

Uniform Operational Consistent Query Answering

Marco Calautti
University of Trento
marco.calautti@unitn.it

Andreas Pieris
University of Edinburgh &
University of Cyprus
apieris@inf.ed.ac.uk

Ester Livshits
University of Edinburgh
ester.livshits@ed.ac.uk

Markus Schneider
University of Edinburgh
m.schneider@ed.ac.uk

ABSTRACT

Operational consistent query answering (CQA) is a recent framework for CQA, based on revised definitions of repairs and consistent answers, which opens up the possibility of efficient approximations with explicit error guarantees. The main idea is to iteratively apply operations (e.g., fact deletions), starting from an inconsistent database, until we reach a database that is consistent w.r.t. the given set of constraints. This gives us the flexibility of choosing the probability with which we apply an operation, which in turn allows us to calculate the probability of an operational repair, and thus, the probability with which a consistent answer is entailed. A natural way of assigning probabilities to operations is by targeting the uniform probability distribution over a reasonable space such as the set of operational repairs, the set of sequences of operations that lead to an operational repair, and the set of available operations at a certain step of the repairing process. This leads to what we generally call uniform operational CQA. The goal of this work is to perform a data complexity analysis of both exact and approximate uniform operational CQA, focusing on functional dependencies (and subclasses thereof), and conjunctive queries. The main outcome of our analysis (among other positive and negative results), is that uniform operational CQA pushes the efficiency boundaries further by ensuring the existence of efficient approximation schemes in scenarios that go beyond the simple case of primary keys, which seems to be the limit of the classical approach to CQA.

ACM Reference Format:

Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. 2022. Uniform Operational Consistent Query Answering. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 33 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Consistent query answering (CQA) is an elegant framework, introduced in the late 1990s by Arenas, Bertossi, and Chomicki [1], that allows us to compute conceptually meaningful answers to queries posed over inconsistent databases, that is, databases that do not

conform to their specifications. 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 *consistent answers*, that is, answers that are entailed by every repair. Since deciding whether a candidate answer is a consistent answer is most commonly intractable in data complexity (in fact, even for primary keys and conjunctive queries, the problem is coNP-hard [7]), there was a great effort on drawing the tractability boundary for CQA; see, e.g., [11–13, 16–18]. Much of this effort led to interesting dichotomy results that precisely characterize when CQA is tractable/intractable in data complexity. However, the tractable fragments do not cover many relevant scenarios that go beyond primary keys.

As extensively argued in [5], the goal of a practically applicable CQA approach should be efficient approximate query answering with explicit error guarantees rather than exact query answering. In the realm of the CQA approach described above, one could try to devise efficient probabilistic algorithms with bounded one- or two-sided error. However, it is unlikely that such algorithms exist since, even for very simple scenarios (e.g., primary keys and conjunctive queries), placing the problem in tractable randomized complexity classes such as RP or BPP would imply that the polynomial hierarchy collapses [15]. Another promising idea is to replace the rather strict notion of consistent answers with the more refined notion of relative frequency, that is, the percentage of repairs that entail an answer, and then try to approximate it via a fully polynomial-time randomized approximation scheme (FPRAS); computing it exactly is, unsurprisingly, $\#P$ -hard [19]. Indeed, for primary keys and conjunctive queries, one can approximate the relative frequency via an FPRAS; this is implicit in [9], and it has been made explicit in [3]. Moreover, a recent experimental evaluation revealed that approximate CQA in the presence of primary keys and conjunctive queries is not unrealistic in practice [4]. However, it seems that the simple case of primary keys is the limit of this approach. We have strong indications that in the case of FDs the problem of computing the relative frequency does not admit an FPRAS, while in the case of keys it is a highly non-trivial problem [6].

The above limitations of the classical CQA approach led the authors of [5] to propose a new framework for CQA, based on revised definitions of repairs and consistent answers, which opens up the possibility of efficient approximations with error guarantees. The main idea underlying this new framework is to replace the declarative approach to repairs with an *operational* one that explains the process of constructing a repair. In other words, we can iteratively

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

apply operations (e.g., fact deletions), starting from an inconsistent database, until we reach a database that is consistent w.r.t. the given set of constraints. This gives us the flexibility of choosing the probability with which we apply an operation, which in turn allows us to calculate the probability of an operational repair, and thus, the probability with which an answer is entailed.

Probabilities can be naturally assigned to operations in many scenarios leading to inconsistencies. This is illustrated by the following example from [5]. Consider a data integration scenario that results in a database containing the facts $\text{Emp}(1, \text{Alice})$ and $\text{Emp}(1, \text{Tom})$ that violate the constraint that the first attribute of the relation name Emp (the id) is a key. Suppose we have a level of trust in each of the sources; say we believe that each is 50% reliable. With probability $0.5 \cdot 0.5 = 0.25$ we do not trust either tuple and apply the operation that removes both facts. With probability $(1 - 0.25)/2 = 0.375$ we remove either $\text{Emp}(1, \text{Alice})$ or $\text{Emp}(1, \text{Tom})$.

The preliminary data complexity analysis of operational CQA performed in [5] revealed that computing the probability of a candidate answer is $\#P$ -hard and inapproximable, even for primary keys and conjunctive queries. However, these negative results should not be seen as the end of the story, but rather as the beginning since operational CQA gives us the flexibility to choose the probabilities assigned to operations. Indeed, the main question left open by [5] is the following: how can we choose the probabilities assigned to operations so that the existence of an FPRAS is guaranteed?

A natural way of choosing those probabilities is to follow the uniform probability distribution over a reasonable space. The obvious candidates for such a space are (i) the set of operational repairs, (ii) the set of sequences of operations that lead to a repair (note that multiple such sequences can lead to the same repair), and (iii) the set of available operations at a certain step of the repairing process. This leads to the so-called *uniform operational CQA*. The obvious question is how the complexity of exact and approximate operational CQA is affected if we assign probabilities to operations according to the above refined ways. In particular, we would like to understand whether uniform operational CQA allows us to go beyond the relatively simple case of primary keys.

Our goal is to perform a complexity analysis of uniform operational CQA, and provide answers to the above central questions. Our main findings can be summarized as follows:

- (1) Exact uniform operational CQA remains $\#P$ -hard, even in the case of primary keys and conjunctive queries.
- (2) Uniform operational CQA admits an FPRAS if we focus on primary keys and conjunctive queries.
- (3) In the case of arbitrary keys and FDs, although assigning probabilities to operations based on uniform repairs and sequences (approaches (i) and (ii) discussed above) does not lead (or it remains open whether it leads) to the approximability of operational CQA, the approach of uniform operations renders the problem approximable. The latter is a significant result since it goes beyond the simple case of primary keys.

2 PRELIMINARIES

We recall the basics on relational databases, functional dependencies, and conjunctive queries. In the rest of the paper, we assume the

disjoint countably infinite sets \mathbf{C} and \mathbf{V} of *constants* and *variables*, respectively. For $n > 0$, let $[n]$ be the set $\{1, \dots, n\}$.

Relational Databases. A (*relational*) *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 $\text{att}(R)$ for the set $\{A_1, \dots, A_n\}$ of attributes. A *fact* over \mathbf{S} is an expression of the form $R(c_1, \dots, c_n)$, where $R/n \in \mathbf{S}$, and $c_i \in \mathbf{C}$ for each $i \in [n]$. A *database* D over \mathbf{S} is a finite set of facts over \mathbf{S} . The *active domain* of D , denoted $\text{dom}(D)$, is the set of constants occurring in D . For a fact $f = R(c_1, \dots, c_n)$, with (A_1, \dots, A_n) being the tuple of attribute names of R , we write $f[A_i]$ for the constant c_i .

Functional Dependencies. A *functional dependency* (FD) ϕ over a schema \mathbf{S} is an expression of the form $R : X \rightarrow Y$, where $R/n \in \mathbf{S}$ and $X, Y \subseteq \text{att}(R)$. When X or Y are singletons, we avoid the curly brackets, and simply write the attribute name. We call ϕ a *key* if $X \cup Y = \text{att}(R)$. Given a set Σ of keys over \mathbf{S} , we say that Σ is a set of *primary keys* if, for each $R \in \mathbf{S}$, there exists at most one key in Σ of the form $R : X \rightarrow Y$. A database D satisfies an FD $\phi = R : X \rightarrow Y$, denoted $D \models \phi$, if, for every two facts $R(\bar{c}_1), R(\bar{c}_2) \in D$ the following holds: $R(\bar{c}_1)[A] = R(\bar{c}_2)[A]$ for every $A \in X$ implies $R(\bar{c}_1)[B] = R(\bar{c}_2)[B]$ 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, we say that D is *inconsistent* w.r.t. Σ .

Conjunctive Queries. A (*relational*) *atom* α over a schema \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 *conjunctive query* (CQ) Q over \mathbf{S} is an expression of the form $\text{Ans}(\bar{x}) :- R_1(\bar{y}_1), \dots, R_n(\bar{y}_n)$, where $R_i(\bar{y}_i)$, for $i \in [n]$, is an atom over \mathbf{S} , \bar{x} are the *answer variables* of Q , and each variable in \bar{x} is mentioned in \bar{y}_i for some $i \in [n]$. We may write $Q(\bar{x})$ to indicate that \bar{x} are the answer variables of Q . When \bar{x} is empty, Q is called *Boolean*. The semantics of CQs is given via homomorphisms. Let $\text{var}(Q)$ and $\text{const}(Q)$ be the set of variables and constants in Q , respectively. A *homomorphism* from a CQ Q of the form $\text{Ans}(\bar{x}) :- R_1(\bar{y}_1), \dots, R_n(\bar{y}_n)$ to a database D is a function $h : \text{var}(Q) \cup \text{const}(Q) \rightarrow \text{dom}(D)$, which is the identity over \mathbf{C} , such that $R_i(h(\bar{y}_i)) \in D$ for each $i \in [n]$. A tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$ is an *answer to Q over D* if there is a homomorphism h from Q to D with $h(\bar{x}) = \bar{c}$. Let $Q(D)$ be the answers to Q over D . For Boolean CQs, we write $D \models Q$, and say that D *entails* Q , if $(\bar{c}) \in Q(D)$.

3 OPERATIONAL CQA

We proceed to recall the recent operational approach to consistent query answering, introduced in [5]. Although this new framework can deal with arbitrary integrity constraints (i.e., tuple-generating dependencies, equality-generating dependencies, and denial constraints), for our purposes we need its simplified version that only deals with functional dependencies.

Operations and Violations. The notion of operation is the building block of the operational approach. In the original framework, the operations are standard updates $+F$ that add a set F of facts to the database, or $-F$ that remove F from the database. However, since in this work we deal with FDs, we only need to remove facts because the addition of a fact would never resolve a conflict. The

formal definition of the notion of operation follows. As usual, we write $\mathcal{P}(S)$ for the powerset of a set S .

Definition 3.1. (Operation) For a database D over a schema S , a D -operation is a function $op : \mathcal{P}(D) \rightarrow \mathcal{P}(D)$ such that, for some non-empty set $F \subseteq D$ of facts, for every $D' \in \mathcal{P}(D)$, $op(D') = D' \setminus F$. We write $-F$ to refer to this operation. ■

The operations $-F$ depend on the database D as they are defined over D . Since D will be clear from the context, we may refer to them simply as operations, omitting D . Also, when F contains a single fact f , we write $-f$ instead of the more formal $-\{f\}$. The main idea of the operational approach to CQA is to iteratively apply operations, starting from an inconsistent database D , until we reach a database $D' \subseteq D$ that is consistent w.r.t. the given set Σ of FDs. However, as discussed in [5], we need to ensure that at each step of this repairing process, at least one violation is resolved. To this end, we need to keep track of all the reasons that cause the inconsistency of D w.r.t. Σ . This brings us to the notion of FD violation.

Definition 3.2. (FD Violation) For a database D over a schema S , a D -violation of an FD $\phi = R : X \rightarrow Y$ over S is a set $\{f, g\} \subseteq D$ of facts such that $\{f, g\} \not\models \phi$. We denote the set of D -violations of ϕ by $V(D, \phi)$. Furthermore, for a set Σ of FDs, we denote by $V(D, \Sigma)$ the set $\{(\phi, v) \mid \phi \in \Sigma \text{ and } v \in V(D, \phi)\}$. ■

Thus, a pair $(\phi, \{f, g\}) \in V(D, \Sigma)$ means that one of the reasons why the database D is inconsistent w.r.t. Σ is because it violates ϕ due to the facts f and g . As discussed in [5], apart from forcing an operation to be fixing, i.e., to fix at least one violation, we also need to force an operation to remove a set of facts only if it contributes as a whole to a violation. Such operations are called justified.

Definition 3.3. (Justified Operation) Let D be a database over a schema S , and Σ a set of FDs over S . For a database $D' \subseteq D$, a D -operation $-F$ is called (D', Σ) -justified if there exists $(\phi, \{f, g\}) \in V(D', \Sigma)$ such that $F \subseteq \{f, g\}$. ■

Note that justified operations do not try to minimize the number of atoms that need to be removed. As argued in [5], a set of facts that collectively contributes to a violation should be considered as a justified operation during the iterative repairing process since we do not know a priori which atoms should be deleted, and therefore, we need to explore all the possible scenarios.

