inequality:

- for all $\nu, \mu, \sigma \in U$, $d(\nu, \sigma) \leq d(\nu, \mu) + d(\mu, \sigma)$.

Example 5 One may define the following distances on $\Lambda_{\mathcal{L}}$:

$$d_U(I, I') = \begin{cases} 1 & \text{if } I \neq I', \\ 0 & \text{otherwise.} \end{cases}$$

$$d_H(I, I') = \mid \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I(p) \neq I'(p)\} \mid.$$

d_U is sometimes called the uniform distance and d_H is known as the Hamming distance. We note that the above distance functions are simple but not always sensitive enough.[2] More sophisticated distances based on aggregation of distances between (sets of) facts, are the Hausdorff distance [25], Eiter and Mannila's distance [28], and distances defined by matching functions [7]. We refer to [7] for demonstrating how the latter distances may be incorporated in a database repairing framework.

Definition 6 A *(numeric) aggregation function* is a function f, whose domain consists of multisets of real numbers and whose range is the real numbers, satisfying the following properties:

- f is non-decreasing when a multiset element is replaced by a larger element,
- $f(\{x_1, \ldots, x_n\}) = 0$ if and only if $x_1 = x_2 = \ldots x_n = 0$, and
- $f(\{x\}) = x$ for every $x \in \mathbb{R}$.

We say that an aggregation function f is *hereditary*, if $f(\{x_1, \ldots, x_n\}) < f(\{y_1, \ldots, y_n\})$ entails that $f(\{x_1, \ldots, x_n, z_1, \ldots, z_m\}) < f(\{y_1, \ldots, y_n, z_1, \ldots, z_m\})$.

In what follows we shall aggregate distance values. Since distances are non-negative numbers, aggregation functions in this case include, e.g., the summation and the maximum functions, the former is also hereditary.

Definition 7 A *distance setting* (for a language $\mathcal{L}$) is a pair $\mathsf{DS} = \langle d, f \rangle$, where d is a pseudo-distance on $\Lambda_{\mathcal{L}}$ and f is an aggregation function.

The next definition is a common way of using distance functions for maintaining inconsistent data (see, e.g, [5, 42, 43]).

Definition 8 For a finite set $\Gamma = \{\psi_1, \ldots, \psi_n\}$ of formulas in $\mathcal{L}$, an interpretation $I \in \Lambda_{\mathcal{L}}$, and a distance setting $\mathsf{DS} = \langle d, f \rangle$ for $\mathcal{L}$, we denote:

- $d_{\mathsf{DS}}(I, \psi_i) = \min\{d(I, I') \mid I' \models \psi_i\}$,
- $\delta_{\mathsf{DS}}(I, \Gamma) = f(\{d_{\mathsf{DS}}(I, \psi_1), \ldots, d_{\mathsf{DS}}(I, \psi_n)\})$.

Proposition 9 [5, 43] *Let Γ be a finite set of formulas in $\mathcal{L}$, ψ a formula in $\mathcal{L}$, I an interpretation in $\Lambda_{\mathcal{L}}$, and $\mathsf{DS} = \langle d, f \rangle$ a distance setting for $\mathcal{L}$. Then it holds that $I \models \psi$ iff $d_{\mathsf{DS}}(I, \psi) = 0$ and $I \models \Gamma$ iff $\delta_{\mathsf{DS}}(I, \Gamma) = 0$.*

The interpretations that are 'closest' to being models of $\Gamma_{\mathcal{DB}}$ are defined as follows:

Definition 10 Given a database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ in $\mathcal{L}$ and a distance setting $\mathsf{DS} = \langle d, f \rangle$ for $\mathcal{L}$, the set of *the most plausible interpretations of* $\mathcal{DB}$ (with respect to DS) is defined as follows:

$$\Delta_{\mathsf{DS}}(\mathcal{DB}) = \big\{I \in mod(\mathcal{IC}) \mid (\forall I')\; I' \in mod(\mathcal{IC}) \Longrightarrow \delta_{\mathsf{DS}}(I, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I', \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))\big\}.$$ [3]

[2]For instance, according to both of these functions, $\{p(a, b)\}$ and $\{p(c, d)\}$ are equally distant from $\{p(a, e)\}$, although $p(a, b)$ and $p(a, e)$ share the first argument.

[3]Closed word assumption is needed here to take into account atomic formulas that are not mentioned in the database instance but appear, e.g., in the integrity constraints or in the user's context environment (see Definition 18 below).

Note 11 Since $\mathcal{IC}$ is satisfiable and $\Lambda_{\mathcal{L}}$ is finite, for every database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ and a distance setting DS for its language, it holds that $\Delta_{\mathsf{DS}}(\mathcal{DB}) \neq \emptyset$.

Definition 12 Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and $\mathsf{DS} = \langle d, f \rangle$ a distance setting. We say that $\mathcal{R}$ is a DS-*repair* of $\mathcal{DB}$, if there is an $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ such that $\mathcal{R} = \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I(p) = T\}$. We shall sometimes denote this repair by $\mathcal{R}(I)$ and say that it is *induced by* I (or that I is the *characteristic model* of $\mathcal{R}$). The set of all the DS-repairs is denoted by $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$, that is, $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I) \mid I \in \Delta_{\mathsf{DS}}(\mathcal{DB})\}$.

An alternative characterization of the DS-repairs of $\mathcal{DB}$ is given next.

Proposition 13 *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and $\mathsf{DS} = \langle d, f \rangle$ a distance setting. Let I_S be the characteristic function of $S \subseteq \mathsf{Atoms}(\mathcal{L})$ (that is, $I_S(p) = T$ if $p \in S$ and $I_S(p) = F$ otherwise). The* DS-inconsistency value *of S (with respect to $\mathcal{DB}$) is defined by:*[4]

$$\mathsf{Inc}_{\mathsf{DS}}^{\mathcal{DB}}(S) = \begin{cases} \delta_{\mathsf{DS}}(I_S, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) & \textit{if } I_S \in \textit{mod}(\mathcal{IC}), \\ \infty & \textit{otherwise.} \end{cases}$$

Then $\mathcal{R} \subseteq \mathsf{Atoms}(\mathcal{L})$ is a DS-repair of $\mathcal{DB}$ iff its DS-inconsistency value is minimal among the DS-inconsistency values of the subsets of $\mathsf{Atoms}(\mathcal{L})$.

Proof. Let $\mathcal{R} \subseteq \mathsf{Atoms}(\mathcal{L})$ such that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$ for every $S \subseteq \mathsf{Atoms}(\mathcal{L})$. Since $\mathcal{IC}$ is satisfiable, $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) < \infty$, and so $I_{\mathcal{R}} \in \textit{mod}(\mathcal{IC})$. Let now $\mathcal{R}'$ be a DS-repair of $\mathcal{DB}$. Then there is an element $I' \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ such that $\mathcal{R}' = \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I'(p) = T\}$. But $\delta_{\mathsf{DS}}(I_{\mathcal{R}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I', \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))$, and so $I_{\mathcal{R}} \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ as well, which implies that $\mathcal{R}$ is a DS-repair of $\mathcal{DB}$.

For the converse, let $\mathcal{R}$ be a DS-repair of $\mathcal{DB}$ and let $S \subseteq \mathsf{Atoms}(\mathcal{L})$. We have to show that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$. Indeed, if $I_S \notin \textit{mod}(\mathcal{IC})$ then $\mathsf{Inc}_{\mathsf{DS}}(S) = \infty$ and so the claim is obtained. Otherwise, both $I_{\mathcal{R}}$ and I_S are models of $\mathcal{IC}$, and since $\mathcal{R}$ is a DS-repair of $\mathcal{DB}$, $I_{\mathcal{R}} \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. It follows that $\delta_{\mathsf{DS}}(I_{\mathcal{R}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I_S, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))$ and so $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$. □

Note 14 Interestingly, viewed as a preferred way to update an inconsistent database (and so to recover its inconsistency), the above construction of $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$ satisfies the five properties listed in [56].[5] This easily follows from Proposition 13 above. Indeed, by this proposition, for every database $\mathcal{DB}$ and a distance setting DS, $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \min_{<_{\mathsf{DS}}}\{S \mid S \subseteq \mathsf{Atoms}(\mathcal{L})\}$, where $S_1 <_{\mathsf{DS}} S_2$ iff $\mathsf{Inc}_{\mathsf{DS}}(S_1) < \mathsf{Inc}_{\mathsf{DS}}(S_2)$. This implies the satisfaction of the following postulates:

P1 *Non-Emptiness:* $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) \neq \emptyset$ because a $<_{\mathsf{DS}}$-minimum over the finite set $2^{\mathsf{Atoms}(\mathcal{L})}$ is always obtained. (Also since $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I) \mid I \in \Delta_{\mathsf{DS}}(\mathcal{DB})\}$ and $\Delta_{\mathsf{DS}}(\mathcal{DB}) \neq \emptyset$ by Note 11).

P2 *Monotonicity:* If $<_{\mathsf{DS}_1} \subseteq <_{\mathsf{DS}_2}$ then $\min_{<_{\mathsf{DS}_2}} 2^{\mathsf{Atoms}(\mathcal{L})} \subseteq \min_{<_{\mathsf{DS}_1}} 2^{\mathsf{Atoms}(\mathcal{L})}$, and so $\mathsf{Repairs}_{\mathsf{DS}_2}(\mathcal{DB}) \subseteq \mathsf{Repairs}_{\mathsf{DS}_1}(\mathcal{DB})$.

P3 *Non-Discrimination:* If $<_{\mathsf{DS}} = \emptyset$ then $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = 2^{\mathsf{Atoms}(\mathcal{L})}$.

P4 *Categoricity:* If $<_{\mathsf{DS}}$ is a total order then $|\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})| = 1$.

P5 *Conservativeness:* $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) \subseteq 2^{\mathsf{Atoms}(\mathcal{L})}$.

