Functional Dependencies with Predicates: What Makes the $g3$-error Easy to Compute?

Simon Vilmin[1,2](✉), Pierre Faure–Giovagnoli[2,3], Jean-Marc Petit[2], and Vasile-Marian Scuturici[2]

[1] Université de Lorraine, CNRS, LORIA, 54000 Villers-lès-Nancy, France
simon.vilmin@loria.fr
[2] Univ Lyon, INSA Lyon, CNRS, UCBL, LIRIS, Villeurbanne UMR5205, France
pierre.faure-giovagnoli@insa-lyon.fr, Jean-Marc.Petit@liris.cnrs.fr
[3] Compagnie Nationale du Rhône, Lyon, France
vasile-marian.scuturici@liris.cnrs.fr

Abstract. The notion of functional dependencies (FDs) can be used by data scientists and domain experts to confront background knowledge against data. To overcome the classical, too restrictive, satisfaction of FDs, it is possible to replace equality with more meaningful binary predicates, and use a coverage measure such as the g_3-error to estimate the degree to which a FD matches the data. It is known that the g_3-error can be computed in polynomial time if equality is used, but unfortunately, the problem becomes **NP**-complete when relying on more general predicates instead. However, there has been no analysis of which class of predicates or which properties alter the complexity of the problem, especially when going from equality to more general predicates. In this work, we provide such an analysis. We focus on the properties of commonly used predicates such as equality, similarity relations, and partial orders. These properties are: reflexivity, transitivity, symmetry, and antisymmetry. We show that symmetry and transitivity together are sufficient to guarantee that the g_3-error can be computed in polynomial time. However, dropping either of them makes the problem **NP**-complete.

Keywords: functional dependencies · g_3-error, predicates

1 Introduction

Functional dependencies (FDs) are database constraints initially devoted to database design [26]. Since then, they have been used for numerous tasks ranging from data cleaning [5] to data mining [28]. However, when dealing with real world data, FDs are also a simple yet powerful way to syntactically express background knowledge coming from domain experts [12]. More precisely, a FD $X \rightarrow A$ between a set of attributes (or features) X and another attribute A depicts a *function* of the form $f(X) = A$. In this context, asserting the existence of a function which determines A from X in a dataset amounts to testing the validity of $X \rightarrow A$ in a relation, *i.e.* to checking that *every pair* of tuples that are *equal* on X are also *equal* on A. Unfortunately, this semantics of satisfaction

M. Ojeda-Aciego et al. (Eds.): ICCS 2023, LNAI 14133, pp. 3–16, 2023.
https://doi.org/10.1007/978-3-031-40960-8_1

suffers from two major drawbacks which makes it inadequate to capture the complexity of real world data: (i) it must be checked on the whole dataset, and (ii) it uses equality.

Drawback (i) does not take into account data quality issues such as outliers, mismeasurements or mistakes, which should not impact the relevance of a FD in the data. To tackle this problem, it is customary to estimate the partial validity of a given FD with a *coverage* measure, rather than its total satisfaction. The most common of these measures is the g_3-error [8,17,21,31], introduced by Kivinen and Mannila [22]. It is the minimum proportion of tuples to remove from a relation in order to satisfy a given FD. As shown for instance by Huhtala et al. [21], the g_3-error can be computed in polynomial time for a single (classical) FD.

As for drawback (ii), equality does not always witness efficiently the closeness of two real-world values. It screens imprecisions and uncertainties that are inherent to every observation. In order to handle closeness (or difference) in a more appropriate way, numerous researches have replaced equality by *binary predicates*, as witnessed by recent surveys on relaxed FDs [6,32].

However, if predicates extend FDs in a powerful and meaningful way with respect to real-world applications, they also make computations harder. In fact, contrary to strict equality, computing the g_3-error with binary predicates becomes **NP**-complete [12,31]. In particular, it has been proven for differential [30], matching [11], metric [23], neighborhood [1], and comparable dependencies [31]. Still, there is no detailed analysis of what makes the g_3-error hard to compute when dropping equality for more flexible predicates. As a consequence, domain experts are left without any insights on which predicates they can use in order to estimate the validity of their background knowledge in their data quickly and efficiently.

This last problem constitutes the motivation for our contribution. In this work, we study the following question: *which properties of predicates make the g_3-error easy to compute?* To do so, we introduce binary predicates on each attribute of a relation scheme. Binary predicates take two values as input and return `true` or `false` depending on whether the values match a given comparison criteria. Predicates are a convenient framework to study the impact of common properties such as reflexivity, transitivity, symmetry, and antisymmetry (the properties of equality) on the hardness of computing the g_3-error. In this setting, we make the following contributions. First, we show that dropping reflexivity and antisymmetry does not make the g_3-error hard to compute. When removing transitivity, the problem becomes **NP**-complete. This result is intuitive as transitivity plays a crucial role in the computation of the g_3-error for dependencies based on similarity/distance relations [6,32]. Second, we focus on symmetry. Symmetry has attracted less attention, despite its importance in partial orders and order FDs [10,15,27]. Even though symmetry seems to have less impact than transitivity in the computation of the g_3-error, we show that when it is removed the problem also becomes **NP**-complete. This result holds in particular for ordered dependencies.