Repairing Sequences. As said above, the main idea of the operational approach is to iteratively apply justified operations. This is formalized via the notion of repairing sequence. Consider a database D and a set Σ of FDs. Given a sequence $s = (op_i)_{1 \leq i \leq n}$ of D -operations, we define $D_0^s = D$ and $D_i^s = op_i(D_{i-1}^s)$ for $i \in [n]$. In other words, D_i^s is obtained by applying to D the first i operations of s . The notion of repairing sequence follows:

Definition 3.4. (Repairing Sequence) Consider a database D and a set Σ of FDs. A sequence of D -operations $s = (op_i)_{1 \leq i \leq n}$ is called (D, Σ) -repairing if, for every $i \in [n]$, op_i is (D_{i-1}^s, Σ) -justified. Let $RS(D, \Sigma)$ be the set of all (D, Σ) -repairing sequences. ■

It is easy to verify that the length of a (D, Σ) -repairing sequence is linear in the size of D . It is also clear that the set $RS(D, \Sigma)$ is finite. For a (D, Σ) -repairing sequence $s = (op_i)_{1 \leq i \leq n}$, we define its *result* as the database $s(D) = D_n^s$, and call it *complete* if $s(D) \models \Sigma$,

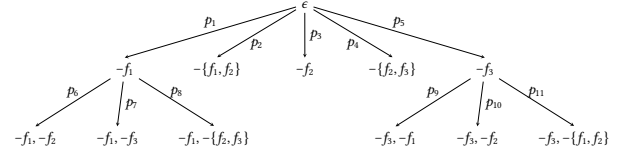


Figure 1: Repairing Markov Chain

i.e., it leads to a consistent database. Let $CRS(D, \Sigma)$ be the set of all complete (D, Σ) -repairing sequences.

Operational Repairs. A candidate (operational) repair of a database D w.r.t. a set Σ of FDs is a database D' such that $D' = s(D)$ for some $s \in CRS(D, \Sigma)$. Let $CORep(D, \Sigma)$ be the set of all candidate repairs of D w.r.t. Σ . Although every database of $CORep(D, \Sigma)$ corresponds to a conceptually meaningful way of repairing the database D , we would like to have a mechanism that allows us to choose which candidate repairs should be considered for query answering purposes, and assign likelihoods to those repairs.

The fact that we can operationally repair an inconsistent database via repairing sequences gives us the flexibility of choosing which operations (that is, fact deletions) are more likely than others, which in turn allows us to talk about the probability of a repair, and thus, the probability with which an answer is entailed. The idea of assigning likelihoods to operations extending sequences can be described as follows: for all possible extensions $s \cdot op_1, \dots, s \cdot op_k$ of a repairing sequence s , we assign probabilities p_1, \dots, p_k to them so they add up to 1. This is done by exploiting a *tree-shaped Markov chain* that arranges its states (i.e., repairing sequences) in a rooted tree, where (i) the empty sequence of operations, which is by definition repairing, is the root, (ii) the children of each state are its possible extensions, and (iii) the set of states corresponding to complete sequences coincide with the set of leaves. We write ϵ for the empty sequence of operations. We further write $Ops_s(D, \Sigma)$ for the set of (D, Σ) -repairing sequences $\{s' \in RS(D, \Sigma) \mid s' = s \cdot op \text{ for some } D\text{-operation } op\}$.

Definition 3.5. (Repairing Markov Chain) For a database D and a set Σ of FDs, a (D, Σ) -repairing Markov chain is an edge-labeled rooted tree $T = (V, E, P)$, where $V = RS(D, \Sigma)$, $E \subseteq V \times V$, and $P : E \rightarrow [0, 1]$, such that:

- (1) the root is the empty sequence ϵ ,
- (2) for a non-leaf node $s \in V$, $\{s' \mid (s, s') \in E\} = Ops_s(D, \Sigma)$,
- (3) for a non-leaf node $s \in V$, $\sum_{t \in \{s' \mid (s, s') \in E\}} P(s, t) = 1$, and
- (4) $\{s \in V \mid s \text{ is a leaf}\} = CRS(D, \Sigma)$.

A *repairing Markov chain generator* w.r.t. Σ is a function M_Σ assigning to every database D a (D, Σ) -repairing Markov chain. ■

We give a simple example, which will serve as a running example, that illustrates the notion of repairing Markov chain:

Example 3.6. Consider the database $D = \{f_1, f_2, f_3\}$ over the schema $S = \{R/3\}$, where $f_1 = R(a_1, b_1, c_1)$, $f_2 = R(a_1, b_2, c_2)$ and $f_3 = R(a_2, b_1, c_2)$. Consider also the set $\Sigma = \{\phi_1, \phi_2\}$ of FDs over S , where $\phi_1 = R : A \rightarrow B$ and $\phi_2 = R : C \rightarrow B$, assuming that (A, B, C) is the tuple of attributes of R . It is easy to see that $D \not\models \Sigma$. In particular, we have that $V(D, \Sigma) = \{(\phi_1, \{f_1, f_2\}), (\phi_2, \{f_2, f_3\})\}$. It is easy to verify that for the edge-labeled rooted tree $T = (V, E, P)$ in

Figure 1, $V = \text{RS}(D, \Sigma)$, for a non-leaf node s the set of its children is $\text{Ops}_s(D, \Sigma)$, and the set of leaves coincides with $\text{CRS}(D, \Sigma)$. Hence, providing that $p_1 + p_2 + p_3 + p_4 + p_5 = 1$, $p_6 + p_7 + p_8 = 1$ and $p_9 + p_{10} + p_{11} = 1$, T is a (D, Σ) -repairing Markov chain. ■

The purpose of a repairing Markov chain generator is to provide a mechanism for defining a family of repairing Markov chains independently of the database. One can design a repairing Markov chain generator M_Σ once, and whenever the database D changes, the desired (D, Σ) -repairing Markov chain is simply $M_\Sigma(D)$.

We now recall the notion of operational repair: they are candidate operational repairs obtained via repairing sequences that are *reachable* leaves of a repairing Markov chain, i.e., leaves with non-zero probability. The probability of a leaf is coming from the so-called leaf distribution of a repairing Markov chain. Formally, given a database D and a set Σ of FDs, the *leaf distribution* of a (D, Σ) -repairing Markov chain $T = (V, E, \mathbf{P})$ is a function π that assigns to each leaf s of T a number from $[0, 1]$ as follows: assuming that $(s_0, s_1), (s_1, s_2), \dots, (s_{n-1}, s_n)$, where $n \geq 0$, $\varepsilon = s_0$ and $s = s_n$, is the unique path in T from ε to s , $\pi(s) = \mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \dots \cdot \mathbf{P}(s_{n-1}, s_n)$. The set of *reachable leaves* of T , denoted $\text{RL}(T)$, is the set of leaves of T that have non-zero probability according to the leaf distribution of T .

Definition 3.7. (Operational Repair) Given a database D , a set Σ of FDs, and a repairing Markov chain generator M_Σ w.r.t. Σ , an *(operational) repair* of D w.r.t. M_Σ is a database $D' \in \text{COPrep}(D, \Sigma)$ such that $D' = s(D)$ for some $s \in \text{RL}(M_\Sigma(D))$. Let $\text{ORep}(D, M_\Sigma)$ be the set of all operational repairs of D w.r.t. M_Σ . ■

An operational repair may be obtainable via multiple repairing sequences that are reachable leaves of the underlying repairing Markov chain. The probability of a repair D' is calculated by summing up the probabilities of all reachable leaves s so that $D' = s(D)$.

Definition 3.8. (Operational Semantics) Given a database D , a set Σ of FDs, and a repairing Markov chain generator M_Σ w.r.t. Σ , the probability of an operational repair D' of D w.r.t. M_Σ is

$$\mathbf{P}_{D, M_\Sigma}(D') = \sum_{s \in \text{RL}(M_\Sigma(D)) \text{ and } D'=s(D)} \pi(s),$$

where π is the leaf distribution of $M_\Sigma(D)$. The *operational semantics* of D w.r.t. M_Σ is defined as the set of repair-probability pairs $[[D]]_{M_\Sigma} = \{(D', \mathbf{P}_{D, M_\Sigma}(D')) \mid D' \in \text{ORep}(D, M_\Sigma)\}$. ■

Operational CQA. We now have in place all the necessary notions to recall the operational approach to consistent query answering, and define the main problem of interest. For a database D , a set Σ of FDs, a Markov chain generator M_Σ w.r.t. Σ , a query $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$, the probability of \bar{c} being an answer to Q over some operational repair of D is defined as

$$\mathbf{P}_{M_\Sigma, Q}(D, \bar{c}) = \sum_{(D', \rho) \in [[D]]_{M_\Sigma} \text{ and } \bar{c} \in Q(D')} \rho.$$

We can now talk about operational consistent answers. In particular, the set of *operational consistent answers* to Q over D according to M_Σ is defined as the set $\{(\bar{c}, \mathbf{P}_{M_\Sigma, Q}(D, \bar{c})) \mid \bar{c} \in \text{dom}(D)^{|\bar{x}|}\}$.

The problem of interest in this context, dubbed OCQA, accepts as input a database D , a set Σ of FDs, a repairing Markov chain generator M_Σ w.r.t. Σ , a query $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,

and asks for the probability $\mathbf{P}_{M_\Sigma, Q}(D, \bar{c})$. We are, in fact, interested in the *data complexity* of OCQA, i.e., for a set Σ of FDs, a repairing Markov chain generator M_Σ w.r.t. Σ , and a query $Q(\bar{x})$, we focus on

PROBLEM : OCQA($\Sigma, M_\Sigma, Q(\bar{x})$)
INPUT : A database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$.
OUTPUT : $\mathbf{P}_{M_\Sigma, Q}(D, \bar{c})$.

Until now, a repairing Markov chain generator is a general function. We proceed to discuss the novel idea of uniform operational CQA, which provides concrete ways of defining such a function.

4 UNIFORM OPERATIONAL CQA

A natural way of defining a repairing Markov chain is to assign probabilities to operations according to the uniform probability distribution over a reasonable space. The obvious options for such a space are (i) the set of candidate operational repairs, (ii) the set of complete repairing sequences, and (iii) the set of available operations at a certain step of the repairing process. More precisely, given a set Σ of FDs, it is natural to consider the repairing Markov chain generators M_Σ^{ur} (uniform repairs), M_Σ^{us} (uniform sequences), and M_Σ^{uo} (uniform operations) w.r.t. Σ such that, for a database D :

- (i) $\text{ORep}(D, M_\Sigma^{\text{ur}}) = \text{COPrep}(D, \Sigma)$, and for $D' \in \text{ORep}(D, M_\Sigma^{\text{ur}})$, $\mathbf{P}_{D, M_\Sigma^{\text{ur}}}(D') = \frac{1}{|\text{ORep}(D, M_\Sigma^{\text{ur}})|}$.
- (ii) For every $s \in \text{CRS}(D, \Sigma)$, assuming that π is the leaf distribution of $M_\Sigma^{\text{us}}(D)$, $\pi(s) = \frac{1}{|\text{CRS}(D, \Sigma)|}$.
- (iii) For every $s, s' \in \text{RS}(D, \Sigma)$, assuming that $M_\Sigma^{\text{uo}}(D) = (V, E, \mathbf{P})$, $s' \in \text{Ops}_s(D, \Sigma)$ implies $\mathbf{P}(s, s') = \frac{1}{|\text{Ops}_s(D, \Sigma)|}$.

We now explain, by means of an example, how these Markov chain generators are defined; the formal definitions are in Appendix A.

In the rest of the section, let D and Σ be the database and the set of FDs, respectively, from Example 3.6. Recall that any (D, Σ) -repairing Markov chain looks as the one depicted in Figure 1 with $p_1 + p_2 + p_3 + p_4 + p_5 = 1$, $p_6 + p_7 + p_8 = 1$ and $p_9 + p_{10} + p_{11} = 1$. Thus, the task of understanding how the Markov chain generators M_Σ^{ur} , M_Σ^{us} and M_Σ^{uo} should be defined boils down to understanding how the probabilities p_1, \dots, p_{11} should be calculated by M_Σ^{ur} , M_Σ^{us} and M_Σ^{uo} in order to guarantee the properties discussed above. We start by explaining how the probabilities are calculated by M_Σ^{us} , which will then help us to explain how the probabilities are calculated by M_Σ^{ur} . We finally discuss M_Σ^{uo} , which is the simplest one.

Uniform Sequences. For a sequence $s \in \text{RS}(D, \Sigma)$, let $\text{CRS}_s(D, \Sigma)$ be the set of all sequences of $\text{CRS}(D, \Sigma)$ that have s as a prefix. Thus, $\text{CRS}_\varepsilon(D, \Sigma) = \text{CRS}(D, \Sigma)$ being the set of leaves. Hence, for $M_\Sigma^{\text{us}}(D) = (V, E, \mathbf{P})$ to induce the uniform distribution over the leaves, it suffices, for $s, s' \in \text{RS}(D, \Sigma)$ with $s' \in \text{Ops}_s(D, \Sigma)$, to let

$$\mathbf{P}(s, s') = \frac{|\text{CRS}_{s'}(D, \Sigma)|}{|\text{CRS}_s(D, \Sigma)|}.$$

Observe that

$$|\text{CRS}_\varepsilon(D, \Sigma)| = 9$$

$$|\text{CRS}_{-f_1}(D, \Sigma)| = |\text{CRS}_{-f_3}(D, \Sigma)| = 3$$

$$|\text{CRS}_{\{-f_1, f_2\}}(D, \Sigma)| = |\text{CRS}_{-f_2}(D, \Sigma)| = |\text{CRS}_{\{-f_2, f_3\}}(D, \Sigma)| = 1.$$

Hence, $p_1 = p_5 = \frac{3}{9}$, $p_2 = p_3 = p_4 = \frac{1}{9}$. Similarly, we obtain that $p_6 = p_7 = p_8 = \frac{1}{3}$, and $p_9 = p_{10} = p_{11} = \frac{1}{3}$. Thus, $\text{RL}(M_\Sigma^{\text{us}}(D)) = \text{CRS}(D, \Sigma)$, and $\pi(s) = \frac{1}{9}$, for each $s \in \text{RL}(M_\Sigma^{\text{us}}(D))$, with π being the leaf distribution of $M_\Sigma^{\text{us}}(D)$, as needed.