By Proposition 9 and Definition 12, we also have the following result:

[4]In what follows, when the underlying database is fixed or clear from the context, we shall omit the superscript $\mathcal{DB}$ from the notations of the inconsistency value.

[5]In [56] these properties are considered with respect to methods for making preferences among all the possible repairs.

Proposition 15 *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and* DS *a distance setting. The following conditions are equivalent:*

1. *$\mathcal{DB}$ is consistent,*
2. *$\Delta_{\mathsf{DS}}(\mathcal{DB}) = mod(\Gamma_{\mathcal{DB}})$,*
3. *$\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{D}\}$,*
4. *The DS-inconsistency value of every DS-repair of $\mathcal{DB}$ is zero.*

Proof. Suppose that $\Delta_{\mathsf{DS}}(\mathcal{DB}) = mod(\Gamma_{\mathcal{DB}})$. By Note 11 this means that $mod(\Gamma_{\mathcal{DB}}) \neq \emptyset$, and so $\mathcal{DB}$ is consistent. Conversely, if $\mathcal{DB}$ is consistent then $\Gamma_{\mathcal{DB}}$ is satisfiable. It's unique model $I_{\mathcal{D}}$ is the following:

$$\forall p \in \mathsf{Atoms}(\mathcal{L})\ I_{\mathcal{D}}(p) = \begin{cases} T & \text{if } p \in \mathcal{D}, \\ F & \text{if } p \notin \mathcal{D}. \end{cases}$$

It follows that for every data setting $\mathsf{DS} = \langle d, f \rangle$, $\delta_{\mathsf{DS}}(I_{\mathcal{D}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) = 0$ and so $\Delta_{\mathsf{DS}}(\mathcal{DB}) = mod(\Gamma_{\mathcal{DB}}) = \{I_{\mathcal{D}}\}$. This also implies that in this case $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I_{\mathcal{D}})\} = \{\mathcal{D}\}$ and so, by Proposition 9, $\mathsf{Inc}^{\mathcal{DB}}_{\mathsf{DS}}(\mathcal{D}) = 0$. Finally, if the DS-inconsistency value of every DS-repair of $\mathcal{DB}$ is zero, then every such repair is induced by a model of $\Gamma_{\mathcal{DB}}$. This means that $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I) \mid I \in \Delta_{\mathsf{DS}}(\mathcal{DB})\}$ may be represented by $\{\mathcal{R}(I) \mid I \in mod(\Gamma_{\mathcal{DB}})\}$, thus $\Delta_{\mathsf{DS}}(\mathcal{DB}) = mod(\Gamma_{\mathcal{DB}})$. □

Example 16 Let us return to the database in Example 1. The projection of the database table on the attributes id and salary is: $\Big\{ \langle 1, \mathsf{70K\$} \rangle , \langle 1, \mathsf{80K\$} \rangle\rangle , \langle 2, \mathsf{90K\$} \rangle \Big\}$. After grounding the database and representing the tuple $\langle \mathsf{empNum}, \mathsf{salary} \rangle$ by a propositional variable $\mathsf{T}^{\mathsf{empNum}}_{\mathsf{salary}}$, we have:

$$\mathcal{D} \cup \mathsf{CWA}(\mathcal{D}) = \Big\{ \mathsf{T}^1_{70K\$}, \mathsf{T}^1_{80K\$}, \neg\mathsf{T}^1_{90K\$}, \neg\mathsf{T}^2_{70K\$}, \neg\mathsf{T}^2_{80K\$}, \mathsf{T}^2_{90K\$} \Big\},$$

and the functional dependency $\mathsf{empNum} \to \mathsf{salary}$ is formulated as follows:

$$\mathcal{IC} = \Big\{ \mathsf{T}^x_y \to \neg\mathsf{T}^x_z \mid y \neq z,\ y, z \in \{70K\$, 80K\$, 90K\$\},\ x \in \{1, 2\} \Big\}.$$

Using the distance d_H from Example 5 and $f = \Sigma$, we compute:

$\mathcal{R}(I)$	$d_H(I, \mathsf{T}^1_{70})$	$d_H(I, \mathsf{T}^1_{80})$	$d_H(I, \neg\mathsf{T}^1_{90})$	$d_H(I, \neg\mathsf{T}^2_{70})$	$d_H(I, \neg\mathsf{T}^2_{80})$	$d_H(I, \mathsf{T}^2_{90})$	$\delta_{d_H\Sigma}(I, \Gamma_{\mathcal{DB}})$
$\emptyset$	1	1	0	0	0	1	3
$\{\mathsf{T}^1_{70}\}$	0	1	0	0	0	1	2
$\{\mathsf{T}^1_{80}\}$	1	0	0	0	0	1	2
...	...	...	...	...	...	...	...
$\{\mathsf{T}^1_{70}, \mathsf{T}^2_{90}\}$	0	1	0	0	0	0	**1**
$\{\mathsf{T}^1_{80}, \mathsf{T}^2_{90}\}$	1	0	0	0	0	0	**1**
...	...	...	...	...	...	...	...

It follows that $\Delta_{\langle d_H, \Sigma \rangle}(\mathcal{DB}) = \{I_1, I_2\}$ and $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I_1), \mathcal{R}(I_2)\}$, where $\mathcal{R}(I_1) = \{\mathsf{T}^1_{70}, \mathsf{T}^2_{90}\}$ and $\mathcal{R}(I_2) = \{\mathsf{T}^1_{80}, \mathsf{T}^2_{90}\}\}$. Note that $\mathcal{R}(I_1)$ means that in order to repair the database the tuple with $\mathsf{empNum} = 1$ and $\mathsf{salary} = 80K\$$ has to be removed, while $\mathcal{R}(I_2)$ indicates that the tuple with $\mathsf{empNum} = 1$ and $\mathsf{salary} = 70K\$$ should to be deleted. Thus, only T^2_{90} holds in all the repairs of $\mathcal{DB}$, that is, only the salary of Employee 2 is certain.

Example 17 Consider the database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ where $\mathcal{D} = \{\mathsf{rain},\ \mathsf{warm}\}$ and $\mathcal{IC} = \{\mathsf{rain} \to \mathsf{take\_umbrella}\}$. By the closed word assumption this database is not consistent, since $\neg\mathsf{take\_umbrella} \in \Gamma_{\mathcal{DB}}$. Using the same distance setting as in Example 16 we again have two ways of repairing the inconsistency in $\mathcal{DB}$, but this time one of them involves an insertion of a new fact to the database: one repair is by removing the assumption that it is rainy, and the other repair is by inserting an indication to take an umbrella.

3 Context-Aware Inconsistency Management

3.1 Context Modeling

As defined in [1],

> "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application."

This notion has been found useful in several domains, such as machine learning and knowledge acquisition (see, e.g., [15, 19]). We shall consider as a context any data that can be used to characterize database-related situations, involving database entities, user contexts and preferences, etcetera [11, 23]. This includes computational environments (e.g., network connectivity, nearby resources), user-related context (such as profile, location, mood, family situation), and measurable contexts (like noise levels, temperature, and time) [16].

There is a wide variety of methods for modeling contexts (see, e.g., [59]). Here we follow the data-centric approach introduced in [58], and refer to contexts using a finite set of special-purpose variables, which may not be part of the database.

Definition 18 A *context environment* (or just a *context*) C is a finite tuple of distinct variables $\langle c_1, \ldots, c_n \rangle$, where each variable c_i $(1 \leq i \leq n)$ has a corresponding range $\mathsf{Range}(c_i)$ of possible values. A *context state* for C (a C-state, for short) is an assignment S such that $S(c_i) \in \mathsf{Range}(c_i)$. The set of context states is denoted by $\mathsf{States}(C)$.

Intuitively, a context environment C represents the parameters that may be taken into consideration for the database inconsistency maintenance.

Example 19 Consider a process of repairing an obsolete database that contains information about the nationality of citizens of European Union countries. Such a database may contain data about people from countries that no longer exist, such as Czechoslovakia or Yugoslavia (violating a constraint listing the valid countries). It might also happen that, e.g., as a result of data integration, former Czechoslovakians are reported as being both Slovak and Czechs. In such a case a person's name could be used as a context to help tracing the correct nationality and resolve the contradiction, satisfying the integrity constraint of unique nationality of each person.

Example 20 A possible context for the database of Example 1 could be $C = \{\mathsf{location}, \mathsf{usermode}\}$, where $\mathsf{Range}(\mathsf{location})$ contains possible countries and $\mathsf{Range}(\mathsf{usermode})$ contains various modes of using the database. Accordingly, a state S in which $S(\mathsf{location}) = \mathsf{US}$ and $S(\mathsf{usermode}) = \mathsf{tax\_assessment}$ reflects an interest in tax-assessments of US companies.

Note 21 As follows from the last example, context variables need not necessarily refer to the content of the database instance. For another illustration of this, note that the decision how to repair the database of Example 17 may be affected by geographic locations or by the relevant season: in winter one may prefer to add the recommendation to take an umbrella, while in summer one may want to remove the assumption that it is rainy.

Note 22 It is possible to refer to a context state as a *function* rather than an *assignment*. According to this view, a context state of C is a tuple $\langle r_1(c_1), \ldots, r_n(c_n) \rangle$, where $r_i(c_i)$ $(1 \leq i \leq n)$ is an application of a ranking function $r(c_i) : \mathsf{Range}(c_i) \to \mathbb{N}$ on c_i. These functions may be useful, e.g., when a context involves fuzzy concepts or when more fine-grained discrimination is required among the possible values that c_i may have. This extended view of context states is beyond the scope of the present paper.

