Counting Database Repairs Entailing a Query: The Case of Functional Dependencies

(Discussion Paper)

Marco Calautti[1], Ester Livshits[2], Andreas Pieris[2,3] and Markus Schneider[2]

[1]*DISI, University of Trento, Italy*
[2]*University of Edinburgh, UK*
[3]*University of Cyprus, Cyprus*

Abstract
A key task in the context of consistent query answering is to count the number of repairs that entail the query, with the ultimate goal being a precise data complexity classification. This has been achieved in the case of primary keys and self-join-free conjunctive queries (CQs) via an FP/♯P-complete dichotomy. We lift this result to the more general case of functional dependencies (FDs). Another important task in this context is whenever the counting problem in question is intractable, to classify it as approximable, i.e., the target value can be efficiently approximated with error guarantees via a fully polynomial-time randomized approximation scheme (FPRAS), or as inapproximable. Although for primary keys and CQs (even with self-joins) the problem is always approximable, we prove that this is not the case for FDs. We show, however, that the class of FDs with a left-hand side chain forms an island of approximability. We see these results, apart from being interesting in their own right, as crucial steps towards a complete classification of approximate counting of repairs in the case of FDs and self-join-free CQs.

1. Introduction

A database is inconsistent if it does not satisfy its integrity constraints. There is a consensus that inconsistency is a real-life phenomenon that arises due to many reasons such as integration of conflicting sources. With the aim of obtaining conceptually meaningful answers to queries posed over inconsistent databases, Arenas, Bertossi, and Chomicki introduced in the late 1990s the notion of Consistent Query Answering (CQA) [1]. The key elements underlying CQA are (i) the notion of *(database) repair* of an inconsistent database D, that is, a consistent database whose difference with D is somehow minimal, and (ii) the notion of query answering based on *certain answers*, that is, answers that are entailed by every repair.

Example 1. *Consider the relation name* $\mathrm{Employee}(\mathrm{id}, \mathrm{name}, \mathrm{dept})$ *that comes with the constraint that the attribute* id *functionally determines* name *and* dept. *Consider also the database D consisting of the tuples: (1,* Bob, HR*), (1,* Bob, IT*), (2,* Alice, IT*), (2,* Tim, IT*). It is easy to see that D is inconsistent since we are uncertain about Bob's department, and the name of the employee with id* 2. *To devise a repair, we need to keep one tuple from each conflicting pair, which leads to a*

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy
marco.calautti@unitn (M. Calautti); ster.livshits@ed.ac.uk (E. Livshits); apieris@inf.ed.ac.uk (A. Pieris); m.schneider@ed.ac.uk (M. Schneider)

maximal subset of D that is consistent. Observe now that the query that asks whether employees 1 and 2 work in the same department is true only in two out of four repairs, and thus, not entailed. ■

Counting Repairs Entailing a Query. A key task in this context is to count the number of repairs of an inconsistent database D w.r.t. a set Σ of constraints that entail a given query Q; for clarity, we base our discussion on Boolean queries. Given a set Σ of constraints and a query Q, the problem $\sharp\mathsf{Repairs}(\Sigma, Q)$ that takes as input a database D, and asks for the number of repairs of D w.r.t. Σ that entail Q, can be tractable or intractable depending on the shape of Σ and Q. This leads to the natural question whether we can establish a complete classification, i.e., for every Σ and Q, classify $\sharp\mathsf{Repairs}(\Sigma, Q)$ as tractable or intractable by simply inspecting Σ and Q. We already know from [2] that for a set Σ of primary keys, and a self-join-free conjunctive queries (SJFCQs) query Q, $\sharp\mathsf{Repairs}(\Sigma, Q)$ is either in FP or $\sharp$P-complete, and we can determine in polynomial time, by simply analyzing Σ and Q, which complexity statement holds. However, the dichotomy result in [2] does not apply when we consider the more general class of functional dependencies (FDs). This brings us to the following question:

Question 1: *Can we lift the dichotomy result for primary keys and SJFCQs to the more general case of functional dependencies?*

The closest known result related to Question 1 is for the problem $\sharp\mathsf{Repairs}(\Sigma)$, where Σ is a set of FDs, that given a database D, asks for the number of repairs of D w.r.t. Σ (without considering a query). In particular, we know from [3] that whenever Σ has a so-called left-hand side (LHS, for short) chain (up to equivalence), $\sharp\mathsf{Repairs}(\Sigma)$ is in FP; otherwise, it is $\sharp$P-complete.