Uniform Repairs. Since multiple complete sequences can lead to the same database (e.g., $-f_1, -\{f_2, f_3\}$ and $-f_3, -\{f_1, f_2\}$) we would like to have a mechanism that gives non-zero probability to exactly one such sequence. To this end, for each set of complete sequences that lead to the same consistent database, we identify a representative one. We say that a (D, Σ) -repairing sequence $s \in \text{CRS}(D, \Sigma)$ is *canonical* if there is no $s' \in \text{CRS}(D, \Sigma)$ such that $s(D) = s'(D)$ and $s' < s$ for some arbitrary ordering $<$ over the set $\text{RS}(D, \Sigma)$. Let $\text{CanCRS}(D, \Sigma)$ be the set of all sequences of $\text{CRS}(D, \Sigma)$ that are canonical. Furthermore, for a sequence $s \in \text{RS}(D, \Sigma)$, we write $\text{CanCRS}_s(D, \Sigma)$ for the set of all sequences s' of $\text{CanCRS}(D, \Sigma)$ that have s as a prefix. Hence, for $s \in \text{RS}(D, \Sigma)$, $\text{CanCRS}_s(D, \Sigma)$ is the set of canonical leaves of the subtree rooted at s , with $\text{CanCRS}_\epsilon(D, \Sigma) = \text{CanCRS}(D, \Sigma)$ being the set of canonical leaves of the tree. We can now follow the same approach discussed above for uniform sequences with the key difference that only canonical sequences are considered. In other words, for $M_\Sigma^{\text{ur}}(D) = (V, E, \mathbf{P})$ to induce the uniform distribution over the set of operational repairs, it suffices, for nodes $s, s' \in \text{RS}(D, \Sigma)$ with $s' \in \text{Ops}_s(D, \Sigma)$, to let

$$\mathbf{P}(s, s') = \frac{|\text{CanCRS}_{s'}(D, \Sigma)|}{|\text{CanCRS}_s(D, \Sigma)|}.$$

Notice that $\mathbf{P}(s, s')$ is not defined if the subtree T_s rooted at s has no canonical leaves, i.e., $\text{CanCRS}_s(D, \Sigma) = \emptyset$. In this case, none of the leaves of T_s is reachable with non-zero probability, and thus, $\mathbf{P}(s, s')$ can get an arbitrary probability, e.g., $\frac{1}{|\text{Ops}_s(D, \Sigma)|}$.

Let us illustrate the above discussion. Assuming, e.g., that for $s, s' \in \text{RS}(D, \Sigma)$, $s < s'$ iff s comes before s' in a depth-first traversal of the tree, we have that $\text{CanCRS}(D, \Sigma)$ consists of the sequences

$$-f_1, -f_2 \quad -f_1, -f_3 \quad -f_1, -\{f_2, f_3\} \quad -f_2 \quad -\{f_2, f_3\}.$$

Therefore, we get that

$$\begin{aligned} |\text{CanCRS}_\epsilon(D, \Sigma)| &= 5 & |\text{CanCRS}_{-f_1}(D, \Sigma)| &= 3 \\ |\text{CanCRS}_{-f_2}(D, \Sigma)| &= |\text{CanCRS}_{-\{f_2, f_3\}}(D, \Sigma)| &= 1 \\ |\text{CanCRS}_{-\{f_1, f_2\}}(D, \Sigma)| &= |\text{CanCRS}_{-f_3}(D, \Sigma)| &= 0. \end{aligned}$$

Hence, $p_1 = \frac{3}{5}$, $p_2 = p_5 = 0$, $p_3 = p_4 = \frac{1}{5}$, $p_6 = p_7 = p_8 = \frac{1}{3}$, and $p_9 = p_{10} = p_{11} = \frac{1}{3}$. Thus, $\text{RL}(M_\Sigma^{\text{ur}}(D)) = \text{CanCRS}(D, \Sigma)$, and $\pi(s) = \frac{1}{5}$, for each $s \in \text{RL}(M_\Sigma^{\text{ur}}(D))$, with π being the leaf distribution of $M_\Sigma^{\text{ur}}(D)$. Hence, $\text{ORep}(D, M_\Sigma^{\text{ur}}) = \{\emptyset, \{f_1\}, \{f_2\}, \{f_3\}, \{f_1, f_3\}\}$ with $\mathbf{P}_{D, M_\Sigma^{\text{ur}}}(D') = \frac{1}{5}$, for each $D' \in \text{ORep}(D, M_\Sigma^{\text{ur}})$, as needed.

Uniform Operations. For M_Σ^{uo} we simply follow our intention. In particular, $M_\Sigma^{\text{uo}}(D) = (V, E, \mathbf{P})$ is such that, for nodes $s, s' \in \text{RS}(D, \Sigma)$ with $s' \in \text{Ops}_s(D, \Sigma)$, $\mathbf{P}(s, s') = \frac{1}{|\text{Ops}_s(D, \Sigma)|}$. Thus, $p_1 = p_2 = p_3 = p_4 = p_5 = \frac{1}{5}$, $p_6 = p_7 = p_8 = \frac{1}{3}$, and $p_9 = p_{10} = p_{11} = \frac{1}{3}$. Notice that, unlike the Markov chain generators M_Σ^{ur} and M_Σ^{us} discussed above, M_Σ^{uo} is intrinsically “local” in the sense that the probabilities assigned to operations at a certain step are completely determined by that step. As we shall see, the local nature of M_Σ^{uo} has a significant impact on operational CQA when it comes to approximations.

Our Main Objective. The data complexity of OCQA for *arbitrary* Markov chain generators has been already studied in [5], showing that it is, in general, intractable. In particular:

THEOREM 4.1 ([5]). *There exist a set Σ of primary keys, a repairing Markov chain generator M_Σ w.r.t. Σ , and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma, Q)$ is $\#\text{P}$ -hard.*

With the above intractability result in place, the authors of [5] asked whether $\text{OCQA}(\Sigma, M_\Sigma, Q(\bar{x}))$ is approximable, i.e., whether the target probability can be approximated via a *fully polynomial-time randomized approximation scheme* (FPRAS, for short). Formally, an FPRAS for $\text{OCQA}(\Sigma, M_\Sigma, Q(\bar{x}))$ is a randomized algorithm A that takes as input a database D , a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$, $\epsilon > 0$, and $0 < \delta < 1$, runs in polynomial time in $\|D\|, \|\bar{c}\|, 1/\epsilon$ and $\log(1/\delta)$, and produces a random variable $A(D, \bar{c}, \epsilon, \delta)$ such that

$$\Pr(|A(D, \bar{c}, \epsilon, \delta) - \mathbf{P}_{M_\Sigma, Q}(D, \bar{c})| \leq \epsilon \cdot \mathbf{P}_{M_\Sigma, Q}(D, \bar{c})) \geq 1 - \delta.$$

It was shown that the problem in question does not admit an FPRAS, under the widely accepted complexity assumption that $\text{RP} \neq \text{NP}$. Recall that RP is the complexity class of problems that are efficiently solvable via a randomized algorithm with a bounded one-sided error (i.e., the answer may mistakenly be “no”) [2].

THEOREM 4.2 ([5]). *Unless $\text{RP} = \text{NP}$, there exist a set Σ of primary keys, a Markov chain generator M_Σ w.r.t. Σ , and a CQ Q such that there is no FPRAS for $\text{OCQA}(\Sigma, M_\Sigma, Q)$.*

Having the natural Markov chain generators discussed above in place, the question is how the complexity of exact and approximate operational CQA is affected, i.e., how Theorems 4.1 and 4.2 are affected if we consider these more refined Markov chain generators instead of an arbitrary one. The goal of this work is to perform such a complexity analysis. Our main findings are as follows:

- (1) The complexity of exact operational CQA remains $\#\text{P}$ -hard, even in the case of primary keys.
- (2) Operational CQA is approximable, i.e., it admits an FPRAS, if we focus on primary keys.
- (3) In the case of arbitrary keys and FDs, although the Markov chain generators based on uniform repairs and sequences do not lead (or it remains open whether they lead) to the approximability of operational CQA, the Markov chain generator based on uniform operations renders the problem approximable.² The latter should be attributed to the “local” nature of the Markov chain generator based on uniform operations.

The rest of the paper is devoted to discussing the high-level ideas underlying the above results; the formal proofs are in the appendix.

5 UNIFORM REPAIRS

We start our complexity analysis by considering the Markov chain generator based on uniform repairs, and show the following result:

- THEOREM 5.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma^{\text{ur}}, Q)$ is $\#\text{P}$ -hard.*
(2) *For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{ur}}, Q)$ admits an FPRAS.*

¹As usual, $\|o\|$ denotes the size of the encoding of a syntactic object o .

²In the case of FDs, the approximability result holds assuming that only operations that remove a single fact (not a pair of facts) are considered; this is discussed in Section 7.

(3) Unless $RP = NP$, there exist a set Σ of FDs, and a CQ Q such that there is no FPRAS for $OCQA(\Sigma, M_{\Sigma}^{ur}, Q)$.

Notice that the above result does not cover the case of arbitrary keys, which remains an open problem. We can extract, however, from the proof of item (3) that for keys, unless $RP = NP$, the problem of *counting* the number of operational repairs does not admit an FPRAS. We see this as an indication that item (3) holds even in the case of keys. We now discuss how Theorem 5.1 is shown.

We start with the simple observation that, for a database D , a set Σ of FDs, a CQ $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,

$$P_{M_{\Sigma}^{ur}, Q}(D, \bar{c}) = \frac{|\{D' \in \text{CORep}(D, \Sigma) \mid \bar{c} \in Q(D')\}|}{|\text{CORep}(D, \Sigma)|}.$$

This ratio is the percentage of candidate operational repairs of D w.r.t. Σ that entail $Q(\bar{c})$, which we call the *repair relative frequency* of $Q(\bar{c})$ w.r.t. D and Σ , and denote $\text{rrfreq}_{\Sigma, Q}(D, \bar{c})$. Therefore, we can conveniently restate the problem $OCQA(\Sigma, M_{\Sigma}^{ur}, Q)$ as the problem of computing the repair relative frequency of $Q(\bar{c})$ w.r.t. D and Σ , which does not depend on the Markov chain generator M_{Σ}^{ur} :

PROBLEM : $\text{RRFreq}(\Sigma, Q(\bar{x}))$
INPUT : A database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$.
OUTPUT : $\text{rrfreq}_{\Sigma, Q}(D, \bar{c})$.

We proceed to discuss how we establish Theorem 5.1 by directly considering the problem $\text{RRFreq}(\Sigma, Q)$ instead of $OCQA(\Sigma, M_{\Sigma}^{ur}, Q)$; further details can be found in Appendix B.

Item (1). We show that $\text{RRFreq}(\Sigma, Q)$ is $\#\text{P}$ -hard for a set Σ consisting of a single key of the form $R : A \rightarrow B$, where R is a binary relation name with (A, B) being its tuple of attributes, and a Boolean CQ Q . This is done via a polynomial-time Turing reduction from a graph-theoretic problem called $\#\text{H-Coloring}$, where H is an undirected graph, to $\text{RRFreq}(\Sigma, Q)$. The problem $\#\text{H-Coloring}$ takes as input an undirected graph G , and asks for the number of homomorphisms from G to H . The key of the proof is to carefully choose H so that (i) $\#\text{H-Coloring}$ is $\#\text{P}$ -hard, and (ii) it allows us to devise the desired polynomial-time Turing reduction, i.e., for an undirected graph G , we can construct in polynomial time in $\|G\|$ a database D_G such that the number of homomorphisms from G to H can be computed in polynomial time in $\|G\|$, assuming that we have access to an oracle for the problem $\text{RRFreq}(\Sigma, Q)$, which we can call to compute the number $\text{rrfreq}_{\Sigma, Q}(D_G, ())$; we use $()$ to denote the empty tuple. For choosing H , we exploit an interesting dichotomy from [10], which characterizes when $\#\text{H-Coloring}$ is solvable in polynomial time or is $\#\text{P}$ -hard, depending on the structure of H .

Item (2). For showing that, for a set Σ of primary keys and a CQ Q , $\text{RRFreq}(\Sigma, Q)$ admits an FPRAS, we rely on Monte Carlo sampling. We first show the existence of an efficient sampler:

LEMMA 5.2. *Given a database D , and a set Σ of primary keys, we can sample elements of $\text{CORep}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

The above lemma tells us that there exists a randomized algorithm SampleRep that takes as input D and Σ , runs in polynomial time in $\|D\|$, and produces a random variable $\text{SampleRep}(D, \Sigma)$ such that $\Pr(\text{SampleRep}(D, \Sigma) = D') = \frac{1}{|\text{CORep}(D, \Sigma)|}$ for every

database $D' \in \text{CORep}(D, \Sigma)$. Notice, however, that the efficient sampler provided by Lemma 5.2 does not immediately imply the existence of an FPRAS for $\text{RRFreq}(\Sigma, Q)$ since the number of samples should be proportional to $\frac{1}{\text{rrfreq}_{\Sigma, Q}(D, \bar{c})}$ [8]. Hence, to obtain an FPRAS using Monte Carlo sampling, we need show that the repair relative frequency is never “too small”.

LEMMA 5.3. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot \|D\|)^{\|Q\|}}$$

whenever $\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) > 0$.

Given a set Σ of primary keys and a CQ Q , by exploiting Lemmas 5.2 and 5.3, we can easily devise an FPRAS for $\text{RRFreq}(\Sigma, Q)$.

Item (3). For showing that there exist a set Σ of FDs and a CQ Q such that, unless $RP = NP$, there is no FPRAS for $\text{RRFreq}(\Sigma, Q)$, we provide a rather involved proof that proceeds in two main steps. We first give an auxiliary lemma that is needed by both steps.

An undirected graph G is called *non-trivially connected* if it contains at least two nodes, and is connected. We write $\text{IS}(G)$ for the set that collects all the independent sets of G . Recall that the *conflict graph* of a database D w.r.t. a set Σ of FDs, denoted $\text{CG}(D, \Sigma)$, is an undirected graph whose node set is D , and it has an edge between f and g if $\{f, g\} \not\models \Sigma$. A database D is *non-trivially Σ -connected* if $\text{CG}(D, \Sigma)$ is non-trivially connected. We then show the following:

LEMMA 5.4. *Consider a non-trivially Σ -connected database D , where Σ is a set of FDs. It holds that $|\text{CORep}(D, \Sigma)| = |\text{IS}(\text{CG}(D, \Sigma))|$.*

Having the above auxiliary lemma in place, we can now describe the two steps of the proof underlying Theorem 5.1(3). The first step establishes the following inapproximability result about keys.