3.2 Context Settings and Context Sensitivity

We are now ready to incorporate context-awareness into distance considerations. We do so by making the 'most plausible' interpretations in $\mathcal{DB}$ (that is, the elements in $\Delta_{\mathsf{DS}}(\mathcal{DB})$) *sensitive to context*, in the sense that more 'relevant' formulas have higher impact on the distance computations than less 'relevant' formulas. Thus, while we still strive to minimize change, the latter will be measured in a more subtle, context-aware way. For this purpose we introduce graded functions which measure the relevance of information depending on context.

Definition 23 A *relevance ranking* for a set Γ of formulas and a context environment C, is a total function $R : \Gamma \times \mathsf{States}(C) \to (0, 1]$.

Given a set Γ and a context environment C, a relevance ranking function for Γ and C assigns to every formula $\psi \in \Gamma$ and every state S of C a (positive) *relevance factor* $R(\psi, S)$ indicating the relevance of ψ according to S. Intuitively, higher values of these factors correspond to higher relevance of their formulas, which makes changes to these formulas in computing database repairs less desirable.

Note 24 Grading information, either by numerical values or by preference orders, is not unusual in AI systems, as it often helps to maintain consistency and cope with other anomalies in the data. In that respect, we may recall the CP-orders on the rules of multi-context systems, which can be used to explain inconsistency in those systems [27], probabilistic qualification of attack relations that help to provide a subtle conflict-free understanding of argumentation systems [39], and the methods for prioritized reasoning in logic programming with preference relations among the rules for providing coherent semantics to such programs (see [36] and the references therein). For some other monographs on preference modeling and further references see e.g. [18] and [48]. Here, relevance factors may be thought of as a context-dependent interpretation of preference/scoring functions [2] or of weights in prioritized theories [6]. However, unlike the grading approaches mentioned above, the domain of the ranking functions is not restricted to the available data, and so it can involve *any* kind of information which may be used to determine the most plausible repairs. In particular, this makes the concepts of contexts and personalization somewhat more abstractive and dynamic.

Definition 25 A *context setting* for a set of formulas Γ is a triple $\mathsf{CS}(\Gamma) = \langle C, S, R \rangle$, where C is a context environment, $S \in \mathsf{States}(C)$ is a C-state, and R is a relevance ranking function for Γ and C. In what follows we shall sometimes denote by $\mathsf{CS}(\mathcal{L})$ a context setting $\mathsf{CS}(\Gamma)$ in which Γ is the set of all the well-formed formulas of $\mathcal{L}$.

Consistency restoration for databases can now be defined as before (see Definitions 8 and 10). The outcome of this is demonstrated by the following example.

Example 26 Let us reconsider the database of Example 17. Let $\mathsf{CS} = \langle C, S, R \rangle$ be a context setting where $C = \{\mathsf{season}\}$ with $\mathsf{Range}(\mathsf{season}) = \{\mathsf{summer}, \mathsf{autumn}, \mathsf{winter}, \mathsf{spring}\}$, and

$$R(\mathsf{rain}, S) = R(\mathsf{take\_umbrella}, S) = \begin{cases} 1, & \text{if } S(\mathsf{season}) = \mathsf{winter} \text{ or } S(\mathsf{season}) = \mathsf{autumn}, \\ 0.5, & \text{otherwise.} \end{cases}$$

$$R(\mathsf{warm}, S) = \begin{cases} 1, & \text{if } S(\mathsf{season}) = \mathsf{summer} \text{ or } S(\mathsf{season}) = \mathsf{spring}, \\ 0.5, & \text{otherwise.} \end{cases}$$

Suppose also that for every $p \in \{\mathsf{rain}, \mathsf{warm}, \mathsf{take\_umbrella}\}$ we have $R(\neg p, S) = \max(1 - R(p, S), \epsilon)$ for some small $\epsilon > 0$.[6] Now, let $\mathsf{DS} = \langle d^{\mathsf{CS}}_{\Sigma}, \Sigma \rangle$ be a corresponding distance setting, where

$$d^{\mathsf{CS}}_{\Sigma}(I, I') = \Sigma(\{R(p, S) \cdot |I(p) - I'(p)| \mid p \in \{\mathsf{rain}, \mathsf{warm}, \mathsf{take\_umbrella}\}\}).$$

[6]We need the ϵ to avoid zeroed R-values. Clearly, its value should be less than 0.5; The exact value of ϵ may depend on various considerations, such as the size of the database instance or the ratio between the 'cost' of the insertion and the deletion operations.

It is not difficult to verify that d^{CS}_{Σ} is indeed a pseudo-distance. Calculations that are similar to those of Example 16 (where f is still Σ but now the distance is d^{CS}_{Σ}) show that in a state S where $S(\mathsf{season}) \in \{\mathsf{winter}, \mathsf{autumn}\}$ the repair in which $\mathsf{take\_umbrella}$ is added to the database will be preferred, and in a state S where $S(\mathsf{season}) \in \{\mathsf{spring}, \mathsf{summer}\}$ the repair in which rainy is removed from the database will be preferred.

As we show below (and as indicated at the beginning of this subsection), to properly reflect the user preference in the database repairs, the distance setting should be tightly linked to the underlying preference setting. More precisely, the underlying distance setting $\mathsf{DS} = \langle d, f\rangle$ should be context-sensitive in the sense that d_{DS} should preserve the order induced by ranking function, as defined next.

Definition 27 Let $\mathsf{CS}(\mathcal{L}) = \langle C, S, R\rangle$ be a context setting for a language $\mathcal{L}$. A distance setting $\mathsf{DS} = \langle d, f\rangle$ is called CS-*sensitive*, if for every two atomic formulas p_1 and p_2 such that $R(p_1, S) > R(p_2, S)$, it holds that $d_{\mathsf{DS}}(I_2, p_1) > d_{\mathsf{DS}}(I_1, p_2)$ for every $I_1 \in mod(p_1) \setminus mod(p_2)$ and $I_2 \in mod(p_2) \setminus mod(p_1)$.

Clearly, the results of Section 2 (e.g., Proposition 15) still hold for context-sensitive distance settings. Next, we demonstrate the effect of incorporating context sensitive distance settings on inconsistency management.

Proposition 28 *Let $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1, p_2\}, \mathcal{IC}\rangle$ be a database*[7], *$\mathsf{CS} = \langle C, S, R\rangle$ a context setting and $\mathsf{DS} = \langle d, f\rangle$ a CS-sensitive distance setting in which f is hereditary. If $R(p_1, S) > R(p_2, S)$, then for every $\mathcal{D}' \subseteq \mathsf{Atoms}(\mathcal{L}) \setminus \{p_1, p_2\}$ such that $\mathcal{D}' \sqcup \{p_1\} \models \mathcal{IC}$, the DS-inconsistency value of $\mathcal{D}_1 = \mathcal{D}' \sqcup \{p_1\}$ is smaller than the DS-inconsistency value of $\mathcal{D}_2 = \mathcal{D}' \sqcup \{p_2\}$.*

Proof. Let $\mathcal{D}' \subseteq \mathsf{Atoms}(\mathcal{L}) \setminus \{p_1, p_2\}$ and $\mathcal{D}_1 = \mathcal{D}' \cup \{p_1\}$. Since $\mathcal{D}_1 \models \mathcal{IC}$, we have that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) < \infty$. Thus, $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) < \mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_2)$ whenever $\mathcal{D}_2 \not\models \mathcal{IC}$. Suppose then that $\mathcal{D}_2 \models \mathcal{IC}$ as well. In this case, in the notations of Proposition 13, we have that $I_{\mathcal{D}_1}$ and $I_{\mathcal{D}_2}$ differ only in the assignments for p_1 and p_2 (I.e., $I_{\mathcal{D}_1}$ satisfies p_1 and falsifies p_2 while $I_{\mathcal{D}_2}$ satisfies p_2 and falsifies p_1. Elsewhere, both interpretations are equal to $I_{\mathcal{D}'}$). Now, since DS is CS-sensitive, by the facts that (i) $R(p_1, S) > R(p_2, S)$, (ii) $I_{\mathcal{D}_1} \in mod(p_1) \setminus mod(p_2)$ and (iii) $I_{\mathcal{D}_2} \in mod(p_2) \setminus mod(p_1)$, we have that $d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2) < d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)$. Let $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D} \sqcup \{p_1, p_2\}) = \{\psi_1, \ldots, \psi_n\}$. By the assumption that f is hereditary,

$$\begin{aligned}
&\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) = \delta_{\mathsf{DS}}(I_{\mathcal{D}_1}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \\
&\quad = f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_n), d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_1), d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) && \text{(by the definition of } \delta_{\mathsf{DS}}) \\
&\quad = f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) && \text{(since } I_{\mathcal{D}_1} \text{ satisfies } p_1) \\
&\quad = f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) && (I_{\mathcal{D}_1}, I_{\mathcal{D}_2} \text{ differ only in } p_1, p_2) \\
&\quad < f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)\}) && (\mathsf{CS}\text{-sensitivity; } f \text{ is hereditary}) \\
&\quad = f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_2), d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)\}) && \text{(since } I_{\mathcal{D}_2} \text{ satisfies } p_2) \\
&\quad = \delta_{\mathsf{DS}}(I_{\mathcal{D}_2}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) = \mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_2) && \text{(by the definition of } \delta_{\mathsf{DS}}) \quad \square
\end{aligned}$$

It follows that when context-sensitive distances are incorporated, "more relevant" formulas are preferred in the repairs. This is shown next.

Proposition 29 *Let $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1, p_2\}, \mathcal{IC}\rangle$ be a database, $\mathsf{CS} = \langle C, S, R\rangle$ a context setting and $\mathsf{DS} = \langle d, f\rangle$ a CS-sensitive distance setting in which f is hereditary. If*

1. *$\mathcal{DB}_1 = \langle \mathcal{D} \sqcup \{p_1\}, \mathcal{IC}\rangle$ is a consistent database,*

2. *$R(p_1, S) > R(p_2, S)$, and*

[7]We denote by $\mathcal{D} \sqcup \{p_1, p_2\}$ the disjoint union of $\mathcal{D}$ and $\{p_1, p_2\}$.