Approximate Counting. Another key task is to classify $\sharp\mathsf{Repairs}(\Sigma, Q)$, whenever it is intractable, as approximable via a so-called fully polynomial-time randomized approximation scheme (FPRAS), or as inapproximable.

For a set Σ of primary keys, and a CQ Q (even with self-joins), $\sharp\mathsf{Repairs}(\Sigma, Q)$ is always approximable [4]. The next question is whether we can obtain a full classification for FDs:

Question 2: *For a set Σ of FDs, and an SJFCQ Q, can we determine whether $\sharp\mathsf{Repairs}(\Sigma, Q)$ admits an FPRAS by inspecting Σ and Q?*

Summary of Contributions. Concerning Question (1), we lift the dichotomy of [2] for primary keys and SJFCQs to the general case of FDs (Theorem 4). Concerning Question (2), although we do not establish a complete classification (which would resolve a challenging open problem), we show that, for every set Σ of FDs with an LHS chain (up to equivalence) and a CQ Q (even with self-joins), $\sharp\mathsf{Repairs}(\Sigma, Q)$ admits an FPRAS. On the other hand, we show that there is a very simple set Σ of FDs such that, for every SJFCQ Q, $\sharp\mathsf{Repairs}(\Sigma, Q)$ does not admit an FPRAS (under a standard complexity assumption).

2. Preliminaries

We consider the disjoint countably infinite sets $\mathbf{C}$ and $\mathbf{V}$ of *constants* and *variables*, respectively. For $n > 0$, let $[n]$ be the set $\{1, \ldots, n\}$, and for a finite set S, let $\sharp S$ be the cardinality of S.

Relational Databases. A *schema* $\mathbf{S}$ is a finite set of relation names with associated arity; we write R/n to denote that R has arity $n > 0$. Each relation name R/n is associated with a tuple of distinct attribute names $(A_1, \dots, A_n)$; we write $\mathsf{att}(R)$ for the set $\{A_1, \dots, A_n\}$. A *position* of $\mathbf{S}$ is a pair (R, A), where $R \in \mathbf{S}$ and $A \in \mathsf{att}(R)$, that essentially identifies the attribute A of R. A *atom* α over $\mathbf{S}$ is an expression of the form $R(t_1, \dots, t_n)$, where $R/n \in \mathbf{S}$, and $t_i \in \mathbf{C} \cup \mathbf{V}$ for each $i \in [n]$. A *fact* is an atom mentioning only constants; we may say R-atom/fact to indicate that the relation name is R. For an atom $\alpha = R(t_1, \dots, t_n)$, with $(A_1, \dots, A_n)$ being the tuple of attribute names of R, we write $\alpha[A_i]$ for the term t_i. A *database* D over $\mathbf{S}$ is a finite set of facts over $\mathbf{S}$. We write D_R, for $R \in \mathbf{S}$, for the database $\{R(\bar{t}) \mid R(\bar{t}) \in D\}$. The *active domain* of D, denoted $\mathsf{dom}(D)$, is the set of constants occurring in D.

Functional Dependencies. A *functional dependency* (FD) ϕ over a schema $\mathbf{S}$ is an expression of the form $R : X \to Y$, where $R/n \in \mathbf{S}$ and $X, Y \subseteq \mathsf{att}(R)$; we say R-FD to indicate that the relation name is R. We call ϕ a *key* if $X \cup Y = \mathsf{att}(R)$. Given a set Σ of FDs over $\mathbf{S}$, we write Σ_R, for $R \in \mathbf{S}$, for the set $\{\phi \in \Sigma \mid \phi \text{ is an } R\text{-FD}\}$. We call Σ a set of *primary keys* if it consists only of keys, and $|\Sigma_R| \leq 1$ for each $R \in \mathbf{S}$. A database D over $\mathbf{S}$ satisfies an FD ϕ over $\mathbf{S}$, denoted $D \models \phi$, if, for every two R-facts $f, g \in D$ the following holds: $f[A] = g[A]$ for every $A \in X$ implies $f[B] = g[B]$ for every $B \in Y$. We say that D is *consistent* w.r.t. a set Σ of FDs, written $D \models \Sigma$, if $D \models \phi$ for every $\phi \in \Sigma$; otherwise, D is *inconsistent* w.r.t. Σ. Two sets of FDs Σ, Σ' are *equivalent* if, for every database D, $D \models \Sigma$ iff $D \models \Sigma'$.