PROPOSITION 5.5. *Unless $RP = NP$, there exists a set Σ of keys over $\{R\}$ such that, given a non-trivially Σ -connected database D , the problem of computing $|\text{CORep}(D, \Sigma)|$ does not admit an FPRAS.*

The above result exploits the fact that, unless $RP = NP$, the problem of counting the number of independent sets of a non-trivially connected undirected graph of bounded degree does not admit an FPRAS.³ In particular, we show that there exists a set Σ_K of keys over the schema $S = \{R/\Delta + 1\}$ such that the following holds: given a non-trivially connected undirected graph G of bounded degree Δ , we can construct in polynomial time in $\|G\|$ a database D_G over S such that $\text{CG}(D_G, \Sigma_K)$ is isomorphic to G . Thus, by Lemma 5.4, $|\text{CORep}(D_G, \Sigma_K)| = |\text{IS}(G)|$. The construction of D_G exploits Vizing’s Theorem, which states that a graph of degree Δ always has a $(\Delta + 1)$ -edge-coloring, as well as the fact that such an edge-coloring can be constructed in polynomial time as long as Δ is bounded [20]. Hence, given a database D , assuming that the problem of computing the number $|\text{CORep}(D, \Sigma_K)|$ admits an FPRAS, we can conclude that the problem of counting the number of independent sets of a non-trivially connected undirected graph of bounded degree admits an FPRAS, which, unless $RP = NP$, leads

³This result is known for arbitrary, not necessarily non-trivially connected graphs [22]. Thus, for our purposes, we had to strengthen it to non-trivially connected graphs.

to a contradiction. Therefore, Proposition 5.5 follows with $\Sigma = \Sigma_K$. Notice that Proposition 5.5 tells us that for keys, unless $RP = NP$, the problem of counting the number of operational repairs does not admit an FPRAS. As said above, we see this as an indication that item (3) of Theorem 5.1 holds even for keys.

We then proceed to show that, unless $RP = NP$, the existence of an FPRAS for $RRFreq(\Sigma, Q)$, where Σ is a set of FDs and Q a CQ, would contradict Proposition 5.5. Let Σ_K be the set of keys provided by Proposition 5.5. We show the following auxiliary result:

LEMMA 5.6. *Assume that $RRFreq(\Sigma, Q)$ admits an FPRAS, for every set Σ of FDs and CQ Q . Given a non-trivially Σ_K -connected database D , the problem of computing $|\text{CORep}(D, \Sigma_K)|$ admits an FPRAS.*

To establish the above result, we show that there exists a set Σ_F of FDs such that, for every non-trivially Σ_K -connected database D , we can construct in polynomial time in $\|D\|$ a database D_F such that $\text{CG}(D_F, \Sigma_F)$ consists of a graph G that is isomorphic to $\text{CG}(D, \Sigma_K)$, and an additional node that is connected via an edge with every node of G . Therefore, by Lemma 5.4, we get that

$$|\text{CORep}(D_F, \Sigma_F)| = |\text{CORep}(D, \Sigma_K)| + 1.$$

Let us clarify that this is the place where we need the power of FDs; it is unclear how we can devise a set of keys that has the same properties as Σ_F . We then construct an atomic Boolean CQ Q_F with

$$\text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) = \frac{1}{|\text{CORep}(D_F, \Sigma_F)|} = \frac{1}{|\text{CORep}(D, \Sigma_K)| + 1};$$

we use $()$ to denote the empty tuple. Now, by exploiting the above equality, the fact that D_F can be constructed in polynomial time, and the FPRAS for $RRFreq(\Sigma_F, Q_F)$ (which exists by hypothesis), we can devise an FPRAS for the problem of computing $|\text{CORep}(D, \Sigma_K)|$ given a non-trivially Σ_K -connected database D , as claimed.

It is now straightforward to see that from Proposition 5.5 and Lemma 5.6, we get that, unless $RP = NP$, there exist a set Σ of FDs and a CQ Q such that there is no FPRAS for $RRFreq(\Sigma, Q)$.

6 UNIFORM SEQUENCES

We now concentrate on the Markov chain generator based on uniform sequences, and establish the following complexity result.

- THEOREM 6.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma^{\text{us}}, Q)$ is $\#\text{P}$ -hard.*
(2) *For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{us}}, Q)$ admits an FPRAS.*

Notice that the above result does not cover the cases of arbitrary keys and FDs. Unfortunately, despite our efforts, we have not managed to prove or disprove the existence of an FPRAS for the problem in question. We conjecture that there is no FPRAS even for keys, i.e., unless $RP = NP$, there exist a set Σ of keys, and a CQ Q such that there is no FPRAS for $\text{OCQA}(\Sigma, M_\Sigma^{\text{us}}, Q)$. We proceed to discuss how Theorem 6.1 is shown.

As for Theorem 5.1, we can conveniently restate the problem in question as a problem of computing a ‘‘relative frequency’’ ratio that does not depend on the Markov chain generator. In particular, for a database D , a set Σ of FDs, a CQ $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,

$$P_{M_\Sigma^{\text{us}}, Q}(D, \bar{c}) = \frac{|\{s \in \text{CRS}(D, \Sigma) \mid \bar{c} \in Q(s(D))\}|}{|\text{CRS}(D, \Sigma)|}.$$

This ratio is the percentage of complete (D, Σ) -repairing sequences that lead to an operational repair that entails $Q(\bar{c})$, which we call the *sequence relative frequency* of $Q(\bar{c})$ w.r.t. D and Σ , and denote $\text{srfreq}_{\Sigma, Q}(D, \bar{c})$. Thus, we can restate $\text{OCQA}(\Sigma, M_\Sigma^{\text{us}}, Q)$ as the problem of computing the sequence relative frequency of $Q(\bar{c})$ w.r.t. D and Σ , which is independent from the Markov chain generator M_Σ^{us} :

PROBLEM : $\text{SRFreq}(\Sigma, Q(\bar{x}))$
INPUT : A database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$.
OUTPUT : $\text{srfreq}_{\Sigma, Q}(D, \bar{c})$.

We now discuss how we establish Theorem 6.1 by directly considering the problem $\text{SRFreq}(\Sigma, Q)$ instead of $\text{OCQA}(\Sigma, M_\Sigma^{\text{us}}, Q)$; further details can be found in Appendix C.

Item (1). Let Σ and Q be the singleton set of primary keys and the Boolean CQ, respectively, for which $RRFreq(\Sigma, Q)$ is $\#\text{P}$ -hard; Σ and Q are obtained from the proof of Theorem 5.1(1). We show that also $\text{SRFreq}(\Sigma, Q)$ is $\#\text{P}$ -hard via a polynomial-time Turing reduction from $\#\text{H}$ -Coloring. Actually, we can exploit the same construction as in the proof of item (1) of Theorem 5.1.

Item (2). For showing that, for a set Σ of primary keys and a CQ $Q(\bar{x})$, $\text{SRFreq}(\Sigma, Q)$ admits an FPRAS, we rely again on Monte Carlo sampling. We first show that an efficient sampler exists. This relies on a non-trivial technical lemma, which states that, for a database D , $|\text{CRS}(D, \Sigma)|$ can be computed in polynomial time in $\|D\|$.

LEMMA 6.2. *For a database D , and a set Σ of primary keys, we can sample elements of $\text{CRS}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

To establish that the problem in question admits an FPRAS based on Monte Carlo sampling, it remains to show the following:

LEMMA 6.3. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{srfreq}_{\Sigma, Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot \|D\|)^{\|Q\|}}$$

whenever $\text{srfreq}_{\Sigma, Q}(D, \bar{c}) > 0$.

Given a set Σ of primary keys and a CQ Q , by exploiting Lemmas 6.2 and 6.3, we can easily devise an FPRAS for $\text{SRFreq}(\Sigma, Q)$.

7 UNIFORM OPERATIONS

We finally consider the Markov chain generator based on uniform operations, and establish the following complexity result.

- THEOREM 7.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ is $\#\text{P}$ -hard.*
(2) *For a set Σ of keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ admits an FPRAS.*

Notice that the above result does not cover the case of FDs, which remains an open problem. However, as we explain below, for FDs we can establish an approximability result under the assumption that only operations that remove a single fact (not a pair of facts) are considered. But let us first discuss the proof of Theorem 7.1.

Unlike Theorems 5.1 and 6.1 presented above, there is no obvious way to conveniently restate the problem of interest as a problem

of computing a “relative frequency” ratio. Thus, the proof of Theorem 7.1, which we discuss next, has to deal with OCQA($\Sigma, M_\Sigma^{\text{uo}}, Q$) for a set Σ of FDs and a CQ Q ; details are in Appendix D.

Item (1). As we did for item (1) of Theorem 6.1, we reuse the construction underlying the proof of item (1) of Theorem 5.1.

Item (2). We show that OCQA($\Sigma, M_\Sigma^{\text{uo}}, Q$), where Σ is a set of keys and Q a CQ, admits an FPRAS by relying once again on Monte Carlo sampling. The existence of an efficient sampler follows easily from the definition of the Markov chain generator M_Σ^{uo} . In particular:

LEMMA 7.2. *Given a database D , and a set Σ of keys, we can sample elements of $\text{RL}(M_\Sigma^{\text{uo}}(D))$ according to the leaf distribution of $M_\Sigma^{\text{uo}}(D)$ in polynomial time in $\|D\|$.*

The interesting task towards an FPRAS for the problem in question is to show that the target probability is never “too small”.

PROPOSITION 7.3. *Consider a set Σ of keys, and a CQ $Q(\bar{x})$. There is a polynomial pol such that, for every database D , and $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) \geq \frac{1}{\text{pol}(\|D\|)}$$

whenever $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) > 0$.

We proceed to discuss the main ideas underlying the proof of the above result. For the sake of clarity, we focus on atomic queries, i.e., CQs with only one atom. The generalization to arbitrary CQs can be found in the appendix. In the sequel, let Σ be a set of keys, $Q(\bar{x})$ an atomic query, D a database, and \bar{c} a tuple of $\text{dom}(D)^{|\bar{x}|}$.

Clearly, if there is no homomorphism h from Q to D with $h(\bar{x}) = \bar{c}$, then $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) = 0$. Assume now that such a homomorphism h exists, and let f be the fact of D obtained after applying h to the single atom of Q . It is not difficult to see that

$$P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) \geq \underbrace{\sum_{D' \in \text{OREp}(D, M_\Sigma^{\text{uo}}) \text{ and } f \in D'} P_{D, M_\Sigma^{\text{uo}}}(D')}_{\Lambda}$$

Thus, it suffices to show that there exists a polynomial pol such that $\Lambda \geq \frac{1}{\text{pol}(\|D\|)}$. Let S_f and $S_{\neg f}$ be the sets of sequences of $\text{RL}(M_\Sigma^{\text{uo}}(D))$ that keep f and remove f , respectively, i.e.,

$$\begin{aligned} S_f &= \{s \in \text{RL}(M_\Sigma^{\text{uo}}(D)) \mid f \in s(D)\} \\ S_{\neg f} &= \{s \in \text{RL}(M_\Sigma^{\text{uo}}(D)) \mid f \notin s(D)\}. \end{aligned}$$

With π being the leaf distribution of $M_\Sigma^{\text{uo}}(D)$, $\Lambda = \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}}$, where

$$\Lambda_f = \sum_{s \in S_f} \pi(s) \quad \text{and} \quad \Lambda_{\neg f} = \sum_{s \in S_{\neg f}} \pi(s).$$

Therefore, to establish the desired lower bound $\frac{1}{\text{pol}(\|D\|)}$ for Λ , it suffices to show that there exists a polynomial pol' such that $\Lambda_{\neg f} \leq \text{pol}'(\|D\|) \cdot \Lambda_f$. Indeed, in this case we can conclude that

$$\Lambda = \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}} \geq \frac{\Lambda_f}{\Lambda_f + \text{pol}'(\|D\|) \cdot \Lambda_f} = \frac{1}{1 + \text{pol}'(\|D\|)},$$

and the claim follows with $\text{pol}(\|D\|) = 1 + \text{pol}'(\|D\|)$. The rest of the proof is devoted to showing that a polynomial pol' such that $\Lambda_{\neg f} \leq \text{pol}'(\|D\|) \cdot \Lambda_f$ exists.

To get this inequality, we establish a rather involved technical lemma that relates the sequences of $S_{\neg f}$ with the sequences of S_f ; as usual, we write π for the leaf distribution of $M_\Sigma^{\text{uo}}(D)$:

LEMMA 7.4. *There exists a function $F : S_{\neg f} \rightarrow S_f$ such that:*

- (1) *There exists a polynomial pol'' such that, for every $s \in S_{\neg f}$,*

$$\pi(s) \leq \text{pol}''(\|D\|) \cdot \pi(F(s)).$$
- (2) *For every $s' \in S_f$, $|\{s \in S_{\neg f} \mid F(s) = s'\}| \leq 2 \cdot \|D\| - 1$.*

For showing item (1) of Lemma 7.4, we transform each sequence $s \in S_{\neg f}$ into a sequence $s' \in S_f$, and let $F(s) = s'$. This is done by first deleting or replacing the operation op in s that removes f . In particular, if $op = -f$, then we simply delete it; otherwise, if $op = -\{f, g\}$, then we replace it with the operation $-g$. Notice, however, that there is no guarantee that the sequence \hat{s} , obtained after removing op from s , is a complete sequence of $\text{CRS}(D, \Sigma)$. This is because $s(D)$ might contain facts that are in a conflict with f , and thus, by keeping f , there is no guarantee that $\hat{s}(D) \models \Sigma$. We then convert \hat{s} into a complete sequence s' by simply adding at the end of \hat{s} additional operations (in some arbitrary order) that resolve all the conflicts. Now, to show that $\pi(s) \leq \text{pol}''(\|D\|) \cdot \pi(s')$, for some polynomial pol'' , we rely on the following two crucial facts: (1) Although the probabilities of the operations in s' coming after the operation in s that removes f might decrease, we can show that they do not decrease “too much”. (2) The number of operations that we need to add at the end of \hat{s} in order to get s' depends only on Σ (not on $\|D\|$). More precisely, by exploiting the fact that Σ consists of keys, we can show that f can be in a conflict with at most $k \geq 0$ facts of $s(D)$, where k is the number of keys in Σ over the relation name of f . This implies that we do not need to add more than k operations at the end of \hat{s} . Note that the above facts do *not* hold for FDs. To establish that $\pi(s) \leq \text{pol}''(\|D\|) \cdot \pi(s')$ using the above facts, we rely on the Cauchy–Schwarz inequality for n -dimensional Euclidean spaces. Finally, once we have F in place, it is then not difficult to show item (2) via a combinatorial argument.