3. $\mathcal{IC} \cup \{p_1, p_2\}$ is (classically) inconsistent.

Then no DS*-repair of* $\mathcal{DB}$ *contains* p_2.[8]

Proof. Suppose otherwise, and let $\mathcal{R}_2 \in \mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$ be a DS-repair of $\mathcal{DB}$ such that $p_2 \in \mathcal{R}_2$. In particular, $\mathcal{R}_2$ is induced by some $I_2 \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. Note that since $p_2 \in \mathcal{R}_2$, necessarily $I_2(p_2) = T$, and since $I_2 \models \mathcal{IC}$ (because $I_2 \in \Delta_{\mathsf{DS}}(\mathcal{DB})$), necessarily $I_2(p_1) = F$ (otherwise I_2 is a model of $\mathcal{IC} \cup \{p_1, p_2\}$ which contradicts the assumption that the latter is inconsistent). It follows that $\mathcal{R}_2 = \mathcal{D}' \sqcup \{p_2\}$ for some $\mathcal{D}' \subseteq \mathcal{D}$. Consider now the set $\mathcal{R}_1 = \mathcal{D}' \sqcup \{p_1\}$. Since $\mathcal{DB}_1$ is a consistent database, $\mathcal{R}_1 \models \mathcal{IC}$, and so by Proposition 28, $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}_1) < \mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}_2)$. Thus, $\delta_{\mathsf{DS}}(I_1, \mathcal{DB}) < \delta_{\mathsf{DS}}(I_2, \mathcal{DB})$, where I_1 is the characteristic model of $\mathcal{R}_1$ (see Definition 12). This contradicts the assumption that $I_2 \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. □

Under a further condition (Definition 30) Proposition 29 can be strengthened (Proposition 31).

Definition 30 A distance setting $\mathsf{DS} = \langle d, f \rangle$ for a language $\mathcal{L}$ is called *uniform*, if for every two interpretations $I_1, I_2 \in \Lambda_{\mathcal{L}}$ and for every atom $p \in \mathsf{Atoms}(\mathcal{L})$, $I_1(p) = I_2(p)$ implies that $d_{\mathsf{DS}}(I_1, p) = d_{\mathsf{DS}}(I_2, p)$.

It is easy to verify that every distance setting in which the distance is one of those mentioned in Example 5, is uniform.

Proposition 31 *Let* DS *be a uniform distance setting. Then in the notations of Proposition 29 and under its assumptions,* $\mathcal{D}_1 = \mathcal{D} \sqcup \{p_1\}$ *is the unique* DS*-repair of* $\mathcal{DB}$.

Proof. We show that $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I_1\}$, where I_1 ia the (unique) model of $\Gamma_{\mathcal{DB}_1}$, defined by $I_1(p) = T$ if $p \in \mathcal{D}_1$ and $I_1(p) = F$ otherwise (such a model exists since $\mathcal{DB}_1$ is consistent). The claim then follows from the fact that $\mathcal{D}_1 = \mathcal{R}(I_1)$, i.e., $\mathcal{D}_1$ is the DS-repair of $\mathcal{DB}$, which is induced by I_1.

Suppose that $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D} \sqcup \{p_1, p_2\}) = \{\psi_1, \ldots, \psi_n\}$, and let $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. Then

$$\delta_{\mathsf{DS}}(I, \mathcal{DB}) = \quad f(\{d_{\mathsf{DS}}(I, \psi_1), \ldots, d_{\mathsf{DS}}(I, \psi_n), d_{\mathsf{DS}}(I, p_1), d_{\mathsf{DS}}(I, p_2)\}).$$

Now, by Proposition 29, since DS is CS-sensitive, $I(p_2) = F$. Also, by the definition of I_1 we have that $I_1(p_2) = F$, and so, since DS is uniform, $d_{\mathsf{DS}}(I, p_2) = d_{\mathsf{DS}}(I_1, p_2)$. It follows that

$$\begin{aligned} \delta_{\mathsf{DS}}(I, \mathcal{DB}) \geq \quad & f(\{0, \ldots, 0, 0, d_{\mathsf{DS}}(I, p_1), d_{\mathsf{DS}}(I, p_2)\}) \geq \\ & f(\{0, \ldots, 0, 0, d_{\mathsf{DS}}(I, p_2)\}) = \\ & f(\{0, \ldots, 0, 0, d_{\mathsf{DS}}(I_1, p_2)\}) = \\ & \delta_{\mathsf{DS}}(I_1, \mathcal{DB}). \end{aligned}$$

Thus, $I_1 \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. On the other hand, if there is some $q \in \{\psi_1, \ldots, \psi_n, p_1\}$ for which $d_{\mathsf{DS}}(I, q) \neq 0$, then since f is hereditary the above inequality becomes strict, which contradicts the assumption that $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. It follows that for every $q \in \{\psi_1, \ldots, \psi_n, p_1\}$ $d_{\mathsf{DS}}(I, q) = d_{\mathsf{DS}}(I_1, q) = 0$, i.e., $I \models q$. One concludes, then, that I is a model of $\mathcal{DB}_1$, that is, $I = I_1$. □

Propositions 29 and 31 can be generalized as follows:

Corollary 32 *Let* $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ *be a database,* $\mathsf{CS} = \langle C, S, R \rangle$ *a context setting and* $\mathsf{DS} = \langle d, f \rangle$ *a* CS*-sensitive distance setting in which* f *is hereditary. Suppose that* $\mathcal{D} = \mathcal{D}' \sqcup \mathcal{D}''$ *can be partitioned to two disjoint nonempty subsets* $\mathcal{D}'$ *and* $\mathcal{D}''$ *such that*

- $\mathcal{DB}' = \langle \mathcal{D}', \mathcal{IC} \rangle$ *is a consistent database,*

[8]Note that this is true even in case that $\mathcal{DB}_2 = \langle \mathcal{D} \sqcup \{p_2\}, \mathcal{IC} \rangle$ is a consistent database.

- *$\forall p'' \in \mathcal{D}''$ $\exists p' \in \mathcal{D}'$ such that $\mathcal{IC} \cup \{p', p''\}$ is not consistent, and*
- *$\forall p' \in \mathcal{D}'$ and $\forall p'' \in \mathcal{D}''$ it holds that $R(p', S) > R(p'', S)$.*

Then for every DS-repair $\mathcal{R}$ of $\mathcal{DB}$, $\mathcal{R} \cap \mathcal{D}'' = \emptyset$.

Proof. Let $p_2 \in \mathcal{D}''$. By Condition (2), there is $p_1 \in \mathcal{D}'$ such that $\mathcal{IC} \cup \{p_1, p_2\}$ is not consistent. Thus, by similar considerations as those in the proof of Proposition 29, no DS-repair of $\mathcal{DB}$ contains the fact p_2. □

Corollary 33 *In case that DS is a uniform distance setting, then in the notations of Proposition 32 and under its assumptions, $\mathcal{D}'$ is the unique DS-repair of $\mathcal{DB}$.*

Proof. Similar to that of Proposition 31, using Corollary 32. □

3.3 A Simple Construction of Context-Sensitive Distance Settings

In what follows we provide a concrete method for defining context-sensitive distance settings and exemplify some of the properties of the settings that are obtained.

Definition 34 Let $\mathsf{CS}(\mathcal{L}) = \langle C, S, R\rangle$ be a context setting for $\mathcal{L}$ and let g be an aggregation function. The (pseudo) distance d_g^{CS} on $\Lambda_{\mathcal{L}}$ is defined as follows:

$$d_g^{\mathsf{CS}}(I, I') = g(\{R(p, S) \cdot |I(p) - I'(p)| \ \mid \ p \in \mathsf{Atoms}(\mathcal{L})\}).$$

It is easy to verify that for any CS and g, the function d_g^{CS} is indeed a pseudo-distance on $\Lambda_{\mathcal{L}}$. In particular, for any context setting $\mathsf{CS}(\mathcal{L}) = \langle C, S, R\rangle$ where R is uniformly 1, d_Σ^{CS} coincides with the Hamming distance d_H in Example 5. Note also that the pseudo distance used in Example 26 is a particular case of the definition above.

Note 35 Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC}\rangle$, $\mathsf{CS} = \langle C, S, R\rangle$, and $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$. For every set $\mathcal{S} \subseteq \mathsf{Atoms}(\mathcal{L})$ whose characteristic function $I_{\mathcal{S}}$ (recall Proposition 13) is a model of $\mathcal{IC}$, we have that:

$$\begin{aligned}&\mathsf{Inc}_{\mathsf{DS}}(\mathcal{S}) = \delta_{\mathsf{DS}}(I_{\mathcal{S}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) = \\ &\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{S}}, p) \mid p \in \mathcal{D}\} \cup \{d_{\mathsf{DS}}(I_{\mathcal{S}}, \neg p) \mid \neg p \in \mathsf{CWA}(\mathcal{D})\}) = \\ &\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{S}}, p) \mid p \in \mathcal{D}\} \cup \{d_{\mathsf{DS}}(I_{\mathcal{S}}, \neg p) \mid p \in \mathsf{Atoms}(\mathcal{L}) \setminus \mathcal{D}\}).\end{aligned}$$

Since $d_{\mathsf{DS}}(I_{\mathcal{S}}, p) = 0$ for every $p \in \mathcal{S}$ and $d_{\mathsf{DS}}(I_{\mathcal{S}}, \neg p) = 0$ for every $p \notin \mathcal{S}$, we have that

$$\begin{aligned}&\mathsf{Inc}_{\mathsf{DS}}(\mathcal{S}) = f(\overline{0} \cup \{d_{\mathsf{DS}}(I_{\mathcal{S}}, p) \mid p \in \mathcal{D} \setminus \mathcal{S}\} \cup \{d_{\mathsf{DS}}(I_{\mathcal{S}}, \neg p) \mid p \in \mathcal{S} \setminus \mathcal{D}\}) = \\ &\quad f(\overline{0} \cup \{g(\overline{0} \cup \{R(p, S)\}) \mid p \in \mathcal{D} \setminus \mathcal{S}\} \cup \{g(\overline{0} \cup \{R(\neg p, S)\}) \mid p \in \mathcal{S} \setminus \mathcal{D}\}).\end{aligned}$$

Denote $f(\overline{0}, x) = f(\{0, \ldots, 0, x, 0, \ldots, 0\})$. Whenever $f(\overline{0}, x) = f(x)$ and $g(\overline{0}, x) = g(x)$ (e.g., when f, g are the summation or the maximum function over non-negative values) we have that

$$\mathsf{Inc}_{\mathsf{DS}}(\mathcal{S}) = f(\{g(R(p, S)) \mid p \in \mathcal{D} \setminus \mathcal{S}\} \cup \{g(R(\neg p, S)) \mid p \in \mathcal{S} \setminus \mathcal{D}\}).$$

For monotonic f and g, then, what matters are the R-values of the atoms in the symmetric difference of D and the set of atoms (i.e., $\mathcal{S}$) under consideration. The latter is a repair of $\mathcal{D}$ (with respect to $\mathcal{IC}$) when these R-values are as minimal as possible.