Conjunctive Queries. A *(Boolean) conjunctive query* (CQ) Q over $\mathbf{S}$ is an expression of the form $\exists \bar{y}_1, \dots, \exists \bar{y}_n \; R_1(\bar{y}_1) \wedge \cdots \wedge R_n(\bar{y}_n)$, where each $R_i(\bar{y}_i)$, for $i \in [n]$, is an atom over $\mathbf{S}$. With abuse of notation we may treat Q as the *set* of its atoms. The query Q is a *self-join-free* CQ (SJFCQ) if it mentions every relation name of $\mathbf{S}$ at most once. Let $\mathsf{var}(Q)$ and $\mathsf{const}(Q)$ be the set of variables and constants in Q, respectively. A *homomorphism* from Q to a database D is a function $h : \mathsf{var}(Q) \cup \mathsf{const}(Q) \to \mathsf{dom}(D)$, which is the identity over $\mathbf{C}$, such that $R_i(h(\bar{y}_i)) \in D$ for $i \in [n]$. We say that D *entails* Q, and write $D \models Q$ if there exists a homomorphism from Q to D. We point out that in this paper we are only dealing with Boolean queries, but all the presented results can be generalized to non-Boolean CQs.

Database Repairs. Given a database D and a set Σ of FDs, both over a schema $\mathbf{S}$, a *repair* of D w.r.t. Σ is a maximal subset D' of D such that $D' \models \Sigma$. Let $\mathsf{rep}_\Sigma(D)$ be the set of repairs of D w.r.t Σ. Given a CQ Q over $\mathbf{S}$, we write $\mathsf{rep}_{\Sigma,Q}(D)$ for $\{D' \in \mathsf{rep}_\Sigma(D) \mid D' \models Q\}$. The following is a useful observation that we will exploit later in the paper:

Lemma 2. *Consider a database D, a set Σ of FDs, and a Boolean SJFCQ Q. We can compute in polynomial time a database D' such that $\sharp\mathsf{rep}_\Sigma(D) = \sharp\mathsf{rep}_{\Sigma,Q}(D')$.*

Problem Definition. For each set Σ of FDs and CQ Q, we focus on the problem $\sharp\mathsf{Repairs}(\Sigma, Q)$, which takes as input a database D, and asks for the number $\sharp\mathsf{rep}_{\Sigma,Q}(D)$. The goal is to classify it as tractable (place it in FP) or as intractable (show that is $\sharp$P-complete).

Another important task is whenever $\sharp\mathsf{Repairs}(\Sigma, Q)$ is intractable to classify it as approximable via a so-called *fully polynomial-time randomized approximation scheme* (FPRAS, for short), or as inapproximable. Formally, an FPRAS for $\sharp\mathsf{Repairs}(\Sigma, Q)$ is a randomized algorithm A that takes as input a database D, and numbers $\epsilon > 0$ and $0 < \delta < 1$, runs in polynomial time in $||D||$, $1/\epsilon$ and $\log(1/\delta)$, and produces a random variable $\mathsf{A}(D, \epsilon, \delta)$ such that $\mathsf{Pr}\left(|\mathsf{A}(D, \epsilon, \delta) - \sharp\mathsf{rep}_{\Sigma,Q}(D)| \leq \epsilon \cdot \sharp\mathsf{rep}_{\Sigma,Q}(D)\right) \geq 1 - \delta$.