Paper Organization. In Sect. 2, we recall some preliminary definitions. Section 3 is devoted to the usual g_3-error. In Sect. 4, we introduce predicates, along with definitions for the relaxed satisfaction of a functional dependency. Section 5 investigates the problem of computing the g_3-error when equality is replaced by predicates on each attribute. In Sect. 6 we relate our results with existing extensions of FDs. We conclude in Sect. 7 with some remarks and open questions for further research.

2 Preliminaries

All the objects we consider are finite. We begin with some definitions on graphs [2] and ordered sets [9]. A *graph* G is a pair (V, E) where V is a set of *vertices* and E is a collection of pairs of vertices called *edges*. An edge of the form (u, u) is called a *loop*. The graph G is *directed* if edges are ordered pairs of elements. Unless otherwise stated, we consider *loopless undirected* graphs. Let $G = (V, E)$ be an undirected graph, and let $V' \subseteq V$. The graph $G[V'] = (V', E')$ with $E' = \{(u, v) \in E \mid \{u, v\} \subseteq V'\}$ is the graph *induced* by V' with respect to G. A *path* in G is a sequence $e_1, \ldots, e_m$ of pairwise distinct edges such that e_i and e_{i+1} share a common vertex for each $1 \leq i < m$. The *length* of a path is its number of edges. An *independent set* of G is a subset I of V such that no two vertices in I are connected by an edge of G. An independent set is *maximal* if it is inclusion-wise maximal among all independent sets. It is *maximum* if it is an independent set of maximal cardinality. Dually, a *clique* of G is a subset K of V such that every pair of distinct vertices in K are connected by an edge of G. A graph G is a *co-graph* if it has no induced subgraph corresponding to a path of length 3 (called P_4). A *partially ordered set* or *poset* is a pair $P = (V, \leq)$ where V is a set and $\leq$ a reflexive, transitive, and antisymmetric binary relation. The relation $\leq$ is called a *partial order*. If for every $x, y \in V$, $x \leq y$ or $y \leq x$ holds, $\leq$ is a *total order*. A poset P is associated to a directed graph $G(P) = (V, E)$ where $(u_i, u_j) \in E$ exactly when $u_i \neq u_j$ and $u_i \leq u_j$. An undirected graph $G = (V, E)$ is a *comparability graph* if its edges can be directed so that the resulting directed graph corresponds to a poset.

We move to terminology from database theory [24]. We use capital first letters of the alphabet (A, B, C, ...) to denote attributes and capital last letters (..., X, Y, Z) for attribute sets. Let U be a universe of attributes, and $R \subseteq U$ a relation scheme. Each attribute A in R takes value in a domain $\mathsf{dom}(A)$. The domain of R is $\mathsf{dom}(R) = \bigcup_{A \in R} \mathsf{dom}(A)$. Sometimes, especially in examples, we write a set as a concatenation of its elements (e.g. AB corresponds to $\{A, B\}$). A *tuple* over R is a mapping $t \colon R \to \mathsf{dom}(R)$ such that $t(A) \in \mathsf{dom}(A)$ for every $A \in R$. The *projection* of a tuple t on a subset X of R is the restriction of t to X, written $t[X]$. We write $t[A]$ as a shortcut for $t[\{A\}]$. A *relation* r over R is a finite set of tuples over R. A *functional dependency* (FD) over R is an expression $X \to A$ where $X \cup \{A\} \subseteq R$. Given a relation r over R, we say that r *satisfies* $X \to A$, denoted by $r \models X \to A$, if for every pair of tuples (t_1, t_2) of r, $t_1[X] = t_2[X]$ implies $t_1[A] = t_2[A]$. In case when r does not satisfy $X \to A$, we write $r \not\models X \to A$.

3 The g_3-error

This section introduces the g_3-error, along with its connection with independent sets in graphs through counterexamples and conflict-graphs [3].

Let r be a relation over R and $X \to A$ a functional dependency. The g_3*-error* quantifies the degree to which $X \to A$ holds in r. We write it as $g_3(r, X \to A)$. It was introduced by Kivinen and Mannila [22], and it is frequently used to estimate the partial validity of a FD in a dataset [6,8,12,21]. It is the minimum proportion of tuples to remove from r to satisfy $X \to A$, or more formally:

Definition 1. *Let R be a relation scheme, r a relation over R and $X \to A$ a functional dependency over R. The g_3-*error *of $X \to A$ with respect to r, denoted by $g_3(r, X \to A)$ is defined as:*

$$g_3(r, X \to A) = 1 - \frac{\mathsf{max}(\{|s| \mid s \subseteq r, s \models X \to A\})}{|r|}$$

In particular, if $r \models X \to A$, we have $g_3(r, X \to A) = 0$. We refer to the problem of computing $g_3(r, X \to A)$ as the *error validation problem* [6,31]. Its decision version reads as follows:

Error Validation Problem (EVP)
Input: A relation r over R, a FD $X \to A$, $k \in \mathbb{R}$.
Question: Is is true that $g_3(r, X \to A) \leq k$?

It is known [6,12] that there is a strong relationship between this problem and the task of computing the size of a maximum independent set in a graph:

Maximum Independent Set (MIS)
Input: A graph $G = (V, E)$, $k \in \mathbb{N}$.
Question: Does G have a maximal independent set I such that $|I| \geq k$?