The next proposition provides a general way of constructing context-sensitive distance settings, based on the functions in Definition 34.

Proposition 36 *Let $\mathsf{CS} = \langle C, S, R\rangle$ be a context setting and let $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$ be a distance setting, where g is a hereditary. Then DS is CS-sensitive.*

Proof. Let p_1 and p_2 be atomic formulas such that $R(p_1,S) > R(p_2,S)$, and let $I_1 \in mod(p_1) \setminus mod(p_2)$ and $I_2 \in mod(p_2) \setminus mod(p_1)$. Again, we denote $g(\overline{0},x) = g(\{0,\ldots,0,x,0,\ldots,0\})$. By Definition 8,

$$\begin{aligned} d_{\mathsf{DS}}(I_1,p_2) &= \min\{d_g^{\mathsf{CS}}(I_1,J) \mid J \models p_2\} \\ &= \min\{g(\{R(p,S) \cdot |I_1(p) - J(p)| \mid p \in \mathsf{Atoms}(\mathcal{L})\}) \mid J \models p_2\}. \end{aligned}$$

Since g is hereditary, the minimum above must be obtained for a model J of p_2 that coincides with I_1 on every atom $p \neq p_2$. It follows, then, that $d_{\mathsf{DS}}(I_1,p_2) = g(\overline{0},R(p_2,S))$. By similar considerations, $d_{\mathsf{DS}}(I_2,p_1) = g(\overline{0},R(p_1,S))$. Now, since $R(p_1,S) > R(p_2,S)$ and since g is hereditary, $d_{\mathsf{DS}}(I_2,p_1) > d_{\mathsf{DS}}(I_1,p_2)$. □

The next proposition demonstrates how CS-sensitive distance settings of the form defined above give precedence to "more relevant" facts: if two facts are the cause for the database inconsistency, the one with higher relevance ranking will be accepted while the other one will be rejected.

Proposition 37 *Let $\mathsf{CS} = \langle C,S,R\rangle$ be a context setting and let $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$ be a distance setting, where g and f are hereditary aggregation functions. Let $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1,p_2\}, \mathcal{IC}\rangle$ be a database such that the following conditions are satisfied:*

1. *$R(p_1,S) > R(p_2,S)$ (i.e., p_1 is more relevant than p_2), and*
2. *$\mathcal{IC} \cup \{p_1,p_2\}$ is not consistent (thus $\mathcal{DB}$ is not a consistent database), but the database $\mathcal{DB}_1 = \langle \mathcal{D} \sqcup \{p_1\}, \mathcal{IC}\rangle$ is consistent.*[9]

Then $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I_1\}$, where I_1 is the (unique) model of $\mathcal{DB}_1$.

Proof. By Proposition 36 DS is CS-sensitive. It is easy to see that DS is also uniform, and so by Proposition 31 the proposition is obtained. □

Example 38 Consider again Example 16, but this time let $\mathcal{D} = \{\mathsf{T}^2_{90K\$}\}$. In terms of that example, the insertion of $\mathsf{T}^1_{70K\$}$ and $\mathsf{T}^1_{80K\$}$ violates $\mathcal{IC}$, thus only the fact with the higher rank will be accepted.

Proposition 37 may be extended in various ways. The next proposition gives one such extension.

Proposition 39 *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC}\rangle$ be a database, $\mathsf{CS} = \langle C,S,R\rangle$ a context setting and $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$ a distance setting where g and f are hereditary aggregation functions. Suppose that $\mathcal{D}$ can be partitioned to two nonempty subsets $\mathcal{D}'$ and $\mathcal{D}''$, such that*

1. *$\mathcal{DB}' = \langle \mathcal{D}', \mathcal{IC}\rangle$ is a consistent database,*
2. *$\forall p'' \in \mathcal{D}''\ \exists p' \in \mathcal{D}'$ s.t. $\mathcal{IC} \cup \{p',p''\}$ is not consistent, and*
3. *$\forall p' \in D'$ and $\forall p'' \in \mathcal{D}''$, $R(p',S) > R(p'',S)$.*

Then $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I'\}$, where I' is the (unique) model of $\mathcal{DB}'$.

Proof. Similar to the proof of Proposition 33, using the fact that DS is uniform and CS-sensitive. Below, we repeat the main arguments, adjusted to the specific construction in Definition 34.

Again, we denote: $g(\overline{0},x) = g(\{0,\ldots,0,x,0,\ldots,0\})$. Then, for every atom p and interpretation I, we have that:

$$d_{\mathsf{DS}}(I,p) = \begin{cases} 0 & \text{if } I \models p, \\ g(\overline{0},R(p,S)) & \text{otherwise.} \end{cases}$$

[9]Again, $\mathcal{DB}_2\langle \mathcal{D} \sqcup \{p_2\}, \mathcal{IC}\rangle$ may be consistent as well, but this is not a prerequisite.

Let $\mathcal{D}' = \{p_1, \ldots, p_n\}$, $\mathcal{D}'' = \{q_1, \ldots, q_m\}$, and $\mathsf{CWA}(\mathcal{D}) = \{\psi_1, \ldots, \psi_k\}$. By similar consideration as in the proof of Proposition 37, we have that:

$$\begin{aligned}
(*) \quad & \delta_{\mathsf{DS}}(I', \mathcal{DB}) = \\
& f(\{d_{\mathsf{DS}}(I', \psi_1), \ldots, d_{\mathsf{DS}}(I', \psi_k), d_{\mathsf{DS}}(I', p_1), \ldots, d_{\mathsf{DS}}(I', p_n), d_{\mathsf{DS}}(I', q_1), \ldots, d_{\mathsf{DS}}(I', q_m)\}) = \\
& f(\{0, \ldots 0, d_{\mathsf{DS}}(I', q_1), \ldots, d_{\mathsf{DS}}(I, q_m)\}) = \\
& f(\{0, \ldots, 0, g(\overline{0}, R(q_1, S)), \ldots, , g(\overline{0}, R(q_m, S))\}).
\end{aligned}$$

Let now $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. By Corollary 32, $I \not\models q_i$ for every $1 \leq i \leq m$, thus $d_{\mathsf{DS}}(I, q_i) = g(\overline{0}, R(q_i, S))$ for every $1 \leq i \leq m$. It follows that:

$$\begin{aligned}
(**) \quad & \delta_{\mathsf{DS}}(I, \mathcal{DB}) = \\
& f(\{d_{\mathsf{DS}}(I,\psi_1), \ldots, d_{\mathsf{DS}}(I,\psi_k), d_{\mathsf{DS}}(I,p_1), \ldots, d_{\mathsf{DS}}(I,p_n), d_{\mathsf{DS}}(I,q_1), \ldots, d_{\mathsf{DS}}(I,q_m)\}) = \\
& f(\{d_{\mathsf{DS}}(I,\psi_1), \ldots, d_{\mathsf{DS}}(I,\psi_k), d_{\mathsf{DS}}(I,p_1), \ldots, d_{\mathsf{DS}}(I,p_n), g(\overline{0}, R(q_1, S)), \ldots, g(\overline{0}, R(q_n, S))\}).
\end{aligned}$$

A pointwise comparison of the arguments of f in the last lines of (*) and (**) above indicates that the i-th argument of f in (*) is less than or equal to the i-th argument of f in (**). Thus $\delta_{\mathsf{DS}}(I', \mathcal{DB}) \leq \delta_{\mathsf{DS}}(I, \mathcal{DB})$, and so $I' \in \Delta_{\mathsf{DS}}(\mathcal{DB})$, i.e., $\mathcal{DB}'$ is a repair of $\mathcal{DB}$. Moreover, if I characterizes a repair $\mathcal{R}$ other than $\mathcal{DB}'$ then the inequality above becomes strict (because f is hereditary and at least one argument of f in (*) is strictly smaller than the corresponding argument of f in (**)), but this is a contradiction to the assumption that $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. This implies that I must coincide with I', and so $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I'\}$. □

Example 40 Consider again the database in Example 1. By Example 16, the distance setting $\mathsf{DS} = \langle d_H, \Sigma \rangle$ leads to the following two equally good repairs:

Repair 1 :

eNum	name	address	salary
1	John	..., UK	70K$
2	Mary	..., US	90K$

Repair 2 :

eNum	name	address	salary
1	John	..., AT	80K$
2	Mary	..., US	90K$

Sensitivity to context may differentiate between these repairs, preferring one to another. Let us again denote by $\mathsf{T}^1_{\mathsf{UK}}$, $\mathsf{T}^1_{\mathsf{AT}}$ and $\mathsf{T}^2_{\mathsf{US}}$ the tuple according to which John lives in the UK and is payed 70K$, John lives in Austria and is payed 80K$, and the tuple with the information about Mary.