LHS Chain FDs. Consider a set Σ of FDs over $\mathbf{S}$, and a relation name $R \in \mathbf{S}$. We say that Σ_R has a *left-hand side chain* (LHS chain) if the FDs of Σ_R can be arranged in a sequence $R : X_1 \rightarrow Y_1, \ldots, R : X_n \rightarrow Y_n$ such that $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_n$ [3]. We call such a sequence an *LHS chain of* Σ_R, and say that Σ has an LHS chain if, for every $R \in \mathbf{S}$, Σ_R has an LHS chain. We know that for sets of FDs with an LHS chain, counting the number of repairs is tractable:

Proposition 3 ([3]). *Given a database D, and a set Σ of FDs with an LHS chain (up to equivalence), $\sharp\mathsf{rep}_\Sigma(D)$ is computable in polynomial time in $||D||$.*

3. Exact Counting

The goal of this section is to discuss the key ideas behind our main result on exact counting:

Theorem 4. *For a set Σ of FDs, and an SJFCQ Q, $\sharp\mathsf{Repairs}(\Sigma, Q)$ is either in* FP *or* $\sharp$P*-complete, and we can decide in polynomial time in $||\Sigma|| + ||Q||$ which of the two cases hold.*

We observe that for every set Σ of FDs, and an SJFCQ Q, $\sharp\mathsf{Repairs}(\Sigma, Q)$ is in $\sharp$P: simply guess (in polynomial time) a subset D' of the given database D, and verify (again in polynomial time) that D' is a repair of D w.r.t. Σ that entails Q. We also note that the non-existence of an LHS chain (up to equivalence) is a preliminary boundary for the hard side of the target dichotomy, regardless of the CQ. We know from [3] that, for a set Σ of FDs, the problem $\sharp\mathsf{Repairs}(\Sigma)$, which given a database D, asks for $\sharp\mathsf{rep}_\Sigma(D)$, is $\sharp$P-hard whenever Σ does not have an LHS chain (up to equivalence). From the above discussion, and Lemma 2, we get the following result:

Proposition 5. *Consider a set Σ of FDs without an LHS chain (up to equivalence), and an SJFCQ Q. $\sharp\mathsf{Repairs}(\Sigma, Q)$ is* $\sharp$P*-complete.*

From Proposition 5, and the fact that checking whether a set Σ of FDs has an LHS chain (up to equivalence) is feasible in polynomial time [3], to obtain Theorem 4, it suffices to provide an analogous result for FDs with an LHS chain (up to equivalence). To this end, following what was done for primary keys in [2], we are going to concentrate on the problem of computing the relative frequency of the query. The *relative frequency* of a CQ Q w.r.t. a database D and a set Σ of FDs is defined as the ratio $\mathsf{rfreq}_{D,\Sigma}(Q) = \frac{\sharp\mathsf{rep}_{\Sigma,Q}(D)}{\sharp\mathsf{rep}_\Sigma(D)}$ that computes the percentage of repairs that entail the query Q. We then define the problem $\mathsf{RelFreq}(\Sigma, Q)$ that takes as input a database D, and asks for $\mathsf{rfreq}_{D,\Sigma}(Q)$. We can indeed focus on $\mathsf{RelFreq}(\Sigma, Q)$ since we know from Proposition 3 that $\sharp\mathsf{Repairs}(\Sigma)$ is in FP whenever Σ has an LHS chain (up to equivalence), which implies that $\sharp\mathsf{Repairs}(\Sigma, Q)$ and $\mathsf{RelFreq}(\Sigma, Q)$ have the same complexity, that is, both are either in FP or $\sharp$P-hard. Consequently, to obtain Theorem 4 it suffices to show the following:

Theorem 6. *For a set Σ of FDs with an LHS chain (up to equivalence), and an SJFCQ Q, $\mathsf{RelFreq}(\Sigma, Q)$ is either in* FP *or* $\sharp$P*-hard, and we can decide in polynomial time in $||\Sigma|| + ||Q||$ which of the two cases hold.*

In the rest, we give some hints on the proof of Theorem 6; we first need some new notions.

Canonical Covers. If Σ has an LHS chain (up to equivalence), we know that it has a single canonical cover[1] Σ', and Σ' has an LHS chain [3]. Furthermore, it can be verified that, for every

[1] That is, (i) the FDs in Σ are not redundant and have no redundant attributes, and (ii) for a relation name R and $X \subseteq \mathsf{att}(R)$, there is at most one FD in Σ of the form $R : X \rightarrow Y$.

relation name R occurring in Σ', there exists a unique sequence $R : X_1 \rightarrow Y_1, \ldots, R : X_n \rightarrow Y_n$ of the FDs of Σ'_R, which we call *the LHS chain* of Σ'_R, such that (i) $X_i \subsetneq X_{i+1}$ for each $i \in [n-1]$, (ii) $X_i \cap Y_j = \emptyset$ for each $i, j \in [n]$, and (iii) $Y_i \cap Y_j = \emptyset$ for each $i, j \in [n]$ with $i \neq j$. Since a canonical cover of a set of FDs can be computed in polynomial time, to establish Theorem 6 it suffices to show it for sets of FDs with an LHS chain that are canonical. Therefore, in the rest of the section, we assume that sets of FDs are canonical.