To see the relationship between EVP and MIS, we need the notions of *counterexample* and *conflict-graph* [3,12]. A *counterexample* to $X \to A$ in r is a pair of tuples (t_1, t_2) such that $t_1[X] = t_2[X]$ but $t_1[A] \neq t_2[A]$. The *conflict-graph* of $X \to A$ with respect to r is the graph $\mathsf{CG}(r, X \to A) = (r, E)$ where a (possibly ordered) pair of tuples (t_1, t_2) in r belongs to E when it is a counterexample to $X \to A$ in r. An independent set of $\mathsf{CG}(r, X \to A)$ is precisely a subrelation of r which satisfies $X \to A$. Therefore, computing $g_3(r, X \to A)$ reduces to finding the size of a maximum independent set in $\mathsf{CG}(r, X \to A)$. More precisely, $g_3(r, X \to A) = 1 - \frac{|I|}{|r|}$ where I is a maximum independent set of $\mathsf{CG}(r, X \to A)$.

Example 1. Consider the relation scheme $R = \{A, B, C, D\}$ with $\mathsf{dom}(R) = \mathbb{N}$. Let r be the relation over R on the left of Fig. 1. It satisfies $BC \to A$ but not $D \to A$. Indeed, (t_1, t_3) is a counterexample to $D \to A$. The conflict-graph $\mathsf{CG}(r, D \to A)$ is given on the right of Fig. 1. For example, $\{t_1, t_2, t_6\}$ is a maximum independent set of $\mathsf{CG}(r, D \to A)$ of maximal size. We obtain:

$$g_3(r, D \to A) = 1 - \frac{|\{t_1, t_2, t_6\}|}{|r|} = 0.5$$

In other words, we must remove half of the tuples of r in order to satisfy $D \to A$.

r	A	B	C	D
t_1	1	2	1	5
t_2	1	1	2	5
t_3	2	1	1	5
t_4	3	2	3	5
t_5	2	3	4	5
t_6	4	4	5	6

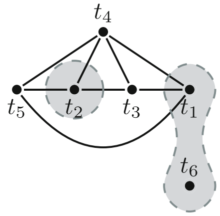

Fig. 1. The relation r and the conflict-graph $\mathsf{CG}(r, D \to A)$ of Example 1.

However, MIS is an **NP**-complete problem [13] while computing $g_3(r, X \to A)$ takes polynomial time in the size of r and $X \to A$ [21]. This difference is due to the properties of equality, namely reflexivity, transitivity, symmetry and antisymmetry. They make $\mathsf{CG}(r, X \to A)$ a disjoint union of complete k-partite graphs, and hence a co-graph [12]. In this class of graphs, solving MIS is polynomial [14]. This observation suggests to study in greater detail the impact of such properties on the structure of conflict-graphs. First, we need to introduce predicates to relax equality, and to define a more general version of the error validation problem accordingly.

4 Predicates to Relax Equality

In this section, in line with previous researches on extensions of functional dependencies [6,32], we equip each attribute of a relation scheme with a binary predicate. We define the new g_3-error and the corresponding error validation problem.

Let R be a relation scheme. For each $A \in R$, let $\phi_A \colon \mathsf{dom}(A) \times \mathsf{dom}(A) \to \{\texttt{true}, \texttt{false}\}$ be a predicate. For instance, the predicate ϕ_A can be equality, a distance, or a similarity relation. We assume that predicates are black-box oracles that can be computed in polynomial time in the size of their input.

Let Φ be a set of predicates, one for each attribute in R. The pair (R, Φ) is a *relation scheme with predicates.* In a relation scheme with predicates, relations and FDs are unchanged. However, the way a relation satisfies (or not) a FD can easily be adapted to Φ.

Definition 2 ***(Satisfaction with predicates).*** *Let (R, Φ) be a relation scheme with predicates, r a relation and $X \to A$ a functional dependency both over (R, Φ). The relation r* satisfies *$X \to A$ with respect to Φ, denoted by $r \models_\Phi X \to A$, if for every pair of tuples (t_1, t_2) of r, the following formula holds:*

$$\left(\bigwedge_{B \in X} \phi_B(t_1[B], t_2[B]) \right) \implies \phi_A(t_1[A], t_2[A])$$

A new version of the g_3-error adapted to Φ is presented in the following definition.

Definition 3. *Let* (R, Φ) *be a relation scheme with predicates,* r *be a relation over* (R, Φ) *and* $X \to A$ *a functional dependency over* (R, Φ)*. The* g_3-error with predicates *of* $X \to A$ *with respect to* r*, denoted by* $g_3^{\Phi}(r, X \to A)$ *is defined as:*

$$g_3^{\Phi}(r, X \to A) = 1 - \frac{\mathsf{max}(\{|s| \mid s \subseteq r, s \models_{\Phi} X \to A\})}{|r|}$$

From the definition of $g_3^{\Phi}(r, X \to A)$, we derive the extension of the error validation problem from equality to predicates:

Error Validation Problem with Predicates (EVPP)
Input: A relation r over (R, Φ), a FD $X \to A$ over R, $k \in \mathbb{R}$.
Question: Is it true that $g_3^{\Phi}(r, X \to A) \leq k$?