It is now easy to establish the existence of the polynomial pol' such that $\Lambda_{\neg f} \leq \text{pol}'(\|D\|) \cdot \Lambda_f$. Indeed, with F and pol'' being the function and the polynomial, respectively, provided by Lemma 7.4,

$$\begin{aligned} \Lambda_{\neg f} &= \sum_{s \in S_{\neg f}} \pi(s) \leq \sum_{s \in S_{\neg f}} \text{pol}''(\|D\|) \cdot \pi(F(s)) \\ &\leq \text{pol}''(\|D\|) \cdot (2 \cdot \|D\| - 1) \cdot \sum_{s \in S_f} \pi(s) \\ &= \text{pol}''(\|D\|) \cdot (2 \cdot \|D\| - 1) \cdot \Lambda_f, \end{aligned}$$

and the claim follows with $\text{pol}'(\|D\|) = \text{pol}''(\|D\|) \cdot (2 \cdot \|D\| - 1)$.

An FPRAS for FDs. Recall that Theorem 7.1 does not cover the case of FDs, which remains an open problem. At this point, one may wonder whether Monte Carlo sampling can be used for devising an FPRAS in the case of FDs. Indeed, the efficient sampler provided by Lemma 7.2 holds even for FDs since the proof of that lemma does not exploit keys in any way, but only the “local” nature of the Markov chain generator. However, we do not have a result analogous to Proposition 7.3, which states that the target probability is never “too small”. In fact, there exist a set Σ of FDs, a Boolean atomic query Q , and a family of databases $\{D_n\}_{n>0}$ with $|D_n| = n$, such that $0 < P_{M_\Sigma^{\text{uo}}, Q}(D_n, ()) \leq \frac{1}{2^{n-1}}$; the proof is in the appendix. Hence,

for devising an FPRAS in the case of FDs (if it exists), we need a more sophisticated machinery than the one based on Monte Carlo sampling. On the other hand, we can establish a result analogous to Proposition 7.3 for FDs, assuming that only operations that remove a single fact (not a pair of facts) are considered. Given a set Σ of FDs, let $M_{\Sigma}^{\text{uo},1}$ be the Markov chain generator defined as M_{Σ}^{uo} , with the difference that only sequences consisting of operations that remove a single fact are considered. We then get the following:

THEOREM 7.5. *For a set Σ of FDs, and a CQQ, $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{uo},1}, Q)$ admits an FPRAS.*

Note that singleton operations do not alter the data complexity of exact operational CQA; we can show that item (1) of Theorem 7.1 continues to hold. Let us also clarify that focusing on singleton operations does not affect Theorem 5.1 and Theorem 6.1; all the details about these results can be found in Appendix E.

8 FUTURE WORK

Although we understand pretty well uniform operational CQA, there are still interesting open problems on approximability: (i) the case of keys and uniform repairs (we only have a negative result for the problem of counting repairs), (ii) the case of keys/FDs and uniform sequences, and (iii) the case of FDs and uniform operations (we only have a positive result assuming singleton operations).

REFERENCES

- [1] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. 1999. Consistent Query Answers in Inconsistent Databases. In *PODS*. 68–79.
- [2] Sanjeev Aror and Boaz Barak. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press.
- [3] Marco Calautti, Marco Console, and Andreas Pieris. 2019. Counting Database Repairs under Primary Keys Revisited. In *PODS*. 104–118.
- [4] Marco Calautti, Marco Console, and Andreas Pieris. 2021. Benchmarking Approximate Consistent Query Answering. In *PODS*. 233–246.
- [5] Marco Calautti, Leonid Libkin, and Andreas Pieris. 2018. An Operational Approach to Consistent Query Answering. In *PODS*. 239–251.
- [6] Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. 2021. Counting Database Repairs Entailing a Query: The Case of Functional Dependencies. *CoRR abs/2112.09617 (2021)*.
- [7] Jan Chomicki and Jerzy Marcinkowski. 2005. Minimal-change integrity maintenance using tuple deletions. *Inf. Comput.* 197, 1-2 (2005), 90–121.
- [8] Paul Dagum, Richard M. Karp, Michael Luby, and Sheldon M. Ross. 2000. An Optimal Algorithm for Monte Carlo Estimation. *SIAM J. Comput.* 29, 5 (2000), 1484–1496.
- [9] Nilesh N. Dalvi and Dan Suciu. 2007. Management of probabilistic data: foundations and challenges. In *PODS*. 1–12.
- [10] Martin E. Dyer and Catherine S. Greenhill. 2000. The complexity of counting graph homomorphisms. *Random Struct. Algorithms* 17, 3-4 (2000), 260–289.
- [11] Ariel Fuxman, Elham Fazli, and Renée J. Miller. 2005. ConQuer: Efficient Management of Inconsistent Databases. In *SIGMOD*. 155–166.
- [12] Ariel Fuxman and Renée J. Miller. 2007. First-order query rewriting for inconsistent databases. *J. Comput. Syst. Sci.* 73, 4 (2007), 610–635.
- [13] Floris Geerts, Fabian Pijcke, and Jef Wijsen. 2015. First-Order Under-Approximations of Consistent Query Answers. In *SUM*. 354–367.
- [14] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. 1986. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* 43 (1986), 169–188.
- [15] Richard M. Karp and Richard J. Lipton. 1980. Some Connections between Nonuniform and Uniform Complexity Classes. In *STOC*. 302–309.
- [16] Paraschos Koutris and Dan Suciu. 2014. A Dichotomy on the Complexity of Consistent Query Answering for Atoms with Simple Keys. In *ICDT*. 165–176.
- [17] Paraschos Koutris and Jef Wijsen. 2015. The Data Complexity of Consistent Query Answering for Self-Join-Free Conjunctive Queries Under Primary Key Constraints. In *PODS*. 17–29.
- [18] Paraschos Koutris and Jef Wijsen. 2021. Consistent Query Answering for Primary Keys in Datalog. *Theory Comput. Syst.* 65, 1 (2021), 122–178.
- [19] Dany Maslowski and Jef Wijsen. 2013. A dichotomy in the complexity of counting database repairs. *J. Comput. Syst. Sci.* 79, 6 (2013), 958–983.
- [20] Jayadev Misra and David Gries. 1992. A constructive proof of Vizing’s theorem. *Inform. Process. Lett.* 41, 3 (1992), 131–133.
- [21] J Scott Provan and Michael O. Ball. 1983. The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected. *SIAM J. Comput.* 12 (1983), 777–788.
- [22] Allan Sly. 2010. Computational Transition at the Uniqueness Threshold. In *FOCS*. 287–296.

A UNIFORM OPERATIONAL CQA

We provide the formal definitions of the ‘‘uniform’’ Markov chain generators discussed in Section 4, and show that they indeed capture our intention. In what follows, for a database D , a set Σ of FDs, and a sequence $s = op_1, \dots, op_n \in RS(D, \Sigma)$, we write s_0 for the empty sequence ε , and s_i for the sequence op_1, \dots, op_i , for $i \in [n]$.

A.1 Uniform Repairs

We start with the Markov chain generator based on the uniform probability distribution over the set of candidate operational repairs. As discussed in the main body, since multiple complete repairing sequences can lead to the same consistent database, we focus on canonical complete sequences. Recall that, for a database D , and a set Σ of FDs, we say that a (D, Σ) -repairing sequence $s \in CRS(D, \Sigma)$ is canonical if there is no $s' \in CRS(D, \Sigma)$ such that $s(D) = s'(D)$ and $s' < s$ for some arbitrary ordering $<$ over the set $RS(D, \Sigma)$, and we write $\text{CanCRS}(D, \Sigma)$ for the set of all sequences of $CRS(D, \Sigma)$ that are canonical. Furthermore, for a sequence $s \in RS(D, \Sigma)$, we write $\text{CanCRS}_s(D, \Sigma)$ for the set of all sequences s' of $\text{CanCRS}(D, \Sigma)$ that have s as a prefix, i.e., $s' = s \cdot s''$ for some (possibly empty) sequence s'' . We are now ready to define the desired Markov chain generator.

Definition A.1. (Uniform Repairs) Consider a set Σ of FDs. Let M_Σ^{ur} be the function assigning to a database D the (D, Σ) -repairing Markov chain (V, E, \mathbf{P}) , where, for each $(s, s') \in E$,

$$\mathbf{P}(s, s') = \begin{cases} \frac{|\text{CanCRS}_{s'}(D, \Sigma)|}{|\text{CanCRS}_s(D, \Sigma)|} & \text{if } \text{CanCRS}_s(D, \Sigma) \neq \emptyset \\ \frac{1}{|\text{Ops}_s(D, \Sigma)|} & \text{otherwise.} \end{cases} \quad \blacksquare$$

Note that the above Markov chain generator is well-defined since, for each $s \in RS(D, \Sigma)$ that is not complete,

$$|\text{CanCRS}_s(D, \Sigma)| = \sum_{s' \in \text{Ops}_s(D, \Sigma)} |\text{CanCRS}_{s'}(D, \Sigma)|,$$

and thus, for a non-leaf node $s \in V$, $\sum_{t \in \{s' \mid (s, s') \in E\}} \mathbf{P}(s, t) = 1$. We now show that the above definition captures our intention.

PROPOSITION A.2. *Consider a set Σ of FDs. For every database D :*

- (1) $\text{ORep}(D, M_\Sigma^{\text{ur}}) = \text{CRep}(D, \Sigma)$.
- (2) For every $D' \in \text{ORep}(D, M_\Sigma^{\text{ur}})$, $\mathbf{P}_{D, M_\Sigma^{\text{ur}}}(D') = \frac{1}{|\text{ORep}(D, M_\Sigma^{\text{ur}})|}$.

PROOF. Item (1). It suffices to prove that $\text{RL}(M_\Sigma^{\text{ur}}(D)) = \text{CanCRS}(D, \Sigma)$. Let $M_\Sigma^{\text{ur}}(D) = (V, E, \mathbf{P})$, and assume that π is its leaf distribution. Recall that for a sequence $s = op_1, \dots, op_n \in CRS(D, \Sigma)$, $\pi(s) = \mathbf{P}(s_0, s_1) \cdots \mathbf{P}(s_{n-1}, s_n)$.

(\supseteq) Assume first that $s = op_1, \dots, op_n \in \text{CanCRS}(D, \Sigma)$. This implies that $\text{CanCRS}_{s_i}(D, \Sigma) \neq \emptyset$, for each $i \in \{0, 1, \dots, n\}$. Therefore, $\mathbf{P}(s_i, s_{i+1}) > 0$, for $i \in \{0, 1, \dots, n\}$, and thus $\pi(s) > 0$. The latter implies that $s \in \text{RL}(M_\Sigma^{\text{ur}}(D))$, which in turn shows that $\text{RL}(M_\Sigma^{\text{ur}}(D)) \supseteq \text{CanCRS}(D, \Sigma)$, as needed.

(\subseteq) Assume now that $s = op_1, \dots, op_n \in \text{RL}(M_\Sigma^{\text{ur}}(D))$. By contradiction, assume that $s \notin \text{CanCRS}(D, \Sigma)$. Since $s \in \text{RL}(M_\Sigma^{\text{ur}}(D))$, s must be complete. The fact that s is complete but not canonical implies that there exists $i \in \{0, \dots, n\}$ such that $\text{CanCRS}_{s_i}(D, \Sigma) = \emptyset$. In particular, let ℓ be the smallest integer in $\{0, 1, \dots, n\}$ such that

$\text{CanCRS}_{s_\ell}(D, \Sigma) = \emptyset$. Clearly, $\ell > 0$, since $\text{CanCRS}_\varepsilon(D, \Sigma)$ is always non-empty. Thus, by the first rule of the expression defining \mathbf{P} in Definition A.1, we have that $\mathbf{P}(s_{\ell-1}, s_\ell) = 0$. Hence, $\pi(s) = 0$, and thus, $s \notin \text{RL}(M_\Sigma^{\text{ur}}(D))$, which contradicts our hypothesis.

Item (2). By the proof of item (1), $\text{RL}(M_\Sigma^{\text{ur}}(D)) = \text{CanCRS}(D, \Sigma)$. Hence, we conclude that

$$|\text{ORep}(D, M_\Sigma^{\text{ur}})| = |\text{CanCRS}(D, \Sigma)| = |\text{RL}(M_\Sigma^{\text{ur}}(D))|.$$

Therefore, it suffices to show that, for $s \in \text{CanCRS}(D, \Sigma)$, $\pi(s) = \frac{1}{|\text{CanCRS}(D, \Sigma)|}$. Let $s = op_1, \dots, op_n \in \text{CanCRS}(D, \Sigma)$. Since $s \in \text{RL}(M_\Sigma^{\text{ur}}(D))$, $\pi(s)$ is equal to

$$\frac{|\text{CanCRS}_{s_1}(D, \Sigma)|}{|\text{CanCRS}_{s_0}(D, \Sigma)|} \cdots \frac{|\text{CanCRS}_{s_n}(D, \Sigma)|}{|\text{CanCRS}_{s_{n-1}}(D, \Sigma)|} = \frac{|\text{CanCRS}_{s_n}(D, \Sigma)|}{|\text{CanCRS}_{s_0}(D, \Sigma)|}.$$

Since $\text{CanCRS}_{s_0}(D, \Sigma) = \text{CanCRS}_\varepsilon(D, \Sigma) = \text{CanCRS}(D, \Sigma)$, and $\text{CanCRS}_{s_n}(D, \Sigma) = \{s_n\}$, then $\pi(s) = \frac{1}{|\text{CanCRS}(D, \Sigma)|}$, as needed. \square

A.2 Uniform Sequences

We now proceed to define the Markov chain generator based on the uniform probability distribution over the set of complete repairing sequences. It is defined similarly to the Markov chain generator above with the difference that we consider arbitrary, not necessarily canonical, complete sequences.