Primary FDs and Positions. Consider now an SJFCQ Q. Let α_R be the R-atom in Q, and let $\Lambda_R = R : X_1 \rightarrow Y_1, \ldots, R : X_n \rightarrow Y_n$ be the LHS chain of Σ_R. We call an FD $R : X_i \rightarrow Y_i$, for some $i \in [n]$, the *primary FD of* Σ_R *w.r.t.* Q if $X_i \cup Y_i$ contains an attribute A such that at position (R, A) in α_R we have a variable, whereas at each position of $\{(R, B) \mid B \in X_j \cup Y_j \text{ for } j < i\}$ in α_R we have a constant. Note that there is no guarantee that the primary FD of Σ_R w.r.t. Q exists. If the primary FD $R : X_i \rightarrow Y_i$ of Σ_R w.r.t. Q exists, for some $i \in [n]$, then the *primary-lhs positions of* α_R *(w.r.t.* Σ*)* are the positions $\{(R, A) \mid A \in X_i\}$, while the *non-primary-lhs positions of* α_R *(w.r.t.* Σ*)* are the positions $\{(R, B) \mid B \in \mathsf{att}(R) \setminus X_i\}$. We further call the sequence of FDs $R : X_1 \rightarrow Y_1, \ldots, R : X_{i-1} \rightarrow Y_{i-1}$ the *primary prefix of* Σ_R *w.r.t.* Q. If the primary FD of Σ_R w.r.t. Q does not exist, then, by convention, the primary-lhs positions of α_R are the positions $\{(R, A) \mid A \in \mathsf{att}(R)\}$, i.e., all the positions in α_R are primary-lhs, which in turn implies that α_R has no non-primary-lhs positions. Moreover, the primary prefix of Σ_R w.r.t. Q is defined as the sequence Λ_R itself. We denote by $\mathsf{pvar}_\Sigma(\alpha_R)$ the set of variables occurring in α_R at primary-lhs positions of α_R.

Query Complex Part. A variable $x \in \mathsf{var}(Q)$ is a *liaison variable (of* Q*)* if it occurs more than once in Q. The *complex part of* Q *w.r.t.* Σ, denoted $\mathsf{comp}_\Sigma(Q)$, is the set of atoms α of Q in which we have a constant or a liaison variable at a non-primary-lhs position (R, A), with α being the R-atom of Q, such that, for every FD $R : X \rightarrow Y$ in the primary prefix of Σ_R w.r.t. Q, $A \notin X \cup Y$. We observe that when $\mathsf{comp}_\Sigma(Q) = \emptyset$, the number of repairs of D w.r.t. Σ that entail Q coincides with the number of repairs (without any query) of the subset $D^{\Sigma,Q}_{\mathsf{conf}}$ of D obtained after removing the facts of D that are somehow in a conflict with Q, and thus, they cannot appear in a repair that entails Q. Since $D^{\Sigma,Q}_{\mathsf{conf}}$ can be computed in polynomial time in $||D||$, by Proposition 3 we obtain the following:

Proposition 7. *Given a database D, a set Σ of FDs with an LHS chain, and an SJFCQ Q with* $\mathsf{comp}_\Sigma(Q) = \emptyset$, *it holds that* $\mathsf{rfreq}_{D,\Sigma}(Q)$ *is computable in polynomial time in* $||D||$.

3.1. The Tractable Side

We start by defining some properties that, when satisfied by a SJFCQ, allow to restate its relative frequency in terms of the relative frequency of simpler queries.

We write $Q_1 \uplus Q_2$ for the CQ consisting of the atoms of two non-empty CQs Q_1, Q_2 that share no atoms and variables, i.e., $Q_1 \cap Q_2 = \emptyset$ and $\mathsf{var}(Q_1) \cap \mathsf{var}(Q_2) = \emptyset$. We also write $Q_{x \mapsto c}$, where $x \in \mathsf{var}(Q)$ and $c \in \mathbf{C}$, for the CQ obtained from Q after replacing x with c.