Observe that according to the definition of satisfaction with predicates (Definition 2), counterexamples and conflict-graphs remain well-defined. However, for a given predicate ϕ_A, $\phi_A(x, y) = \phi_A(y, x)$ needs not be true in general, meaning that we have to consider ordered pairs of tuples. That is, an ordered pair of tuples (t_1, t_2) in r is a counterexample to $X \to A$ if $\bigwedge_{B \in X} \phi_B(t_1[B], t_2[B]) = \texttt{true}$ but $\phi_A(t_1[A], t_2[A]) \neq \texttt{true}$.

We call $\mathsf{CG}_{\Phi}(r, X \to A)$ the conflict-graph of $X \to A$ in r. In general, $\mathsf{CG}_{\Phi}(r, X \to A)$ is directed. It is undirected if the predicates of Φ are symmetric (see Sect. 5). In particular, computing $g_3^{\Phi}(r, X \to A)$ still amounts to finding the size of a maximum independent set in $\mathsf{CG}_{\Phi}(r, X \to A)$.

Example 2. We use the relation of Fig. 1. Let $\Phi = \{\phi_A, \phi_B, \phi_C, \phi_D\}$ be the collection of predicates defined as follows, for every $x, y \in \mathbb{N}$:

- $\phi_A(x, y) = \phi_B(x, y) = \phi_C(x, y) = \texttt{true}$ if and only if $|x - y| \leq 1$. Thus, ϕ_A is reflexive and symmetric but not transitive (see Sect. 5),
- ϕ_D is the equality.

The pair (R, Φ) is a relation scheme with predicates. We have $r \models_{\Phi} AB \to D$ but $r \not\models_{\Phi} C \to A$. In Fig. 2, we depict $\mathsf{CG}_{\Phi}(r, C \to A)$. A maximum independent set of this graph is $\{t_1, t_2, t_3, t_5\}$. We deduce

$$g_3^{\Phi}(r, C \to A) = 1 - \frac{|\{t_1, t_2, t_3, t_5\}|}{|r|} = \frac{1}{3}$$

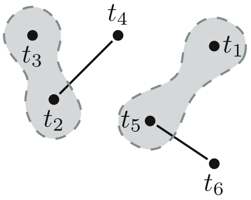

Fig. 2. The conflict-graph $\mathsf{CG}_{\Phi}(r, C \rightarrow A)$ of Example 2.

Thus, there is also a strong relationship between EVPP and MIS, similar to the one between EVP and MIS. Nonetheless, unlike EVP, the problem EVPP is **NP**-complete [31]. In the next section, we study this gap of complexity between EVP and EVPP via different properties of predicates.

5 Predicates Properties in the g_3-error

In this section, we study properties of binary predicates that are commonly used to replace equality. We show how each of them affects the error validation problem.

First, we define the properties of interest in this paper. Let (R, Φ) be a relation scheme with predicates. Let $A \in R$ and ϕ_A be the corresponding predicate. We consider the following properties:

(`ref`) $\phi_A(x, x) = \texttt{true}$ for all $x \in \mathsf{dom}(A)$ (reflexivity)
(`tra`) for all $x, y, z \in \mathsf{dom}(A)$, $\phi_A(x, y) = \phi_A(y, z) = \texttt{true}$ implies $\phi_A(x, z) = \texttt{true}$ (transitivity)
(`sym`) for all $x, y \in \mathsf{dom}(A)$, $\phi_A(x, y) = \phi_A(y, x)$ (symmetry)
(`asym`) for all $x, y \in \mathsf{dom}(A)$, $\phi_A(x, y) = \phi_A(y, x) = \texttt{true}$ implies $x = y$ (antisymmetry).

Note that symmetry and antisymmetry together imply transitivity, as $\phi_A(x, y) = \texttt{true}$ entails $x = y$.

As a first step, we show that symmetry and transitivity are sufficient to make EVPP solvable in polynomial time. In fact, we prove that the resulting conflict-graph is a co-graph, as with equality.

Theorem 1. *The problem* EVPP *can be solved in polynomial time if the predicates used on each attribute are transitive (*`tra`*) and symmetric (*`sym`*).*

Proof. Let (R, Φ) be a relation scheme with predicates. Let r be relation over (R, Φ) and $X \rightarrow A$ be a functional dependency, also over (R, Φ). We assume that each predicate in Φ is transitive and symmetric. We show how to compute the size of a maximum independent set of $\mathsf{CG}_{\Phi}(r, X \rightarrow A)$ in polynomial time.

As ϕ_A is not necessarily reflexive, a tuple t in r can produce a counter-example (t, t) to $X \rightarrow A$. Indeed, it may happen that $\phi_B(t[B], t[B]) = \texttt{true}$ for each $B \in X$, but $\phi_A(t[A], t[A]) = \texttt{false}$. However, it follows that t never belongs to a subrelation s of r satisfying $s \models_{\Phi} X \rightarrow A$. Thus, let $r' = r \setminus \{t \in r \mid \{t\} \not\models_{\Phi} X \rightarrow A\}$. Then, a subrelation of r satisfies $X \rightarrow A$ if and only if it is an independent set of $\mathsf{CG}_{\Phi}(r, X \rightarrow A)$ if and only if it is an independent set of $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$. Consequently, computing $g_3^{\Phi}(r, X \rightarrow A)$ is solving MIS in $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$.