Definition A.3. (Uniform Sequences) Consider a set Σ of FDs. Let M_Σ^{us} be the function assigning to a database D the (D, Σ) -repairing Markov chain (V, E, \mathbf{P}) , where, for each $(s, s') \in E$,

$$\mathbf{P}(s, s') = \frac{|\text{CRS}_{s'}(D, \Sigma)|}{|\text{CRS}_s(D, \Sigma)|} \quad \blacksquare$$

Observe that the above Markov chain generator is well-defined since, for each $s \in RS(D, \Sigma)$ that is not complete,

$$|\text{CRS}_s(D, \Sigma)| = \sum_{s' \in \text{Ops}_s(D, \Sigma)} |\text{CRS}_{s'}(D, \Sigma)|,$$

and thus, for a non-leaf node $s \in V$, $\sum_{t \in \{s' \mid (s, s') \in E\}} \mathbf{P}(s, t) = 1$. We can easily show that M_Σ^{us} captures our intention:

PROPOSITION A.4. *Consider a set Σ of FDs. For every database D :*

- (1) $\text{RL}(M_\Sigma^{\text{us}}(D)) = \text{CRS}(D, \Sigma)$.
- (2) For every $s \in \text{CRS}(D, \Sigma)$, assuming that π is the leaf distribution of $M_\Sigma^{\text{us}}(D)$, $\pi(s) = \frac{1}{|\text{CRS}(D, \Sigma)|}$.

PROOF. Item (1). This item follows from the fact that each $s \in \text{RL}(M_\Sigma^{\text{us}}(D))$ is complete by definition, and each $s = op_1, \dots, op_n \in \text{CRS}(D, \Sigma)$ is such that $\text{CRS}_{s_i}(D, \Sigma) \neq \emptyset$, for $i \in \{0, \dots, n\}$, and thus $\pi(s) > 0$, where π is the leaf distribution of $M_\Sigma^{\text{us}}(D)$.

Item (2). It is shown via a proof similar to the one used above for item (2) of Proposition A.2. \square

A.3 Uniform Operations

We finally define the Markov chain generator based on the uniform probability distribution over the set of available operations at a single step of the repairing process.

Definition A.5. (Uniform Operations) Consider a set Σ of FDs. Let M_Σ^{uo} be the function assigning to a database D the (D, Σ) -repairing Markov chain (V, E, \mathbf{P}) , where, for each $(s, s') \in E$,

$$P(s, s') = \frac{1}{|\text{Ops}_s(D, \Sigma)|} \quad \blacksquare$$

It is straightforward to see that the function M_Σ^{uo} captures our intention; in fact, the following holds by definition:

PROPOSITION A.6. Consider a set Σ of FDs. For every database D :

- (1) $\text{RL}(M_\Sigma^{\text{uo}}(D)) = \text{CRS}(D, \Sigma)$.
- (2) Assuming that $M_\Sigma^{\text{uo}}(D) = (V, E, \mathbf{P})$, $(s, s') \in E$ implies $P(s, s') = \frac{1}{|\text{Ops}_s(D, \Sigma)|}$.

B PROOFS OF SECTION 5

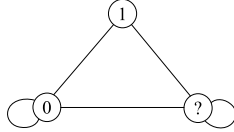
In this section, we prove the main result of Section 5, which we recall here for the sake of readability:

- THEOREM 5.1. (1) There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma^{\text{ur}}, Q)$ is $\#P$ -hard.
- (2) For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{ur}}, Q)$ admits an FPRAS.
- (3) Unless $\text{RP} = \text{NP}$, there exist a set Σ of FDs, and a CQ Q such that there is no FPRAS for $\text{OCQA}(\Sigma, M_\Sigma^{\text{ur}}, Q)$.

As discussed in Section 5, we actually need to prove the above result for the problem $\text{RRFreq}(\Sigma, Q)$.

B.1 Proof of Item (1) of Theorem 5.1

Consider the undirected graph $H = (V_H, E_H)$, where $V_H = \{0, 1, ?\}$ and $E_H = \{\{u, v\} \mid (u, v) \in (V_H \times V_H) \setminus \{(1, 1)\}\}$, i.e., the graph:



Given an undirected graph G , a homomorphism from G to H is a mapping $h : V_G \rightarrow V_H$ such that $\{u, v\} \in E_G$ implies $\{h(u), h(v)\} \in E_H$. We write $\text{hom}(G, H)$ for the set of homomorphisms from G to H . The problem $\#H$ -Coloring is defined as follows:

PROBLEM : $\#H$ -Coloring
INPUT : An undirected graph G .
OUTPUT : The number $|\text{hom}(G, H)|$.

It is implicit in [10] that $\#H$ -Coloring is $\#P$ -hard. In fact, [10] establishes the following dichotomy result: $\#H$ -Coloring is $\#P$ -hard if H has a connected component which is neither an isolated node without a loop, nor a complete graph with all loops present, nor a complete bipartite graph without loops; otherwise, it is solvable in polynomial time. Since our fixed graph H above consists of a single connected component which is neither a single node, nor a complete graph with all loops present (the loop $(1, 1)$ is missing), nor a bipartite graph, we get that $\#H$ -Coloring is indeed $\#P$ -hard.

We proceed to show via a polynomial-time Turing reduction from $\#H$ -Coloring that $\text{RRFreq}(\Sigma, Q)$ is $\#P$ -hard, where Σ and Q are

as follows. Let S be the schema $\{V/2, E/2, T/1\}$, and let (A, B) be the tuple of attributes of V . The set Σ consists of the single key

$$V : A \rightarrow B$$

and the (constant-free) Boolean CQ Q is

$$\text{Ans}() :- E(x, y), V(x, z), V(y, z), T(z).$$

Given an undirected graph $G = (V_G, E_G)$, we define the following database over S encoding G :

$$D_G = \{V(u, 0), V(u, 1) \mid u \in V_G\} \cup \{E(u, v) \mid \{u, v\} \in E_G\} \cup \{T(1)\}.$$

We then define the algorithm HOM , which accepts as input an undirected graph $G = (V_G, E_G)$, as follows:

- (1) Construct the database D_G .
- (2) Compute the number $r = \text{rrfreq}_{\Sigma, Q}(D_G, ())$.
- (3) Output the number $3^{|V_G|} \cdot (1 - r)$.

It is clear that $\text{HOM}(G)$ runs in polynomial time in $\|G\|$ assuming access to an oracle for the problem $\text{RRFreq}(\Sigma, Q)$. It remains to show that $|\text{hom}(G, H)| = \text{HOM}(G)$. Recall that

$$\text{rrfreq}_{\Sigma, Q}(D_G, ()) = \frac{|\text{CORep}(D_G, \Sigma, Q)|}{|\text{CORep}(D_G, \Sigma)|},$$

where $\text{CORep}(D_G, \Sigma, Q)$ is the set of candidate repairs D of D_G w.r.t. Σ such that $D \models Q$. Observe that there are $3^{|V_G|}$ candidate repairs of D_G w.r.t. Σ , i.e., in each such a repair D , for each node $u \in V$ of G , either $V(u, 0) \in D$ and $V(u, 1) \notin D$, or $V(u, 0) \notin D$ and $V(u, 1) \in D$, or $V(u, 0), V(u, 1) \notin D$. Therefore,

$$\text{rrfreq}_{\Sigma, Q}(D_G, ()) = \frac{|\text{CORep}(D_G, \Sigma, Q)|}{3^{|V_G|}}.$$

Thus, $\text{HOM}(G)$ coincides with

$$3^{|V_G|} \cdot \left(1 - \frac{|\text{CORep}(D_G, \Sigma, Q)|}{3^{|V_G|}} \right) = 3^{|V_G|} - |\text{CORep}(D_G, \Sigma, Q)|.$$

Since D_G has $3^{|V_G|}$ candidate repairs w.r.t. Σ , we can conclude that $\text{HOM}(G)$ is precisely the cardinality of the set $\text{CORep}(D_G, \Sigma, \neg Q)$, which collects the candidate repairs D of D_G w.r.t. Σ such that $D \not\models Q$. We proceed to show that:

LEMMA B.1. $|\text{hom}(G, H)| = |\text{CORep}(D_G, \Sigma, \neg Q)|$.

PROOF. It suffices to show that there exists a bijection from $\text{hom}(G, H)$ to $\text{CORep}(D_G, \Sigma, \neg Q)$. To this end, we define the mapping $\mu : \text{hom}(G, H) \rightarrow \mathcal{P}(D_G)$ as follows: for each $h \in \text{hom}(G, H)$,

$$\mu(h) = \{V(u, \star) \mid u \in V_G \text{ and } h(u) = \star \in \{0, 1\}\} \cup \{E(u, v) \mid \{u, v\} \in E_G\} \cup \{T(1)\}.$$

We proceed to show the following three statements:

- (1) μ is correct, that is, it is indeed a function from $\text{hom}(G, H)$ to $\text{CORep}(D_G, \Sigma, \neg Q)$.
- (2) μ is injective.
- (3) μ is surjective.

The mapping μ is correct. Consider an arbitrary homomorphism $h \in \text{hom}(G, H)$. We need to show that there exists a (D_G, Σ) -repairing sequence s_h such that $\mu(h) = s_h(D_G), s_h(D_G) \models \Sigma$ (i.e., s_h

is complete), and $Q(s_h(D_G)) = \emptyset$. Let $V_G = \{u_1, \dots, u_n\}$. Consider the sequence $s_h = op_1, \dots, op_n$ such that, for every $i \in [n]$:

$$op_i = \begin{cases} -V(u_i, 1) & \text{if } h(u_i) = 0 \\ -V(u_i, 0) & \text{if } h(u_i) = 1 \\ -\{V(u_i, 0), V(u_i, 1)\} & \text{if } h(u_i) = ? \end{cases}$$

In simple words, the homomorphism h guides the repairing process, i.e., $h(u_i) = 0$ (resp., $h(u_i) = 1$) implies $V(u_i, 0)$ (resp., $V(u_i, 1)$) should be kept, while $h(u_i) = ?$ implies none of the atoms $V(u_i, 0), V(u_i, 1)$ should be kept. It is easy to verify that s_h is indeed a (D_G, Σ) -repairing sequence s_h such that $\mu(h) = s_h(D_G)$ and $s_h(D_G) \models \Sigma$. The fact that $Q(s_h(D_G)) = \emptyset$ follows from the fact that, for every edge $\{u, v\} \in E_G$, $\{h(u), h(v)\} \in E_H$ cannot be the self-loop on node 1, since it is not in H . This implies that for every $\{u, v\} \in E_G$, it is not possible that the atoms $V(u, 1), V(v, 1)$ coexist in $s_h(D_G)$, which in turn implies that $Q(s_h(D_G)) = \emptyset$, as needed.

The mapping μ is injective. Assume that there are two distinct homomorphisms $h, h' \in \text{hom}(G, H)$ such that $\mu(h) = \mu(h')$. By the definition of μ , we get that $h(u) = h'(u)$, for every node $u \in V_G$. But this contradicts the fact that h and h' are different homomorphisms of $\text{hom}(G, H)$. Therefore, for every two distinct homomorphisms $h, h' \in \text{hom}(G, H)$, $\mu(h) \neq \mu(h')$, as needed.

The mapping μ is surjective. Consider an arbitrary candidate repair $D \in \text{COPrep}(D_G, \Sigma, \neg Q)$. We need to show that there exists $h \in \text{hom}(G, H)$ such that $\mu(h) = D$. We define the mapping $h_D : V_G \rightarrow V_H$ as follows: for every $u \in V_G$:

$$h_D(u) = \begin{cases} 1 & \text{if } V(u, 1) \in D \text{ and } V(u, 0) \notin D \\ 0 & \text{if } V(u, 1) \notin D \text{ and } V(u, 0) \in D \\ ? & \text{if } V(u, 1) \notin D \text{ and } V(u, 0) \notin D \end{cases}$$

It is clear that h_D is well-defined: for every $u \in V_G$, $h_D(u) = x$ and $h_D(u) = y$ implies $x = y$. It is also clear that $\mu(h_D) = D$. It remains to show that $h_D \in \text{hom}(G, H)$. Consider an arbitrary edge $\{u, v\} \in E_G$. By contradiction, assume that $\{h_D(u), h_D(v)\} \notin E_H$. This implies that $h_D(u) = 1$ and $h_D(v) = 1$. Therefore, D contains both atoms $V(u, 1)$ and $V(v, 1)$, which in turn implies that $Q(D) \neq \emptyset$, which contradicts the fact that $D \in \text{COPrep}(D_G, \Sigma, \neg Q)$. \square

Since $\text{HOM}(G) = |\text{COPrep}(D_G, \Sigma, \neg Q)|$, Lemma B.1 implies

$$\text{HOM}(G) = |\text{hom}(G, H)|,$$

which shows that indeed HOM is a polynomial-time Turing reduction from $\#H$ -Coloring to $\text{RRFreq}(\Sigma, Q)$.

B.2 Proof of Item (2) of Theorem 5.1

We prove that, for a set Σ of primary keys, and a CQ Q , the problem $\text{RRFreq}(\Sigma, Q)$ admits an FPRAS. Our proof consists of two main steps, which we briefly explain before going into the detailed proofs.