Theorem 8. *Consider a set Σ of FDs with an LHS chain, and a SJFCQ Q. For every database D, the following hold:*

1. *If* $Q = Q_1 \uplus Q_2$, *then* $\mathsf{rfreq}_{D,\Sigma}(Q) = \mathsf{rfreq}_{D,\Sigma}(Q_1) \times \mathsf{rfreq}_{D,\Sigma}(Q_2)$.

2. *If* $\mathsf{comp}_\Sigma(Q) \neq \emptyset$*, and there exists a variable* $x \in \mathsf{var}(Q)$ *such that* $x \in \mathsf{pvar}_\Sigma(\alpha)$ *for every* $\alpha \in \mathsf{comp}_\Sigma(Q)$*, then there is* $D^* \subseteq D$*, computable in polynomial time in* $||D||$*, such that*

$$\mathsf{rfreq}_{D,\Sigma}(Q) \quad = \quad \left(1 - \prod_{c \in \mathsf{dom}(D)} \left(1 - \mathsf{rfreq}_{D^*,\Sigma}(Q_{x\mapsto c})\right)\right) \times \frac{\sharp\mathsf{rep}_\Sigma\left(D \setminus D^{\Sigma,Q}_{\mathsf{conf}}\right)}{\sharp\mathsf{rep}_\Sigma(D)}.$$

3. *If* $\mathsf{comp}_\Sigma(Q) \neq \emptyset$*, and there exists an atom* $\alpha = R(\bar{t}) \in \mathsf{comp}_\Sigma(Q)$ *such that* $\mathsf{pvar}_\Sigma(\alpha) = \emptyset$*, and a variable* x *occurs in* α *at a position of* $\{(R,A) \mid A \in Y\}$*, where* $R : X \rightarrow Y$ *is the primary FD of* Σ_R *w.r.t.* Q*, then* $\mathsf{rfreq}_{D,\Sigma}(Q) = \sum_{c \in \mathsf{dom}(D)} \mathsf{rfreq}_{D,\Sigma}(Q_{x\mapsto c})$.

Item (1) of Theorem 8 is straightforward, and implies that when $Q = Q_1 \uplus Q_2$, one can compute the relative frequency of Q in polynomial time, if the same can be done for the relative frequencies of Q_1 and Q_2. Items (2) and (3) are much more involved, and we refer the reader to the full version of the paper [5]. The main message is that when a SJFCQ Q satisfies the conditions of item (2), then its relative frequency is polynomial-time computable if so is for the relative frequency of $Q_{x\mapsto c}$, for each constant c of the database, over a certain easily computable subset D^* of D, where x is as defined in item (2). Similarly, when Q satisfies the conditions of item (3), then its relative frequency is polynomial-time computable, if so is for the relative frequency of $Q_{x\mapsto c}$, for each constant c of the database, where x is as defined in item (3).

For a set Σ of FDs with an LHS chain and an SJFCQ Q, we say that Q is Σ-*safe* if either its complex part is empty, i.e., $\mathsf{comp}_\Sigma(Q) = \emptyset$, or after recursively applying the conditions of items (1)-(3) of Theorem 8 to Q, we are only left with queries having an empty complex part.[2] By definition of Σ-safe queries, and by Proposition 7 and Theorem 8, we obtain the following:

Theorem 9. *Consider a set* Σ *of FDs with an LHS chain, and an SJFCQ* Q*. If* Q *is* Σ*-safe, then* $\mathsf{RelFreq}(\Sigma, Q)$ *is in* FP.

Interestingly, safety exhausts the tractable side of the dichotomy stated in Theorem 6. That is, if Q is not Σ-safe, then $\mathsf{RelFreq}(\Sigma, Q)$ is $\sharp$P-hard, as we discuss in the next section. Let us stress that checking whether Q is Σ-safe is feasible in polynomial time in $||\Sigma|| + ||Q||$, which establishes the second part of Theorem 6.