Now, consider the context setting $\mathsf{CS}(\mathcal{L}) = \langle C, S, R \rangle$ and the distance setting $\mathsf{DS} = \langle d^{\mathsf{CS}}_{\Sigma}, \Sigma \rangle$, where the context environment is $C = \{\mathsf{country}\}$, its range is $\mathsf{Range}(\mathsf{country}) = \{\mathsf{US}, \mathsf{UK}, \mathsf{AT}\}$, the state is $S(\mathsf{country}) = \mathsf{UK}$, and the relevance ranking is given by the following functions:

$$R(\mathsf{T}^{\mathsf{i}}_{\mathsf{c}}, S) = \begin{cases} 1, & \text{if } c = S(\mathsf{country}), \\ 0.5, & \text{otherwise.} \end{cases} \qquad R(\neg\mathsf{T}^{\mathsf{i}}_{\mathsf{c}}, S) = \begin{cases} 0.5, & \text{if } c = S(\mathsf{country}), \\ 1, & \text{otherwise.} \end{cases}$$

Computation of Δ_{DS} is given in the table below (where we abbreviate $d(\psi, S)$ for $d^{\mathsf{CS}}_{\Sigma}(\psi, S)$).

$\mathcal{R}(I)$	$d(I, \mathsf{T}^1_{\mathsf{UK}}, S)$	$d(I, \mathsf{T}^1_{\mathsf{AT}}, S)$	$d(I, \neg\mathsf{T}^1_{\mathsf{US}}, S)$	$d(I, \neg\mathsf{T}^2_{\mathsf{UK}}, S)$	$d(I, \neg\mathsf{T}^2_{\mathsf{AT}}, S)$	$d(I, \mathsf{T}^2_{\mathsf{US}}, S)$	$\delta_{\mathsf{DS}}(I, \Gamma, S)$
$\emptyset$	1	0.5	0	0	0	0.5	2
$\{\mathsf{T}^1_{\mathsf{UK}}\}$	0	0.5	0	0	0	0.5	1
$\{\mathsf{T}^1_{\mathsf{AT}}\}$	1	0	0	0	0	0.5	1.5
$\{\mathsf{T}^1_{\mathsf{US}}\}$	1	0.5	1	0	0	0.5	3
...	...	...	...	...	...	...	...
$\{\mathsf{T}^1_{\mathsf{UK}}, \mathsf{T}^2_{\mathsf{US}}\}$	0	0.5	0	0	0	0	**0.5**
$\{\mathsf{T}^1_{\mathsf{AT}}, \mathsf{T}^2_{\mathsf{US}}\}$	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...

According to CS, the single element in $\Delta_{\mathsf{DS}}(\mathcal{DB})$ satisfies $\{\mathsf{T}^1_{\mathsf{UK}}, \mathsf{T}^2_{\mathsf{US}}\}$, and so Repair 1 is preferred. Dually, in a state S' where $S'(\mathsf{country}) = \mathsf{AT}$, Repair 2 is preferred. Thus, context-aware considerations lead us to choose different repairs according to the relevance ranking, as indeed guaranteed by Propositions 37 and 39 (see also Example 38).

Note 41 A more sophisticated relevance ranking, which makes preferences among locations according to their distances from the state location, could be

$$R(\mathsf{T}^x_y, s) = 1 - \frac{\mathsf{Dist}(y, S(\mathsf{location}))}{N+1},$$

where N denotes the maximal distance from $S(\mathsf{location})$. Similarly, if one prefers higher salary values, the relevance ranking for the same database could assign to each salary its value normalized to the $(0,1]$-interval. If lower salary values are preferred, one may consider instead a ranking assigning $\frac{1}{x}$ to a salary of x, and so forth. A proper choice of the ranking functions is of-course a crucial issue here, but this is beyond the scope of the present paper.

4 Some Notes on Applications

In this section we comment on some concrete applications of our framework. Specifically, we demonstrate how two different approaches to database repair may be interpreted as contexts in our framework and as a consequence may be applied using our setting. These approaches involve two types of considerations: repair operations and user preferences specified as partial orders.

4.1 Repair Operations

A common way to repair an inconsistent database is by minimizing the number of changes in the database instance (see, e.g., [4, 7, 12, 13, 45]). This 'cardinality-based' approach is reproduced in the next definition.

Definition 42 Given a (possibly inconsistent) database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ for a language $\mathcal{L}$, a *pairwise repair* of $\mathcal{DB}$ is a pair $\mathcal{R} = (R^+, R^-)$, where $R^+, R^- \subseteq \mathsf{Atoms}(\mathcal{L})$, such that:

(a) $R^+ \cap \mathcal{D} = \emptyset$ and $R^- \subseteq \mathcal{D}$,[10]

(b) The database $\langle \mathcal{D} \cup R^+ - R^-, \mathcal{IC} \rangle$ is consistent, and

(c) (R^+, R^-) is minimal in its cardinality: there is no pair $\langle S^+, S^- \rangle$ that satisfies Conditions (a) and (b), and for which $|S^+ \cup S^-| < |R^+ \cup R^-|$.

Intuitively, R^+ is the set of atoms that should be inserted to $\mathcal{D}$ and R^- is the set of atoms that should be deleted from $\mathcal{D}$ for restoring the consistency of $\mathcal{DB}$. Repaired databases are then consistent databases which are derived from a given database by means of a minimal number of insertions and deletions. The correspondence between the repairs in Definition 42 and in Definition 12 is realized in [7], where (a variation of) the next result is shown:

Proposition 43 *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and denote by $\mathsf{Repairs}_{\mathsf{card}}(\mathcal{DB})$ the set of (cardinality-based) pairwise repairs of $\mathcal{DB}$, as defined in Definition 42. Then there is a one-to-one correspondence between the elements in $\mathsf{Repairs}_{\mathsf{card}}(\mathcal{DB})$ and the elements in $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$ for $\mathsf{DS} = \langle d_U, \Sigma \rangle$. Moreover, it holds that:*

[10]In particular, $R^+ \cap R^- = \emptyset$.

1. If $\mathcal{R} \in \mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$ *then* $\langle \mathcal{R} - \mathcal{D}, \mathcal{D} - \mathcal{R} \rangle \in \mathsf{Repairs}_{\mathsf{card}}(\mathcal{DB})$*,*

2. If $(R^+, R^-) \in \mathsf{Repairs}_{\mathsf{card}}(\mathcal{DB})$ *then* $\mathcal{D} \cup R^+ - R^- \in \mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB})$*.*

We note that another common way of repairing databases is obtained by exchanging the cardinality-based requirement in Condition (c) of Definition 42 by a set-inclusion criterion (stating that there is no $\langle S^+, S^- \rangle$ satisfying Conditions (a) and (b), for which $S^+ \cup S^- \subsetneq R^+ \cup R^-$). This is the basic idea behind the repairing method introduced in [3], followed by the works in, e.g., [4, 8, 34, 37, 44]. Clearly, every pairwise repair that is obtained by the cardinality-based Definition 42 is also a repair according to the set-inclusion approach, but the converse is not necessarily true (see also [12, 13, 45]).

In [34], Greco et al. introduced a qualitative approach to database repair, using polynomial functions that assign a numeric value to each repair that reflects its quality. A certain repair of $\mathcal{DB}$ is considered preferred with respect to a given function f, if its f-value is minimal among the f-values of the repairs of $\mathcal{DB}$. Among the functions used in [34] for repair evaluations are those that count the number of inserted atoms (thus, repairs with a minimal number of insertions are preferred over other repairs), the number deleted atoms (so the amount of retractions is minimized), and the number of modified atoms. It is easy to see that these preference criteria may be simulated by corresponding contexts in our framework. Indeed, given a database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$, let $\langle C, S, R \rangle$ be the context setting where $C = \{\mathsf{minimized\ action}\}$, $\mathsf{Range}(\mathsf{minimized\ action}) = \{\mathsf{insertion}, \mathsf{deletion}\}$, and, for some small[11] $\epsilon > 0$,

$$R(p,S) = \begin{cases} 1-\epsilon, & \text{if } S(\mathsf{minimized\ action}) = \mathsf{deletion} \text{ and } p \in \mathcal{D} \\ & \text{or } S(\mathsf{minimized\ action}) = \mathsf{insertion} \text{ and } p \notin \mathcal{D}, \\ \epsilon, & \text{if } S(\mathsf{minimized\ action}) = \mathsf{deletion} \text{ and } p \notin \mathcal{D} \\ & \text{or } S(\mathsf{minimized\ action}) = \mathsf{insertion} \text{ and } p \in \mathcal{D}. \end{cases}$$

For every p and S we let $R(\neg p, S) = 1 - R(p,S)$. Using the distance setting $\mathsf{DS} = \langle d_{\Sigma}^{\mathsf{CS}}, \Sigma \rangle$, we have that in a state S where $S(\mathsf{minimized\ action}) = \mathsf{deletion}$ repairs with a minimal number of deletions are preferred, and when $S(\mathsf{minimized\ action}) = \mathsf{insertion}$ repairs with a minimal number of insertions are preferred.

4.2 User Preference

User preference has been widely explored in the database community, in particular in the context of query personalization. It reflects the subjective *value of information*, a notion that is well-studied in information systems and related communities (see, e.g., [53]). While the information preferred by the user is highly subjective and hard to measure, measuring the value of information in the presence of inconsistency is even more challenging. Yet, one could say that the relation between inconsistency and information value is a kind of an inverse dependency: the information becomes less valuable as the inconsistency in it increases. Moreover, inconsistency related to more valuable information is often more significant. Thus, given the user preference, we can express it in terms of information value, and then strive to minimize the significance of inconsistency.