We prove now that $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$ is a co-graph. Assume for contradiction that $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$ has an induced path P with 4 elements, say t_1, t_2, t_3, t_4 with edges (t_1, t_2), (t_2, t_3) and (t_3, t_4). Remind that edges of $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$ are counterexamples to $X \rightarrow A$ in r'. Hence, by symmetry and transitivity of the predicates of Φ, we deduce that for each pair (i, j) in $\{1, 2, 3, 4\}$, $\bigwedge_{B \in X} \phi_B(t_i[B], t_j[B]) = \texttt{true}$. Thus, we have $\bigwedge_{B \in X} \phi_B(t_3[B], t_1[B]) = \bigwedge_{B \in X} \phi_B(t_1[B], t_4[B]) = \texttt{true}$. However, neither (t_1, t_3) nor (t_1, t_4) belong to $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$ since P is an induced path by assumption. Thus, $\phi_A(t_3[A], t_1[A]) = \phi_A(t_1[A], t_4[A]) = \texttt{true}$ must hold. Nonetheless, the transitivity of ϕ_A implies $\phi_A(t_3[A], t_4[A]) = \texttt{true}$, a contradiction with (t_3, t_4) being an edge of $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$. We deduce that $\mathsf{CG}_{\Phi}(r', X \rightarrow A)$ cannot contain an induced P_4, and that it is indeed a co-graph. As MIS can be solved in polynomial time for co-graphs [14], the theorem follows. □

One may encounter non-reflexive predicates when dealing with strict orders or with binary predicates derived from `SQL` equality. In the 3-valued logic of `SQL`, comparing the `null` value with itself evaluates to `false` rather than `true`. With this regard, it could be natural for domain experts to use a predicate which is transitive, symmetric and reflexive almost everywhere but on the `null` value. This would allow to deal with missing information without altering the data.

The previous proof heavily makes use of transitivity, which has a strong impact on the edges belonging to the conflict-graph. Intuitively, conflict-graphs can become much more complex when transitivity is dropped. Indeed, we prove an intuitive case: when predicates are not required to be transitive, EVPP becomes intractable.

Theorem 2. *The problem* EVPP *is* **NP***-complete even when the predicates used on each attribute are symmetric (*`sym`*) and reflexive (*`ref`*).*

The proof is omitted due to space limitations, it can be found in [33]. It is a reduction from the problem (dual to MIS) of finding the size of a maximum clique in general graphs. It uses arguments similar to the proof of Song et al. [31] showing the **NP**-completeness of EVPP for comparable dependencies.

We turn our attention to the case where symmetry is dropped from the predicates. In this context, conflict-graphs are directed. Indeed, an ordered pair of tuples (t_1, t_2) may be a counterexample to a functional dependency, but not (t_2, t_1). Yet, transitivity still contributes to constraining the structure of conflict-graphs, as suggested by the following example.

Example 3. We consider the relation of Example 1. We equip A, B, C, D with the following predicates:

- $\phi_C(x, y) = \texttt{true}$ if and only if $x \leq y$

- $\phi_A(x, y)$ is defined by

$$\phi_A(x,y) = \begin{cases} \texttt{true} & \text{if } x = y \\ \texttt{true} & \text{if } x = 1 \text{ and } y \in \{2,4\} \\ \texttt{true} & \text{if } x = 3 \text{ and } y = 4 \\ \texttt{false} & \text{otherwise.} \end{cases}$$

- ϕ_B and ϕ_D are the equality.

Let $\Phi = \{\phi_A, \phi_B, \phi_C, \phi_D\}$. The conflict-graph $\mathsf{CG}_\Phi(C \to A)$ is represented in Fig. 3. Since ϕ_C is transitive, we have $\phi_C(t_3[C], t_j[C]) = \texttt{true}$ for each tuple t_j of r. Moreover, $\phi_A(t_3[A], t_6[A]) = \texttt{false}$ since (t_3, t_6) is a counterexample to $C \to A$. Therefore, the transitivity of ϕ_A implies either $\phi_A(t_3[A], t_4[A]) = \texttt{false}$ or $\phi_A(t_4[A], t_6[A]) = \texttt{false}$. Hence, at least one of (t_3, t_4) and (t_4, t_6) must be a counterexample to $C \to A$ too. In the example, this is (t_3, t_4).

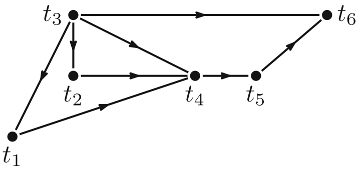

Fig. 3. The conflict-graph $\mathsf{CG}_\Phi(r, C \to A)$ of Example 3.

Nevertheless, if transitivity constrains the complexity of the graph, dropping symmetry still allows new kinds of graph structures. Indeed, in the presence of symmetry, a conflict-graph cannot contain induced paths with more than 3 elements because of transitivity. However, such paths may exist when symmetry is removed.

Example 4. In the previous example, the tuples t_2, t_4, t_5, t_6 form an induced P_4 of the underlying undirected graph of $\mathsf{CG}_\Phi(r, C \to A)$, even though ϕ_A and ϕ_C enjoy transitivity.

Therefore, we are left with the following intriguing question: can the loss of symmetry be used to break transitivity, and offer conflict-graphs a structure sufficiently complex to make EVPP intractable? The next theorem answers this question affirmatively.