3.2. The Hard Side

We conclude this section by briefly discussing how the intractable side of the dichotomy stated in Theorem 6 is obtained. We know from [2] that, for every set Σ of primary keys, and an SJFCQ Q that is not Σ-safe, $\mathsf{RelFreq}(\Sigma, Q)$ is $\sharp$P-hard. Hence, with Theorem 9 in place, stating that query safety ensures tractability, we can obtain the hard side of the dichotomy by showing:

Theorem 10. *There is a set* Σ' *of primary keys, and a non* Σ'*-safe SJFCQ* Q' *such that, for each set* Σ *of FDs with an LHS chain and non* Σ*-safe SJFCQ* Q*,* $\mathsf{RelFreq}(\Sigma', Q')$ *reduces to* $\mathsf{RelFreq}(\Sigma, Q)$.

[2] We remark that for items (2) and (3), it is enough to recursively apply the conditions of Theorem 8 to $Q_{x\mapsto c}$ for only *one* arbitrarily chosen constant from $\mathbf{C}$, and thus the notion of Σ-safety is database independent.

Theorem 10 is shown in two main steps. We first prove that there exists a set Σ^* of FDs, and an SJFCQ Q^* such that $\mathsf{RelFreq}(\Sigma^*, Q^*)$ can be reduced to $\mathsf{RelFreq}(\Sigma, Q)$ by means of a set of rewriting rules that extend the ones of [2]. Second, we show that there is a set Σ' of primary keys, and an SJFCQ Q' that is not Σ'-safe such that, for every database D, $\mathsf{rfreq}_{D,\Sigma^*}(Q^*) = \mathsf{rfreq}_{D,\Sigma'}(Q')$, which then implies that $\mathsf{RelFreq}(\Sigma', Q')$ reduces to $\mathsf{RelFreq}(\Sigma^*, Q^*)$, as needed.

4. Approximate Counting

We now briefly discuss our results on approximate counting. The main contribution of this section is the following theorem; BPP is the class of decision problems that are efficiently solvable via a randomized algorithm with a bounded two-sided error [6].

Theorem 11. *For every set Σ of FDs with an LHS chain (up to equivalence), and a CQ Q, $\sharp\mathsf{Repairs}(\Sigma, Q)$ admits an FPRAS. Moreover, for $\Sigma^h = \{R : A_1 \rightarrow A_2, R : A_3 \rightarrow A_4\}$, where $\mathsf{att}(R) = \{A_i \mid i \in [4]\}$, $\sharp\mathsf{Repairs}(\Sigma^h, Q)$ admits no FPRAS, for every SJFCQ Q, unless* NP $\subseteq$ BPP.

To prove the approximability result we observe that $\sharp\mathsf{Repairs}(\Sigma, Q)$ can be seen as an instantiation of the so-called *union of sets* problem [7], which given a succinct representation of $n \geq 1$ sets $S_1, \ldots, S_n$, asks for the number $|\bigcup_{i\in[n]} S_i|$. Indeed, given a database D, let $\mathsf{hom}_{D,\Sigma}(Q)$ be the set of all homomorphic images of Q in D that are consistent w.r.t. Σ. Formally, $\mathsf{hom}_{D,\Sigma}(Q) = \{h(Q) \mid h \text{ is a homomorphism from } Q \text{ to } D \text{ such that } h(Q) \models \Sigma\}$. Hence, $\sharp\mathsf{rep}_{\Sigma,Q}(D) = \left|\bigcup_{i\in[n]} \mathsf{rep}_{\Sigma,H_i}(D)\right|$, assuming that $\mathsf{hom}_{D,\Sigma}(Q) = \{H_1, \ldots, H_n\}$.[3]

From the above discussion, and the results of [7] on the approximability of the union of sets problem, showing the existence of an FPRAS for $\sharp\mathsf{Repairs}(\Sigma, Q)$ boils down to showing that for each $H \in \mathsf{hom}_{D,\Sigma}(Q)$ we can (i) compute $\sharp\mathsf{rep}_{\Sigma,H}(D)$, (ii) sample elements of $\mathsf{rep}_{\Sigma,H}(D)$ uniformly at random, and (iii) check whether a database is in $\mathsf{rep}_{\Sigma,H}(D)$, all in polynomial time in $||D||$. We prove that the above properties hold when Σ has an LHS chain (up to equivalence), and the first part of Theorem 11 follows.