A common way of expressing user preferences is by partial orders [21, 56]. Below, we assume that preferred data is assigned higher values.

Definition 44 Let $\mathbf{L}$ be a finite set of literals. A *preference* $\mathbf{P}$ for $\mathbf{L}$ is an irreflexive and transitive partial order on $\mathbf{L}$. A *path* in $\mathbf{P}$ is a sequence $p = \langle l_1, \ldots, l_n \rangle \in \mathbf{L}$, such that $(l_1, l_2), \ldots, (l_{n-1}, l_n) \in \mathbf{P}$. In this case, for every $1 \le i \le n$ we denote $weight_p(l_i) = \frac{i}{n}$. Finally, for every literal $l \in \mathbf{L}$, we define: $val_{\mathbf{P}}(l) = min\{weight_p(l) \mid p \text{ is a path in } \mathbf{P} \text{ and } l \in p\}$.

[11]Here, again, the exact value of ϵ may depend on various considerations, like the relative cost of insertions versus deletions, to what extent the content of the database is reliable (and so whether deletions are allowed), etc.

Given an inconsistent database $\mathcal{DB}$ and a preference $\mathbf{P}$ for $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D})$, we can use $\mathbf{P}$ for defining a context setting for the computation of the most plausible repairs.

Definition 45 Let $\mathcal{DB}$ be a database, and let $\mathbf{PR}$ be the set of all possible preferences on $\Gamma = \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})$. This induces a variety of context settings $\mathsf{CS}(\Gamma) = \langle C, S, R \rangle$ for Γ, in which $C = \langle \mathsf{pref} \rangle$ with $Range(\mathsf{pref}) = \mathbf{PR}$, $S \in \mathbf{PR}$, and the relevance ranking $R : \Gamma \times \mathsf{States}(C) \to (0, 1]$ is defined for every $l \in \Gamma$ and $\mathbf{P} \in \mathsf{PR}$ by $R(l, \mathbf{P}) = val_{\mathbf{P}}(l)$.[12]

Example 46 In Example 16, let $\mathbf{P} = \{(\mathsf{T}^1_{70K}, \mathsf{T}^1_{80K}), (\mathsf{T}^2_{70K}, \mathsf{T}^2_{80K})\}$ be a preference on $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D})$. Then, e.g., $R(\mathsf{T}^1_{70K}, \mathbf{P}) = \frac{1}{2}$ and $R(\mathsf{T}^1_{80K}, \mathbf{P}) = 1$. Since none of the negative literals occurs in $\mathbf{P}$, all of them are evaluated by 1.

We note, finally, that our approach may be useful also for providing distance-based indications about the inconsistency of $\mathcal{DB}$, like those considered, e.g., in [33]. Such a measurement could be, for instance, the numerical value $\delta_{\mathsf{DS}}(I, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))$ where $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. Since this value is the same for *every* $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$, it reflects a property of the database itself.[13] Note that this measurement is zeroed only if $\mathcal{DB}$ is consistent and is strictly positive otherwise.

5 Concluding Remarks and Further Research

This paper focuses on personalizing the process of management of inconsistent information in database systems by means of context-aware considerations. As observed in [29], contexts are still understudied in the AI community. In the scope of database systems, context awareness has only recently been addressed in relation to user preference in querying (consistent) databases [52, 57]. To the best of our knowledge, the approach presented here is the first one to directly apply context-aware considerations for an automated inconsistency management. Combined with the extensive work available on personalization and automatically determining user's context and preferences (see, e.g., [10, 24, 38]), it may open the door to new inconsistency management solutions and novel database technologies. A by-product of this work is therefore a step towards linking works on consistent-query answering in database systems (like [3, 4, 12, 13, 22, 37, 63]) and disciplines that are originated from information science perspective (e.g., [1, 10, 11, 15, 19, 20, 23, 29, 58, 59]). Implementation and evaluation of the methods in this paper are currently a work in progress.[14]

There are a number of directions for further research. First, we mainly focus here on propositional databases. As hinted, e.g., in Example 5, more sophisticated definitions of context and distance settings are available for first-order languages (see also [49] and [63]), which are yet to be incorporated in our framework.

Another issue for further exploration is considering knowledge bases which may contain also complex formulas. In this case one may take further advantage of the basic ideas of mathematical fuzzy logic, namely that grades are combined using some logical operators and define, e.g., the following principles for ranking complex formulas:

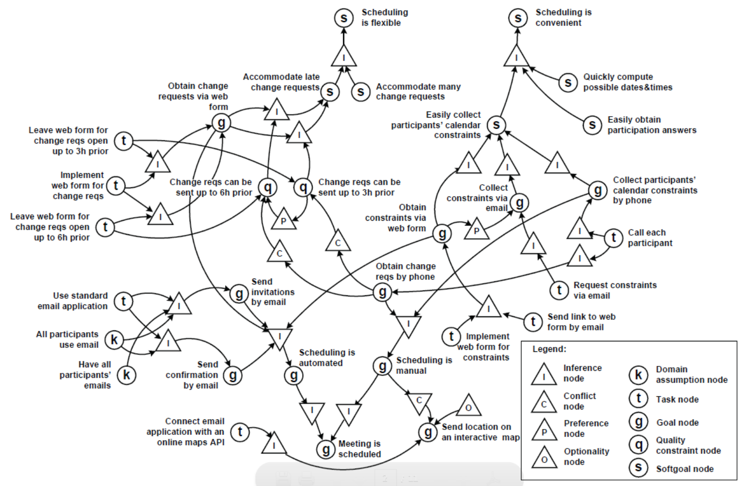
$$\begin{array}{lcl} R(\neg\psi, S) & = & 1 - R(\psi, S),^{15} \\ R(\psi_1 \wedge \psi_2, S) & = & \min(R(\psi_1, S), R(\psi_2, S)), \\ R(\psi_1 \vee \psi_2, S) & = & \max(R(\psi_1, S), R(\psi_2, S)), \\ R(\psi_1 \supset \psi_2, S) & = & \max(R(\neg\psi_1, S), R(\psi_2, S)). \end{array}$$

[12]Note that each literal l has at least one path to which it belongs: the path $\langle l \rangle$ of size 1. Thus, in the absence of longer paths, the relevance ranking of l is 1.

[13]In fact, this value may be expressed by: $\min\{\mathsf{Inc}^{\mathcal{DB}}_{\mathsf{DS}}(S) \mid S \subseteq \mathsf{Atoms}(\mathcal{L})\}$; See Proposition 13.

[14]See `http://mailng.hevra.haifa.ac.il/∼annazam/publications/publications.html` for a (Java-based) demonstration of computing context-aware repairs by d^{CS}_{Σ} and by d^{CS}_{max}. Extending this tool to a more general setting could also allow for an integration with the Tweety project libraries [62].

One interesting domain in which the ideas described in this paper can be potentially useful is that of requirement engineering (RE). The requirements problem was originally formulated by Zave and Jackson in [65] as an abductive problem: given requirements R and domain assumptions D, find specification S, satisfying $D \cup S \vdash R$ under the condition that $D \cup S$ is consistent. The last condition reflects the standard approach in RE that all contradictions must be eliminated *before* the solutions are identified. It has been acknowledged that this is rarely possible to achieve (see, e.g., [50, 51]) and so several inconsistency-tolerant substitute approaches were proposed. One such approach is a paraconsistent language for modeling requirements, called Techne [30, 40]. This language allows to relax the consistency assumption and is based on the idea of looking for the most suitable maximally consistent subsets of the theory. This is done by means of *r-nets*, which are directed labeled graphs representing goals, tasks, and relationships between them, such as conflict (that is understood as logical inconsistency), optionality, and preference. An example of an r-net (capturing requirements of a scheduler) is shown in Figure 1. The relationships of conflicts are represented by arcs labeled by C-marked triangles.[16]


Scheduling is flexible
Scheduling is convenient
Accommodate late change requests
Accommodate many change requests
Obtain change requests via web form
Quickly compute possible dates&times
Easily collect participants' calendar constraints
Easily obtain participation answers
Leave web form for change reqs open up to 3h prior
Implement web form for change reqs
Leave web form for change reqs open up to 6h prior
Change reqs can be sent up to 6h prior
Change reqs can be sent up to 3h prior
Collect constraints via email
Collect participants' calendar constraints by phone
Obtain constraints via web form
Call each participant
Obtain change reqs by phone
Request constraints via email
Use standard email application
Send invitations by email
Send link to web form by email
All participants use email
Implement web form for constraints
Have all participants' emails
Send confirmation by email
Scheduling is automated
Scheduling is manual
Connect email application with an online maps API
Meeting is scheduled
Send location on an interactive map
Legend:
Inference node
Conflict node
Preference node
Optionality node
Domain assumption node
Task node
Goal node
Quality constraint node
Softgoal node

An r-net capturing requirements about the meeting scheduler problem.

Figure 1: An r-net (taken from [17])

Finding maximally consistent subsets of requirements in r-nets can be simulated in our framework by translating an r-net into a logical theory, taking the nodes representing requirements as its literals, and formulas encoding the relationships of conflicts and non-optional nodes as integrity constraints. That is, a conflict between $r_1, \ldots, r_n$ can be represented via the formula $r_1 \wedge \ldots \wedge r_n \to \bot$ (where $\bot$ is a propositional constant representing falsity). Preference can then be transformed into a relevance ranking as described above. An actual implementation of this idea is left for future work.

Future work also involves computational considerations regarding our framework. Results concerning the computational complexity of decision problems in related frameworks show that (as expected) these problems are not tractable. For instance, Theorem 1 in [14] shows that even for simple integrity constraints of the form of functional dependencies or inclusion dependencies, determining whether there is a repair whose weighted distance[17] from the database theory is not bigger than a certain

[15] In this case, one may restrict the range of R to the *open* unit interval $(0, 1)$ to avoid zeroed values of R.