Theorem 3. *The problem* EVPP *is* **NP***-complete even when the predicates used on each attribute are transitive (*`tra`*), reflexive (*`ref`*), and antisymmetric (*`asym`*).*

The proof is omitted due to space limitations. It is given in [33]. It is a reduction from MIS in 2-subdivision graphs [29].

Theorem 1, Theorem 2 and Theorem 3 characterize the complexity of EVPP for each combination of predicates properties. In the next section, we discuss the granularity of these, and we use them as a framework to compare the complexity of EVPP for some known extensions of functional dependencies.

6 Discussions

Replacing equality with various predicates to extend the semantics of classical functional dependencies is frequent [6,32]. Our approach offers to compare these extensions on EVPP within a unifying framework based on the properties of the predicates they use. We can summarize our results with the hierarchy of classes of predicates given in Fig. 4.

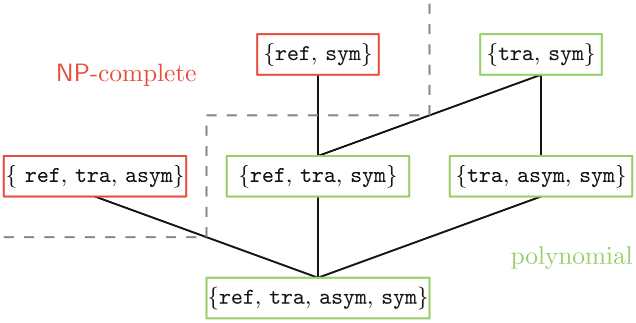

Fig. 4. Complexity of EVPP with respect to the properties of predicates.

Regarding the computation of the g_3-error, most existing works have focused on similarity/distance predicates. First, the g_3-error can be computed in polynomial time for classical functional dependencies [20]. Then, Song et al. [31] show that EVPP is **NP**-complete for a broad range of extensions of FDs which happen to be reflexive (`ref`) and symmetric (`sym`) predicates, which coincides with Theorem 2. However, they do not study predicate properties as we do in this paper. More precisely, they identify the hardness of EVPP for differential [30], matching [11], metric [23], neighborhood [1], and comparable dependencies [31]. For some of these dependencies, predicates may be defined over sets of attributes. Using one predicate per attribute and taking their conjunction is a particular case of predicate on attribute sets.

Some extensions of FDs use partial orders as predicates. This is the case of ordered dependencies [10,15], ordered FDs [27], and also of some sequential dependencies [16] and denial constraints [4] for instance. To our knowledge, the role of symmetry in EVPP has received little attention. For sequential dependencies [16], a measure different than the g_3-error have been used. The predicates of Theorem 3 are reflexive, transitive and antisymmetric. Hence they are partial orders. Consequently, the FDs in this context are *ordered functional dependencies* as defined by Ng [27]. We obtain the following corollary:

Corollary 1. EVPP *is* **NP**-*complete for ordered functional dependencies.*

Ordered functional dependencies are a restricted case of ordered dependencies [15], sequential dependencies [16], and denial constraints [4] (see [32]). The hardness of computing the g_3-error for these dependencies follows from Corollary 1.

The hierarchy depicts quite accurately the current knowledge about EVPP and the delimitation between tractable and intractable cases. However, this analysis may require further refinements. Indeed, there may be particular types of FDs with predicates where EVPP is tractable in polynomial time, even though their predicates belong to a class for which the problem is **NP**-complete. For instance, assume that each attribute A in R is equipped with a *total* order ϕ_A. We show in Proposition 1 and Corollary 2 that in this case, EVPP can be solved in polynomial time, even though the predicates are reflexive, transitive and antisymmetric.

Proposition 1. *Let (R, Φ) be a relation scheme with predicates. Then,* EVPP *can be solved in polynomial time for a given FD $X \rightarrow A$ if ϕ_B is transitive for each $B \in X$ and ϕ_A is a total order.*

Proof. Let (R, Φ) be a relation scheme with predicates and $X \rightarrow A$ a functional dependency. Assume that ϕ_B is transitive for each $B \in X$ and that ϕ_A is a total order. Let r be a relation over (R, Φ). Let $G = (r, E)$ be the undirected graph underlying $\mathsf{CG}_\Phi(r, X \rightarrow A)$, that is, $(t_i, t_j) \in E$ if and only if (t_i, t_j) or (t_j, t_i) is an edge of $\mathsf{CG}_\Phi(r, X \rightarrow A)$.

We show that G is a comparability graph. To do so, we associate the following predicate $\leq$ to $\mathsf{CG}_\Phi(r, X \rightarrow A)$: for each pair t_i, t_j of tuples of r, $t_i \leq t_i$ and $t_i \leq t_j$ if (t_i, t_j) is a counterexample to $X \rightarrow A$. We show that $\leq$ is a partial order:

- *reflexivity.* It follows by definition.
- *antisymmetry.* We use contrapositive. Let t_i, t_j be two distinct tuples of r and assume that (t_i, t_j) belongs to $\mathsf{CG}_\Phi(r, X \rightarrow A)$. We need to prove that (t_j, t_i) does not belong to $\mathsf{CG}_\Phi(r, X \rightarrow A)$, *i.e.* it is not a counterexample to $X \rightarrow A$. First, $(t_i, t_j) \in \mathsf{CG}_\Phi(r, X \rightarrow A)$ implies that $\phi_A(t_i[A], t_j[A]) = \texttt{false}$. Then, since ϕ_A is a total order, $\phi_A(t_j[A], t_i[A]) = \texttt{true}$. Consequently, (t_j, t_i) cannot belong to $\mathsf{CG}_\Phi(r, X \rightarrow A)$ and $\leq$ is antisymmetric.
- *transitivity.* Let t_i, t_j, t_k be tuples of r such that (t_i, t_j) and (t_j, t_k) are in $\mathsf{CG}_\Phi(r, X \rightarrow A)$. Applying transitivity, we have that $\bigwedge_{B \in X} \phi_B(t_i[B], t_k[B]) = \texttt{true}$. We show that $\phi_A(t_i[A], t_k[A]) = \texttt{false}$. Since (t_i, t_j) is a counterexample to $X \rightarrow A$, we have $\phi_A(t_i[A], t_j[A]) = \texttt{false}$. As ϕ_A is a total order, we deduce that $\phi_A(t_j[A], t_i[A]) = \texttt{true}$. Similarly, we obtain $\phi_A(t_k[A], t_j[A]) = \texttt{true}$. As ϕ_A is transitive, we derive $\phi_A(t_k[A], t_i[A]) = \texttt{true}$. Now assume for contradiction that $\phi_A(t_i[A], t_k[A]) = \texttt{true}$. Since, $\phi_A(t_k[A], t_j[A]) = \texttt{true}$, we derive $\phi_A(t_i[A], t_j[A]) = \texttt{true}$ by transitivity of ϕ_A, a contradiction. Therefore, $\phi_A(t_i[A], t_k[A]) = \texttt{false}$. Using the fact that $\bigwedge_{B \in X} \phi_B(t_i[B], t_k[B]) = \texttt{true}$, we conclude that (t_i, t_k) is also a counterexample to $X \rightarrow A$. The transitivity of $\leq$ follows. □

Consequently, $\leq$ is a partial order and G is indeed a comparability graph. Since MIS can be solved in polynomial time for comparability graphs [18], the result follows.

We can deduce the following corollary on total orders, that can be used for ordered dependencies.

Corollary 2. *Let (R, Φ) be a relation scheme with predicates. Then,* EVPP *can be solved in polymomial time if each predicate in Φ is a total order.*

In particular, Golab et al. [16] proposed a polynomial-time algorithm for a variant of g_3 applied to a restricted type of sequential dependencies using total orders on each attribute.

7 Conclusion and Future Work

In this work, we have studied the complexity of computing the g_3-error when equality is replaced by more general predicates. We studied four common properties of binary predicates: reflexivity, symmetry, transitivity, and antisymmetry. We have shown that when symmetry and transitivity are taken together, the g_3-error can be computed in polynomial time. Transitivity strongly impacts the structure of the conflict-graph of the counterexamples to a functional dependency in a relation. Thus, it comes as no surprise that dropping transitivity makes the g_3-error hard to compute. More surprisingly, removing symmetry instead of transitivity leads to the same conclusion. This is because deleting symmetry makes the conflict-graph directed. In this case, the orientation of the edges weakens the impact of transitivity, thus allowing the conflict-graph to be complex enough to make the g_3-error computation problem intractable.

We believe our approach sheds new light on the problem of computing the g_3-error, and that it is suitable for estimating the complexity of this problem when defining new types of FDs, by looking at the properties of predicates used to compare values.

We highlight now some research directions for future works. In a recent paper [25], Livshits et al. study the problem of computing optimal repairs in a relation with respect to a set of functional dependencies. A repair is a collection of tuples which does not violate a prescribed set of FDs. It is optimal if it is of maximal size among all possible repairs. Henceforth, there is a strong connection between the problem of computing repairs and computing the g_3-error with respect to a collection of FDs. In their work, the authors give a dichotomy between tractable and intractable cases based on the structure of FDs. In particular, they use previous results from Gribkoff et al. [19] to show that the problem is already **NP**-complete for 2 FDs in general. In the case where computing an optimal repair can be done in polynomial time, it would be interesting to use our approach and relax equality with predicates in order to study the tractability of computing the g_3-error on a collection of FDs with relaxed equality.

From a practical point of view, the exact computation of the g_3-error is extremely expensive in large datasets. Recent works [7,12] have proposed to use approximation algorithms to compute the g_3-error both for equality and predicates. It could be of interest to identify properties or classes of predicates where more efficient algorithms can be adopted. It is also possible to extend the existing algorithms calculating the classical g_3-error (see *e.g.* [21]). They use the projection to identify equivalence classes among values of A and X. However, when dropping transitivity (for instance in similarity predicates), separating the values of a relation into *"similar classes"* requires to devise a new projection operation, a seemingly tough but fascinating problem to investigate.

Acknowledgment. we thank the reviewers for their constructive feedback and the Datavalor initiative of Insavalor (subsidiary of INSA Lyon) for funding part of this work.