The proof of the inapproximability of $\sharp\mathsf{Repairs}(\Sigma^h, Q)$, for every SJFCQ Q, employs a so-called gap preserving reduction from the *promise* problem $\mathsf{Gap3SAT}_\gamma$, for some fixed $\gamma \in (0, \frac{1}{8})$, i.e., the problem that given a Boolean formula φ in 3CNF, asks whether φ is satisfiable. The promise is that if φ is unsatisfiable, then every truth assignment makes at most $\frac{7}{8} + \gamma$ of the clauses of φ true. For every $\gamma \in (0, \frac{1}{8})$, $\mathsf{Gap3SAT}_\gamma$ is NP-complete [8].

For a fixed $\gamma \in (0, \frac{1}{8})$, the reduction maps each formula φ to a database $\mathsf{db}(\varphi)$ such that the following holds: for any *yes* instance φ_Y of $\mathsf{Gap3SAT}_\gamma$, and any *no* instance φ_N of $\mathsf{Gap3SAT}_\gamma$, the "gap" (i.e., the ratio) between the numbers $\sharp\mathsf{rep}_{\Sigma^h,Q}(\mathsf{db}(\varphi_Y))$ and $\sharp\mathsf{rep}_{\Sigma^h,Q}(\mathsf{db}(\varphi_N))$ is so large that an FPRAS for $\sharp\mathsf{Repairs}(\Sigma^h, Q)$ could be used, by employing a small enough error ϵ, to distinguish between *yes* and *no* instances of $\mathsf{Gap3SAT}_\gamma$ with high probability, and thus, placing the NP-complete problem $\mathsf{Gap3SAT}_\gamma$ in BPP.

The Difficulty Underlying a Dichotomy. Despite our efforts, we could not get a complete approximability/inapproximability classification. However, in the full version of the paper [5]

[3] By abuse of terminology, we treat each H_i as a CQ consisting of facts.

we show that obtaining such a classification is as hard as solving the challenging open question whether $\sharp$MaxMatch admits an FPRAS, i.e., the problem that given a bipartite graph G, asks for the number of maximal matchings in G; please refer to the full version for more details.

5. Conclusion

The obvious problems that remain open are the following: (1) lift the dichotomy of Theorem 4 for FDs and self-join-free CQs to arbitrary CQs with self-joins, and (2) establish an approximability/inapproximability dichotomy for FDs and SJFCQs.

We point out that approximate versions of CQA have already been considered, both in the database and ontological setting. In the database setting, to the best of our knowledge [4, 9, 10] are the only works that focus on approximations in CQA. In particular, [4] studies approximation algorithms for the relative frequency *under primary keys only*, which have also been experimentally evaluated in [9]. In [10], FDs are considered, but the notion of repair is based on value updates, rather than tuple deletions, as we do. In the ontological setting, different approximations have been considered. However, in this setting, approximations are understood as *subsets* (i.e., sound but not complete) of the *standard consistent answers* (e.g., see [11, 12]).

References

[1] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: PODS, 1999, pp. 68–79.

[2] D. Maslowski, J. Wijsen, A dichotomy in the complexity of counting database repairs, J. Comput. Syst. Sci. 79 (2013) 958–983.

[3] E. Livshits, B. Kimelfeld, J. Wijsen, Counting subset repairs with functional dependencies, J. Comput. Syst. Sci. 117 (2021) 154–164.

[4] M. Calautti, M. Console, A. Pieris, Counting database repairs under primary keys revisited, in: PODS, 2019, pp. 104–118.

[5] M. Calautti, E. Livshits, A. Pieris, M. Schneider, Counting database repairs entailing a query: The case of functional dependencies, in: PODS (to appear), 2022, pp. 91–103.

[6] S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press, 2009.

[7] R. M. Karp, M. Luby, N. Madras, Monte-carlo approximation algorithms for enumeration problems, J. Algorithms 10 (1989) 429–448.

[8] J. Håstad, Some optimal inapproximability results, J. ACM 48 (2001) 798–859.

[9] M. Calautti, M. Console, A. Pieris, Benchmarking approximate consistent query answering, in: PODS, 2021, pp. 233–246.

[10] S. Greco, C. Molinaro, Probabilistic query answering over inconsistent databases, Ann. Math. Artif. Intell. 64 (2012).

[11] S. Greco, C. Molinaro, I. Trubitsyna, Computing approximate query answers over inconsistent knowledge bases, in: IJCAI, 2018, pp. 1838–1846.

[12] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in datalog+/-, in: IJCAI, 2018, pp. 1921–1927.