The first step is to show that, given a database D , we can sample elements of $\text{COPrep}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$. The existence of such an efficient sampler implies that we can employ Monte Carlo Sampling to obtain a *polynomial-time randomized approximation with additive (or absolute) error* for $\text{RRFreq}(\Sigma, Q(\bar{x}))$, that is, a randomized algorithm A that takes a input a database D , a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$, $\epsilon > 0$, and $0 < \delta < 1$ runs

in polynomial time in $\|D\|$, $\|\bar{c}\|$, $1/\epsilon$ and $\log(1/\delta)$, and produces a random variable $A(D, \bar{c}, \epsilon, \delta)$ such that

$$\Pr\left(|A(D, \bar{c}, \epsilon, \delta) - \text{rrfreq}_{\Sigma, Q}(D, \bar{c})| \leq \epsilon\right) \geq 1 - \delta.$$

More precisely, $A(D, \bar{c}, \epsilon, \delta)$ samples $N = O\left(\frac{\log(\frac{1}{\delta})}{\epsilon^2}\right)$ elements of the set $\text{COPrep}(D, \Sigma)$, and returns the number $\frac{S}{N} \cdot |\text{COPrep}(D, \Sigma)|$, where S is the number of sampled repairs D' such that $\bar{c} \in Q(D')$.

However, in general, the existence of an efficient sampler does not guarantee the existence of an FPRAS, which bounds the *multiplicative (or relative) error*. In order to obtain an FPRAS via Monte Carlo Sampling, the number of samples should be proportional to $\frac{1}{\text{rrfreq}_{\Sigma, Q}(D, \bar{c})}$ [8]. This brings us to the second step of our proof, where we show that the ratio $\text{rrfreq}_{\Sigma, Q}(D, \bar{c})$ is never “too small”. Formally, we show that, for every database D , it either holds that $\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) = 0$ or $\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) \geq \frac{1}{\text{pol}(\|D\|)}$ for some polynomial pol . In this case, we can use Monte Carlo Sampling (with a different, yet polynomial, number of samples) to obtain an FPRAS.

We now proceed to formally show the existence of an efficient sample, and the fact that target ratio is never “too small”

Step 1: Efficient Sampler. The formal statement, already given in the main body of the paper, and its proof follow:

LEMMA 5.2. *Given a database D , and a set Σ of primary keys, we can sample elements of $\text{COPrep}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

PROOF. For every relation name R of the underlying schema with a primary key $R : X \rightarrow Y$ in Σ , we partition the set of facts of D over R into blocks of facts that agree on the values of all the attributes of X . Clearly, two facts that belong to the same block, always jointly violate the key of the corresponding relation; hence, an operational repair will contain, for every block B with $|B| > 1$, either a single fact of B or none of the facts of B (hence, there are $|B| + 1$ possible options). An operational repair of the first type can be obtained, for example, via a sequence that removes the facts of B one by one until there is one fact left. An operational repair of the second type can be obtained, for example, by removing the facts of B one by one until there are two facts left, and then removing the last two facts together. For a block B such that $|B| = 1$, there is no justified operation that removes the single fact of B ; hence, this fact will appear in every operational repair.

We denote all the blocks of D (over all the relations of the schema) that have at least two facts by B_1, \dots, B_n . To sample a repair of $\text{COPrep}(D, \Sigma)$, we select, for each block B_i , one of its $|B_i| + 1$ possible outcomes, with probability $\frac{1}{|B_i| + 1}$. Then, an operational repair is obtained by taking the union of all the selections, as well as all the facts of D over every relation R of the schema that has no primary key in Σ , and all the facts that belong to blocks consisting of a single fact. It is rather straightforward that the probability of obtaining each operational repair is $\frac{1}{|\text{COPrep}(D, \Sigma)|}$, as the number of operational repairs is

$$|\text{COPrep}(D, \Sigma)| = (|B_1| + 1) \times \dots \times (|B_n| + 1),$$

and the claim follows. \square

	A_1	A_2
$f_{1,1}$	a_1	b_1
$f_{1,2}$	a_1	b_2
$f_{1,3}$	a_1	b_3
$f_{2,1}$	a_2	b_1
$f_{3,1}$	a_3	b_1
$f_{3,2}$	a_3	b_2

Figure 2: A database over $\{R/2\}$ that is inconsistent w.r.t. the primary key $R : A_1 \rightarrow A_2$.

We give a simple example that illustrates the proof of Lemma 5.2.

Example B.2. Consider the database depicted in Figure 2 over the schema $\{R/2\}$, with (A_1, A_2) being the tuples of attributes of R , and the set $\Sigma = \{R : A_1 \rightarrow A_2\}$ consisting of a single key. We write $f_{i,j}$ for $R(a_i, b_j)$. Clearly, for $j \neq k$, $\{f_{i,j}, f_{i,k}\} \not\models \Sigma$. The database consists of three blocks w.r.t. $R : A_1 \rightarrow A_2$:

$$\{f_{1,1}, f_{1,2}, f_{1,3}\} \quad \{f_{2,1}\} \quad \{f_{3,1}, f_{3,2}\}$$

Since the fact $f_{2,1}$ is not involved in any violations of the constraints, it will appear in every operational repair; however, every operational repair will contain at most one fact of the first block and at most one fact of the third block. The number of operational repairs according to the formula in the proof of Lemma 5.2 is then $(3 + 1) \times (2 + 1) = 12$. (Note that the blocks of size one are not considered in the computation.) Indeed, there are twelve repairs:

$$\begin{aligned} &\{f_{2,1}\} \quad \{f_{1,1}, f_{2,1}\} \quad \{f_{1,2}, f_{2,1}\} \quad \{f_{1,3}, f_{2,1}\} \\ &\{f_{2,1}, f_{3,1}\} \quad \{f_{1,1}, f_{2,1}, f_{3,1}\} \quad \{f_{1,2}, f_{2,1}, f_{3,1}\} \quad \{f_{1,3}, f_{2,1}, f_{3,1}\} \\ &\{f_{2,1}, f_{3,2}\} \quad \{f_{1,1}, f_{2,1}, f_{3,2}\} \quad \{f_{1,2}, f_{2,1}, f_{3,2}\} \quad \{f_{1,3}, f_{2,1}, f_{3,2}\} \end{aligned}$$

The repair $\{f_{1,1}, f_{2,1}, f_{3,1}\}$, for example, is obtained by keeping the fact $f_{1,1}$ of the first block with probability $\frac{1}{4}$ (as there are three facts in the block, there are four possible options: (1) keep $f_{1,1}$, (2) keep $f_{1,2}$, (3) keep $f_{1,3}$, or (4) remove all the facts of the block), and the fact $f_{3,1}$ of the third block with probability $\frac{1}{3}$. Hence, the probability of selecting this operational repair is $\frac{1}{4} \times \frac{1}{3} = \frac{1}{12}$, and the same holds for any other operational repair. ■

Step 2: Polynomial Lower Bound. Now that we have an efficient sampler for the operational repairs, we proceed to show that there is a polynomial lower bound on $\text{rrfreq}_{\Sigma, Q}(D, \bar{c})$. The formal statement, already given in the main body of the paper, and its proof follow:

LEMMA 5.3. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot ||D||)^{||Q||}}$$

whenever $\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) > 0$.

PROOF. By abuse of notation, we treat the CQ Q as the set of atoms on the right-hand side of :- (hence, $|Q|$ is the number of atoms occurring in Q). Consider a database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$. If there is no homomorphism h from Q to D such that $h(Q) \models \Sigma$ and $h(\bar{x}) = \bar{c}$, then it clearly holds that $\text{rrfreq}_{\Sigma, Q}(D, \bar{c}) = 0$.

Consider now the case that such a homomorphism h exists. Assuming that $Q = \{R_i(\bar{y}_i) \mid i \in [n]\}$, let $h(Q) = \{R_i(h(\bar{y}_i)) \mid i \in$

$[n]\}$. Assume that $|h(Q)| = m$ for some $m \leq |Q|$. For every relation name R of the schema with a key $R : X \rightarrow Y$ in Σ , we partition the set of facts of D over R into blocks of facts that agree on the values of all the attributes of X . For a relation name R with no key in Σ , we assume that every fact is a separate block. Let B_1, \dots, B_n be the blocks of D w.r.t. Σ (over all the relation names of the schema). We assume, without loss of generality, that the facts of $h(Q)$ belong to the blocks B_1, \dots, B_m . Clearly, no two facts of $h(Q)$ belong to the same block; otherwise, $h(Q) \not\models \Sigma$, which is a contradiction.

Let $R_{D, \Sigma, h(Q)}^{\text{ne}}$ be the set of repairs $E \in \text{CORep}(D, \Sigma)$ such that $E \cap B_j \neq \emptyset$ for every $j \in [m]$. Let $R_{D, \Sigma, h(Q)}^e$ be the set of repairs $E \in \text{CORep}(D, \Sigma)$ such that $E \cap B_j = \emptyset$ for some $j \in [m]$. Clearly,

$$|\text{CORep}(D, \Sigma)| = \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right| + \left| R_{D, \Sigma, h(Q)}^e \right|.$$

Now, consider a repair E of $R_{D, \Sigma, h(Q)}^e$, and assume that E is disjoint with precisely ℓ blocks of $\{B_1, \dots, B_m\}$. Assume, without loss of generality, that these are the blocks B_1, \dots, B_ℓ . We can transform the repair E into a repair $E' \in R_{D, \Sigma, h(Q)}^{\text{ne}}$ by bringing back an arbitrary fact of each block B_j for $j \in [\ell]$. Therefore, the repair E is mapped to $|B_1| \times \dots \times |B_\ell|$ distinct repairs of $R_{D, \Sigma, h(Q)}^{\text{ne}}$.

Observe that at most $2^m - 1$ repairs $E \in R_{D, \Sigma, h(Q)}^e$ are mapped to the same repair $E' \in R_{D, \Sigma, h(Q)}^{\text{ne}}$. This holds since the repair E' determines, for every block B_j that is not one of B_1, \dots, B_m , whether we keep a fact of B_j in the repair and which fact of B_j we keep. For the blocks B_1, \dots, B_m , a repair E that is mapped to E' can either contain the same fact as E' contains from this block, or none of the facts of the block. Hence, there are two possibilities for each block of $\{B_1, \dots, B_m\}$ and the total number of possibilities is 2^m . However, we have to disregard one of these possibilities, as it represents E' itself (where for every block of $\{B_1, \dots, B_m\}$ we keep the same fact as E'). We conclude that

$$\left| R_{D, \Sigma, h(Q)}^e \right| \leq (2^m - 1) \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|$$

and

$$\begin{aligned} |\text{CORep}(D, \Sigma)| &= \left| R_{D, \Sigma, h(Q)}^e \right| + \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right| \\ &\leq (2^m - 1) \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right| + \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right| \\ &= 2^m \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|. \end{aligned}$$

Note that $(2^m - 1) \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|$ is just an upper bound on $\left| R_{D, \Sigma, h(Q)}^e \right|$ because, as said above, each repair E of $R_{D, \Sigma, h(Q)}^e$ is mapped to several distinct repairs of $R_{D, \Sigma, h(Q)}^{\text{ne}}$.

Finally, each repair of $R_{D, \Sigma, h(Q)}^{\text{ne}}$ keeps a fact of every block in $\{B_1, \dots, B_m\}$. Here, we are interested in the repairs that keep all the facts of $h(Q)$, as these repairs E satisfy $\bar{c} \in Q(E)$. Clearly,

$$|\{E \in \text{CORep}(D, \Sigma) \mid h(Q) \subseteq E\}| = \frac{1}{|B_1| \times \dots \times |B_m|} \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|$$

as all the facts of a block are symmetric. Hence, we conclude that:

$$\frac{|\{E \in \text{CORep}(D, \Sigma) \mid h(Q) \subseteq E\}|}{|\text{CORep}(D, \Sigma)|} \geq \frac{\frac{1}{|B_1| \times \dots \times |B_m|} \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|}{2^m \times \left| R_{D, \Sigma, h(Q)}^{\text{ne}} \right|}$$

$$\begin{aligned}
&= \frac{1}{|B_1| \times \dots \times |B_m| \times 2^m} \\
&\geq \frac{1}{|D|^m \times 2^m} \geq \frac{1}{|D||Q| \times 2|Q|} \\
&= \frac{1}{(2|D|)|Q|} \geq \frac{1}{(2|D|)|Q|}
\end{aligned}$$

and this is clearly a lower bound on $\text{rrfreq}_{\Sigma, Q}(D, \bar{c})$, as needed. \square

Here is a simple example that illustrates the argument given in the proof of Lemma 5.3.