References

1. Bassée, R., Wijsen, J.: Neighborhood dependencies for prediction. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 562–567. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45357-1_59
2. Berge, C.: Graphs and Hypergraphs. North-Holland Pub. Co., Amsterdam (1973)
3. Bertossi, L.: Database repairing and consistent query answering. Synth. Lect. Data Manag. **3**(5), 1–121 (2011)
4. Bertossi, L., Bravo, L., Franconi, E., Lopatenko, A.: Complexity and approximation of fixing numerical attributes in databases under integrity constraints. In: Bierman, G., Koch, C. (eds.) DBPL 2005. LNCS, vol. 3774, pp. 262–278. Springer, Heidelberg (2005). https://doi.org/10.1007/11601524_17
5. Bohannon, P., Fan, W., Geerts, F., Jia, X., Kementsietsidis, A.: Conditional functional dependencies for data cleaning. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 746–755. IEEE (2007)
6. Caruccio, L., Deufemia, V., Polese, G.: Relaxed functional dependencies-a survey of approaches. IEEE Trans. Knowl. Data Eng. **28**(1), 147–165 (2015)
7. Caruccio, L., Deufemia, V., Polese, G.: On the discovery of relaxed functional dependencies. In: Proceedings of the 20th International Database Engineering & Applications Symposium, pp. 53–61 (2016)
8. Cormode, G., Golab, L., Flip, K., McGregor, A., Srivastava, D., Zhang, X.: Estimating the confidence of conditional functional dependencies. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, pp. 469–482. Association for Computing Machinery, New York (2009). https://doi.org/10.1145/1559845.1559895
9. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (2002)
10. Dong, J., Hull, R.: Applying approximate order dependency to reduce indexing space. In: Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data, pp. 119–127 (1982)
11. Fan, W.: Dependencies revisited for improving data quality. In: Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 159–170 (2008)
12. Faure-Giovagnoli, P., Petit, J.M., Scuturici, V.M.: Assessing the existence of a function in a dataset with the g3 indicator. In: IEEE International Conference on Data Engineering (2022)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability, vol. 174. Freeman, San Francisco (1979)

14. Giakoumakis, V., Roussel, F., Thuillier, H.: On p_4-tidy graphs. Disc. Math. Theor. Comput. Sci. **1**, 17–41 (1997)
15. Ginsburg, S., Hull, R.: Order dependency in the relational model. Theor. Comput. Sci. **26**(1–2), 149–195 (1983)
16. Golab, L., Karloff, H., Korn, F., Saha, A., Srivastava, D.: Sequential dependencies. Proc. VLDB Endow. **2**(1), 574–585 (2009)
17. Golab, L., Karloff, H., Korn, F., Srivastava, D., Yu, B.: On generating near-optimal tableaux for conditional functional dependencies. Proc. VLDB Endow. **1**(1), 376–390 (2008)
18. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Elsevier, Amsterdam (2004)
19. Gribkoff, E., Van den Broeck, G., Suciu, D.: The most probable database problem. In: Proceedings of the First International Workshop on Big Uncertain Data (BUDA), pp. 1–7 (2014)
20. Huhtala, Y., Karkkainen, J., Porkka, P., Toivonen, H.: Efficient discovery of functional and approximate dependencies using partitions. In: Proceedings 14th International Conference on Data Engineering, pp. 392–401. IEEE (1998)
21. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: Tane: an efficient algorithm for discovering functional and approximate dependencies. Comput. J. **42**(2), 100–111 (1999)
22. Kivinen, J., Mannila, H.: Approximate inference of functional dependencies from relations. Theor. Comput. Sci. **149**(1), 129–149 (1995)
23. Koudas, N., Saha, A., Srivastava, D., Venkatasubramanian, S.: Metric functional dependencies. In: 2009 IEEE 25th International Conference on Data Engineering, pp. 1275–1278. IEEE (2009)
24. Levene, M., Loizou, G.: A Guided Tour of Relational Databases and Beyond. Springer, Heidelberg (2012). https://doi.org/10.1007/978-0-85729-349-7
25. Livshits, E., Kimelfeld, B., Roy, S.: Computing optimal repairs for functional dependencies. ACM Trans. Datab. Syst. (TODS) **45**(1), 1–46 (2020)
26. Mannila, H., Räihä, K.J.: The Design of Relational Databases. Addison-Wesley Longman Publishing Co., Inc., Boston (1992)
27. Ng, W.: An extension of the relational data model to incorporate ordered domains. ACM Trans. Datab. Syst. (TODS) **26**(3), 344–383 (2001)
28. Novelli, N., Cicchetti, R.: Functional and embedded dependency inference: a data mining point of view. Inf. Syst. **26**(7), 477–506 (2001)
29. Poljak, S.: A note on stable sets and colorings of graphs. Commentationes Mathematicae Universitatis Carolinae **15**(2), 307–309 (1974)
30. Song, S.: Data dependencies in the presence of difference. Ph.D. thesis, Hong Kong University of Science and Technology (2010)
31. Song, S., Chen, L., Philip, S.Y.: Comparable dependencies over heterogeneous data. VLDB J. **22**(2), 253–274 (2013)
32. Song, S., Gao, F., Huang, R., Wang, C.: Data dependencies extended for variety and veracity: a family tree. IEEE Trans. Knowl. Data Eng. **34**, 4717–4736 (2020). https://doi.org/10.1109/TKDE.2020.3046443
33. Vilmin, S., Faure-Giovagnoli, P., Petit, J.M., Scuturici, V.M.: Functional dependencies with predicates: what makes the g_3-error easy to compute? arXiv preprint arXiv:2306.09006 (2023)