[16] For instance, the requirements that the location should be sent on an interactive map is in conflict with the requirement that scheduling is manual.

[17] That is, the DS-inconsistency value of the repair, where distances are factored by numeric data (as in Definition 34).

threshold, is NP-complete. Yet, our conjecture is that the incorporation of context-awareness ingredients in the distance computations does not increase the complexity of the related problems, at least as far as the context sensitive distances in Section 3.3 are concerned. Known techniques for query answering that are based on (answer set) logic programming [4, 26, 31], as well as studies on the computational complexity of related problems [56, 61] may be helpful to verify these issues. Likewise, experience gained in similar implementations [44] and experiments with algorithms for reasoning with distance semantics [9, 32] and for producing similar repairs [14] can help in checking the suitability of our framework for handling practical cases.

References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[2] R. Agrawal and E. L. Wimmers. A framework for expressing and combining preferences. In *Proc. SIGMOD Conference on Management of Data*, pages 297–306. ACM, 2000.

[3] M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. PODS'99*, pages 68–79. ACM, 1999.

[4] M. Arenas, L. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming*, 3(4–5):393–424, 2003.

[5] O. Arieli. Distance-based paraconsistent logics. *International Journal of Approximate Reasoning*, 48(3):766–783, 2008.

[6] O. Arieli. Reasoning with prioritized information by iterative aggregation of distance functions. *Journal of Applied Logic*, 6(4):589–605, 2008.

[7] O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Annals of Mathematics and Artificial Intelligence*, 50(3–4):389–415, 2007.

[8] O. Arieli, M. Denecker, B. Van Nuffelen, and M. Bruynooghe. Coherent integration of databases by abductive logic programming. *Artificial Intelligence Research*, 21:245–286, 2004.

[9] O. Arieli and A. Zamansky. Simplified forms of computerized reasoning with distance semantics. *Journal of Applied Logic*, 9(1):1–22, 2011.

[10] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[11] M. Bazire and P. Brézillon. Understanding context before using it. In *Proc. CONTEXT'05*, LNCS 3554, pages 29–40. Springer, 2005.

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

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

[14] P. Bohannon, M. Flaster, W. Fan, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proc. SIGMOD Conference on Management of Data*, pages 143–154. ACM, 2005.

[15] C. Bolchini, C. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A. Schreiber, and L. Tanca. And what can context do for data? *Commumicationms of the ACM*, 52(11):136–140, 2009.

[16] C. Bolchini, C. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A data-oriented survey of context models. *ACM Sigmod Record*, 36(4):19–26, 2007.

[17] A. Borgida, N. Ernst, I. Jureta, A. Lapouchnian, S. Liaskos, and J. Mylopoulos. Techne: A(nother) requirements modeling language. Technical Report No. CSRG-593, Computer Systems Research Institute, University of Toronto, 2009.

[18] R. Brafman and C. Domshlak. Preference handling – An introductory tutorial. *AI Magazine*, 30(1):58–86, 2009.

[19] P. Brézillon. Context in artificial intelligence: I. A survey of the literature. *Computers and Artificial Intelligence*, 18(4), 1999.

[20] P. J. Brown, J. D. Bovey, and X. Chen. Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communication*, 4(5):58–64, 1997.

[21] J. Chomicki. Querying with intrinsic preferences. In *Proc. EDBT'02*, LNCS 2287, pages 34–51. Springer, 2002.

[22] J. Chomicki. Consistent query answering: Five easy pieces. In *Proc. ICDT'07*, LNCS 4353, pages 1–17. Springer, 2007.

[23] A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[24] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2):97–166, 2001.

[25] J. Dieudonné. *Foundations of Modern Analysis.* Academic Press, 1969.

[26] T. Eiter. Data integration and answer set programming. In *Proc. LPNMR'05*, LNCS 3662, pages 13–25. Springer, 2005.

[27] T. Eiter, M. Fink, P. Schüller, and A. Weinzierl. Finding explanations of inconsistency in multi-context systems. *Journal of Artificial Intelligence*, 216:233–274, 2014.

[28] T. Eiter and H. Mannila. Distance measure for point sets and their computation. *Acta Informatica*, 34:109–133, 1997.

[29] H. R. Ekbia and A. G. Maguitman. Context and relevance: A pragmatic approach. In *Modeling and Using Context*, pages 156–169. Springer, 2001.

[30] N. A. Ernst, A. Borgida, J. Mylopoulos, and I. J. Jureta. Agile requirements evolution via paraconsistent reasoning. In *Advanced Information Systems Engineering*, pages 382–397. Springer, 2012.

[31] E. Franconi, A. L. Palma, N. Leone, S. Perri, and F. Scarcello. Census data repair: A challenging application of disjunctive logic lrogramming. In *Proc. LPAR'01*, LNCS 2250, pages 561–578. Springer, 2001.

[32] N. Gorogiannis and A. Hunter. Implementing semantic merging operators using binary decision diagrams. *International Journal of Approximate Reasoning*, 49(1):234–251, 2008.

[33] J. Grant and A. Hunter. Distance-based measures of inconsistency. In *Proc. ECSQARU'13*, LNCS 7958, pages 230–241. Springer, 2013.

[34] S. Greco, C. Sirangelo, I. Trubitsyna, and E. Zumpano. Preferred repairs for inconsistent databases. In *Proc. IDEAS'03*, pages 202–211. IEEE, 2003.

[35] S. Greco, C. Sirangelo, I. Trubitsyna, and E. Zumpano. Feasibility conditions and preference criteria in querying and repairing inconsistent databases. In *Proc. DEXA'04*, LNCS 3180, pages 44–55. Springer, 2004.

[36] S. Greco, I. Trubitsyna, and E. Zumpano. On the semantics of logic programs with preferences. *Journal of Artificial Intelligent Research*, 30:501–523, 2007.

[37] S. Greco and E. Zumpano. Querying inconsistent databases. In *Proc. LPAR'2000*, LNAI 1955, pages 308–325. Springer, 2000.

[38] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, 2006.

[39] Anthony Hunter. Probabilistic qualification of attack in abstract argumentation. *Journal of Approximate Reasoning*, 55(2):607–638, 2014.

[40] I. Jureta, A. Borgida, N. A. Ernst, and J. M. Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling. In *Proc. RE'10*, pages 115–124, 2010.

[41] Y. Katsis, A. Deutsch, Y. Papakonstantinou, and V. Vassalos. Inconsistency resolution in online databases. In *Proc. ICDE'10*, pages 1205–1208. IEEE, 2010.

[42] S. Konieczny, J. Lang, and P. Marquis. DA2 merging operators. *Artificial Intelligence*, 157(1–2):49–79, 2004.

[43] S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *Logic and Computation*, 12(5):773–808, 2002.

[44] N. Leone, T. Eiter, W. Faber, M. Fink, G. Gottlob, and G. Greco. Boosting information integration: The INFOMIX system. In *Proc. SEBD'05*, pages 55–66, 2005.

[45] A. Lopatenko and L. Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In *Proc. ICDT'07*, LNCS 4353, pages 179–193. Springer, 2007.

[46] M. V. Martinez, F. Parisi, A. Pugliese, G. I. Simari, and V. S. Subrahmanian. Inconsistency management policies. In *Proc. KR'08*, pages 367–377. AAAI Press, 2008.

[47] J. McCarthy. Circumscription – A form of non monotonic reasoning. *Artificial Intelligence*, 13(1–2):27–39, 1980.

[48] S. Moretti, M. Öztürk, and A. Tsoukiàs. Preference modelling. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research and Management Science*, pages 27–59. Springer, 2012.

[49] S. H. Nienhuys-Cheng. Distance between Herbrand interpretations: A measure for approximations to a target concept. In *Proc. ILP'97*, LNCS 1297, pages 213–226. Springer, 1997.

[50] B. Nuseibeh, S. Easterbrook, and A. Russo. Leveraging inconsistency in software development. *IEEE Computer*, 33(4):24–29, 2000.

[51] B. Nuseibeh, S. Easterbrook, and A. Russo. Making inconsistency respectable in software development. *Journal of Systems and Software*, 58(2):171–180, 2001.

[52] E. Pitoura, K. Stefanidis, and P. Vassiliadis. Contextual database preferences. *IEEE Data Engeneering Bulletin*, 34(2):19–26, 2011.

[53] S. Rafaeli and D. Raban. Experimental investigation of the subjective value of information in trading. *Journal of the Association for Information Systems*, 4(5):119–139, 2003.

[54] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced interaction in context. In *Proc. HUC'99*, LNCS 1707, pages 89–101. Springer, 1999.

[55] Y. Shoham. *Reasoning about Change.* MIT Press, 1988.

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

[57] K. Stefanidis and E. Pitoura. Fast contextual preference scoring of database tuples. In *Proc. EDBT'08*, volume 261 of *ACM International Conference Proceeding Series*, pages 344–355. ACM, 2008.

[58] K. Stefanidis, E. Pitoura, and P. Vassiliadis. Managing contextual preferences. *Information Systems*, 36(8):1158–1180, 2011.

[59] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[60] V. S. Subrahmanian and L. Amgoud. A general framework for reasoning about inconsistency. In *Proc. IJCAI'07*, pages 599–504, 2007.

[61] B. ten Cate, G. Fontaine, and P. Kolaitis. On the data complexity of consistent query answering. In *Proc. ICDT'12*, pages 22–33. ACM, 2012.

[62] M. Thimm. Tweety: A comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation. In *Proc. KR'14*, 2014.

[63] J. Wijsen. Database repairing using updates. *ACM Transactions on Database Systems*, 30(3):722–768, 2005.

[64] A. Zamansky, O. Arieli, and K. Stefanidis. Context-aware distance semantics for inconsistent database systems. In *Proc. IPMU'14, Part II*, CCIS 443, pages 194–203. Springer, 2014.

[65] P. Zave and M. Jackson. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1):1–30, 1997.