Example B.3. Consider again the database D depicted in Figure 2, and the set $\Sigma = \{R : A_1 \rightarrow A_2\}$ consisting of a single key. Let Q be the CQ $\text{Ans}(x) :- R(a_1, x)$. A homomorphism h from Q to D with $h(Q) \models \Sigma$ and $h(x) = b_1$ is such that $h(Q) = \{R(a_1, b_1)\}$. The fact $R(a_1, b_1)$ belongs to the block $\{f_{1,1}, f_{1,2}, f_{1,3}\}$. Hence, the set $R_{D, \Sigma, h(Q)}^{\text{ne}}$ consists of the repairs:

$$\begin{aligned}
&\{f_{1,1}, f_{2,1}\} \quad \{f_{1,2}, f_{2,1}\} \quad \{f_{1,3}, f_{2,1}\} \\
&\{f_{1,1}, f_{2,1}, f_{3,1}\} \quad \{f_{1,2}, f_{2,1}, f_{3,1}\} \quad \{f_{1,3}, f_{2,1}, f_{3,1}\} \\
&\{f_{1,1}, f_{2,1}, f_{3,2}\} \quad \{f_{1,2}, f_{2,1}, f_{3,2}\} \quad \{f_{1,3}, f_{2,1}, f_{3,2}\}
\end{aligned}$$

and the set $R_{D, \Sigma, h(Q)}^{\text{e}}$ consists of the repairs:

$$\{f_{2,1}\} \quad \{f_{2,1}, f_{3,1}\} \quad \{f_{2,1}, f_{3,2}\}$$

According to the mapping defined in the proof of Lemma 5.3, the repair $\{f_{2,1}\}$ is mapped to the repairs in

$$\{\{f_{1,1}, f_{2,1}\}, \{f_{1,2}, f_{2,1}\}, \{f_{1,3}, f_{2,1}\}\}$$

that have one additional fact from the block of $R(a_1, b_1)$. Similarly, the repair $\{f_{2,1}, f_{3,1}\}$ is mapped to the repairs in

$$\{\{f_{1,1}, f_{2,1}, f_{3,1}\}, \{f_{1,2}, f_{2,1}, f_{3,1}\}, \{f_{1,3}, f_{2,1}, f_{3,1}\}\}$$

and the repair $\{f_{2,1}, f_{3,2}\}$ is mapped to the repairs in

$$\{\{f_{1,1}, f_{2,1}, f_{3,2}\}, \{f_{1,2}, f_{2,1}, f_{3,2}\}, \{f_{1,3}, f_{2,1}, f_{3,2}\}\}.$$

Hence, each repair of $R_{D, \Sigma, h(Q)}^{\text{e}}$ is mapped to precisely three repairs of $R_{D, \Sigma, h(Q)}^{\text{ne}}$, since three is the size of the block of $R(a_1, b_1)$. Moreover, in this case, $2^m - 1 = 1$, and a single repair of $R_{D, \Sigma, h(Q)}^{\text{e}}$ is mapped to every repair of $R_{D, \Sigma, h(Q)}^{\text{ne}}$.

Since all the facts of a block are symmetric with each other, precisely $\frac{1}{3}$ of the repairs in $R_{D, \Sigma, h(Q)}^{\text{ne}}$ contain the fact $R(a_1, b_1)$ —three repairs. Thus, it holds that

$$|\{E \in \text{CORep}(D, \Sigma) \mid h(Q) \subseteq E\}| = \frac{1}{3} \times 9 = 3$$

and

$$|\text{CORep}(D, \Sigma)| = 12.$$

We conclude that

$$\frac{|\{E \in \text{CORep}(D, \Sigma) \mid h(Q) \subseteq E\}|}{|\text{CORep}(D, \Sigma)|} = \frac{3}{12} = \frac{1}{4}.$$

Note that

$$\frac{1}{(2|D|)|Q|} = \frac{1}{12}$$

is indeed a lower bound on that value, and it is also a lower bound on the ratio $\text{rrfreq}_{\Sigma, Q}(D, (b_1))$ that, in this case, equals $\frac{1}{4}$. \blacksquare

B.3 Proof of Item (3) of Theorem 5.1

As discussed in the main body of the paper, the proof of item (3) of Theorem 5.1 proceeds in two main steps, which correspond to Proposition 5.5 and Lemma 5.6. Before giving the formal proofs, we first need to introduce some auxiliary notions and results. In the sequel, we concentrate on undirected graphs without self-loops.

Auxiliary Notions and Results. Consider an undirected graph $G = (V, E)$ and an integer $\Delta \geq 0$. We say that G has *degree* Δ if each node of V participates in at most Δ edges. Moreover, G is connected if there is a path between every two nodes of G . We call G *trivially connected* if $|V| \leq 1$; otherwise, it is *non-trivially connected*. Finally, $\text{IS}(G)$ denotes the set of *all* independent sets of G .

For a database D and a set Σ of FDs, the *conflict graph of D w.r.t. Σ* is the undirected graph $\text{CG}(D, \Sigma) = (V, E)$, where $V = D$, and $\{f, g\} \in E$ if $\{f, g\} \not\models \Sigma$. We call D *non-trivially* (resp., *trivially*) Σ -connected if $\text{CG}(D, \Sigma)$ is non-trivially (resp., trivially) connected.

In order to prove the desired claims, we establish an auxiliary result that relates the number of candidate repairs of an inconsistent database that is non-trivially connected with the number of independent sets of the underlying conflict graph; this is Lemma 5.4 in the main body, which we recall and prove here:

LEMMA 5.4. *Consider a non-trivially Σ -connected database D , where Σ is a set of FDs. It holds that $|\text{CORep}(D, \Sigma)| = |\text{IS}(\text{CG}(D, \Sigma))|$.*

PROOF. (\subseteq) Consider a candidate repair $D' \in \text{CORep}(D, \Sigma)$. By definition, D' is consistent, i.e., there are no two facts $f, g \in D'$ such that $\{f, g\} \not\models \Sigma$. By definition of the conflict graph of D w.r.t. Σ , we conclude that no two facts $f, g \in D'$ are connected via an edge in $\text{CG}(D, \Sigma)$. Hence, D' is an independent set of $\text{CG}(D, \Sigma)$.

(\supseteq) Consider now an independent set $D' \in \text{IS}(\text{CG}(D, \Sigma))$. Since there are no two facts $f, g \in D'$ that are connected via an edge of $\text{CG}(D, \Sigma)$, D' is consistent w.r.t. Σ . It remains to show that there exists a sequence $s \in \text{CRS}(D, \Sigma)$ such that $s(D) = D'$; we distinguish the two cases where either $D' \neq \emptyset$ or $D' = \emptyset$.

Case 1. Let us first concentrate on the case where $D' \neq \emptyset$. In order to define the repairing sequence $s \in \text{CRS}(D, \Sigma)$ such that $s(D) = D'$, we first define a convenient stratification of the facts of D . We inductively define the strata L_0, L_1, \dots as follows:

- $L_0 = D'$.
- For each $i \geq 1$,

$$\begin{aligned}
L_i &= \{f \in D \mid f \notin L_0 \cup \dots \cup L_{i-1} \text{ and} \\
&\quad \text{there is } f' \in L_{i-1} \text{ with } \{f, f'\} \not\models \Sigma\}.
\end{aligned}$$

Observe that, since $\text{CG}(D, \Sigma)$ is connected, each fact $f \in D$ occurs in some L_i , i.e., if n is the smallest integer such that $L_\ell = \emptyset$, for $\ell > n$, we have that $D = \bigcup_{i=0}^n L_i$.

Let $L_i = \{f_1^i, \dots, f_{|L_i|}^i\}$, for each $i \in [n]$. We now construct the desired sequence s as follows. We let the first $|L_n|$ operations be $-f_1^n, \dots, -f_{|L_n|}^n$. To see that $-f_j^n$ is a (D_{j-1}^s, Σ) -justified operation for every $1 \leq j \leq |L_n|$, observe that, by definition of L_n , each f_j^n is in a violation with some fact in L_{n-1} , which has not been removed yet. The operations $op_{|L_n|+1}, \dots, op_{|L_n|+|L_{n-1}|}$ will be $-f_1^{n-1}, \dots, -f_{|L_{n-1}|}^{n-1}$, which are all justified because there exists a

violation for each fact with some fact from L_{n-2} that has not been removed yet. The next operations of s are defined in the same way for the remaining strata, until the last $|L_1|$ operations, which will be $-f_1^1, \dots, -f_{|L_1|}^1$. Again, these operations are justified since, by definition, each of the facts $f_1^1, \dots, f_{|L_1|}^1$ is in conflict with some fact from $L_0 = D'$. Summing up, the sequence s is

$$-f_1^n, \dots, -f_{|L_n|}^n, -f_1^{n-1}, \dots, -f_{|L_{n-1}|}^{n-1}, \dots, -f_1^1, \dots, -f_{|L_1|}^1.$$

We have that $s \in \text{RS}(D, \Sigma)$ and that $s(D) = D' \models \Sigma$. Hence, $s \in \text{CRS}(D, \Sigma)$, which implies $D' \in \text{CORep}(D, \Sigma)$, as needed.

Case 2. The case where $D' = \emptyset$ is treated similarly. We only need to slightly adjust the last operation of the sequence s . Fix some fact $f^* \in D$. We stratify the facts of D as in the first case, but we let $L_0 = \{f^*\}$. We then define s in the same way as above. Let L_n be again the last non-empty stratum. Since $\text{CG}(D, \Sigma)$ is non-trivially connected, $D \not\models \Sigma$, and thus, we have that $n > 0$, i.e., at least stratum L_1 is non-empty. Then, we have that the first $|L_n|$ operations are $-f_1^n, \dots, -f_{|L_n|}^n$. We continue with the remaining strata L_{n-1} to L_2 as before. The last $|L_1|$ operations are defined as $-f_1^1, \dots, -f_{|L_1|-1}^1, -\{f_{|L_1|}^1, f^*\}$. Note that, by definition of L_1 , every fact $f_1^1, \dots, f_{|L_1|-1}^1$ is in a violation with f^* , and thus, their removal is a justified operation. Now, there are only two facts left, $f_{|L_1|}^1$ and f^* , which together violate Σ (recall that L_1 is non-empty). Therefore, $-\{f_{|L_1|}^1, f^*\}$ is a justified operation, and we have that $s \in \text{RS}(D, \Sigma)$ and $s(D) = D' = \emptyset \models \Sigma$. Hence, $s \in \text{CRS}(D, \Sigma)$, which in turn implies that $D' \in \text{CORep}(D, \Sigma)$, as needed. \square

We are now ready to proceed with the two main steps of the proof of item (3) of Theorem 5.1, which correspond to Proposition 5.5 and Lemma 5.6, respectively. Note that both results are essentially dealing with the following counting problem for a set Σ of FDs:

PROBLEM : $\#\text{CORep}^{\text{con}}(\Sigma)$
INPUT : A non-trivially Σ -connected database D .
OUTPUT : The number $|\text{CORep}(D, \Sigma)|$.

Step 1: An Inapproximability Result About Keys. The formal statement, already given in the main body, and its proof follow. Note that the statement of Proposition 5.5 given below is more compact than the one given in the main body of the paper since we explicitly use the name of the problem $\#\text{CORep}^{\text{con}}(\Sigma)$.

PROPOSITION 5.5. *Unless $\text{RP} = \text{NP}$, there exists a set Σ of keys over $\{R\}$ such that $\#\text{CORep}^{\text{con}}(\Sigma)$ does not admit an FPRAS.*

Before giving the proof of Proposition 5.5, we need an auxiliary result about the problem of counting the number of independent sets of undirected graphs. For an integer $\Delta \geq 0$, we define

PROBLEM : $\#\text{IS}_\Delta$
INPUT : An undirected graph G of degree Δ .
OUTPUT : The number $|\text{IS}(G)|$.

We know from [22] that the following holds:

PROPOSITION B.4. *For every $\Delta \geq 6$, unless $\text{RP} = \text{NP}$, $\#\text{IS}_\Delta$ does not admit an FPRAS.*

Note that the above result states the inapproximability of $\#\text{IS}_\Delta$ for arbitrary, not necessarily non-trivially connected graphs. However, for showing Proposition 5.5, we need the stronger version of Proposition B.4 that establishes the inapproximability of $\#\text{IS}_\Delta$ even for non-trivially connected graphs. Let $\#\text{IS}_\Delta^{\text{con}}$ be the problem defined as $\#\text{IS}_\Delta$ with the difference that the input is a non-trivially connected undirected graph. We proceed to show the following:

LEMMA B.5. *For every $\Delta \geq 6$, unless $\text{RP} = \text{NP}$, $\#\text{IS}_\Delta^{\text{con}}$ does not admit an FPRAS.*

PROOF. By contradiction, assume that $\#\text{IS}_\Delta^{\text{con}}$ admits an FPRAS, for some $\Delta \geq 6$, i.e., there is a randomized algorithm A that takes as input a non-trivially connected graph $G = (V, E)$ of degree Δ , $\epsilon > 0$, and $0 < \delta < 1$, runs in polynomial time in $\|G\|$, $1/\epsilon$, and $\log(1/\delta)$, and produces a random variable $A(G, \epsilon, \delta)$ such that

$$\Pr((1 - \epsilon) \cdot |\text{IS}(G)| \leq A(G, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{IS}(G)|) \geq 1 - \delta.$$

From this, we can construct an FPRAS A' for $\#\text{IS}_\Delta$ as follows. Given a graph $G = (V, E)$ of degree Δ , let the connected components, i.e., the maximal connected subgraphs, of G be $(\text{CC}_i)_{1 \leq i \leq n}$ with $\text{CC}_i = (V_i, E_i)$. Furthermore, assume, w.l.o.g., that $\text{CC}_1, \dots, \text{CC}_\ell$, are all the trivially connected components of G , for some $\ell \leq n$. Given G , $\epsilon > 0$, and $0 < \delta < 1$, A' is defined as

$$A'(G, \epsilon, \delta) = 2^\ell \cdot \prod_{i=\ell+1}^n A\left(\text{CC}_i, \frac{\epsilon}{2n}, \frac{\delta}{2n}\right).$$

Note that a run of A does not depend on any other run, and thus, the random variables $A(\text{CC}_i, \frac{\epsilon}{2n}, \frac{\delta}{2n})$ are independent from each other. It is also easy to see that

$$|\text{IS}(G)| = 2^\ell \cdot \prod_{i=\ell+1}^n |\text{IS}(\text{CC}_i)|.$$

Therefore, since A is an FPRAS for $\#\text{IS}_\Delta^{\text{con}}$, we have that

$$\Pr\left(\left(1 - \frac{\epsilon}{2n}\right)^n \cdot |\text{IS}(G)| \leq A'(G, \epsilon, \delta) \leq \left(1 + \frac{\epsilon}{2n}\right)^n \cdot |\text{IS}(G)|\right) \geq \left(1 - \frac{\delta}{2n}\right)^n.$$

Finally, we know (see, e.g., [14]) that the following inequalities hold: for $0 \leq x \leq 1$ and $m \geq 1$,

$$1 - x \leq \left(1 - \frac{x}{2m}\right)^m \quad \text{and} \quad \left(1 + \frac{x}{2m}\right)^m \leq 1 + x.$$

Consequently,

$$\Pr((1 - \epsilon) \cdot |\text{IS}(G)| \leq A'(G, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{IS}(G)|) \geq 1 - \delta.$$

Hence, A' fulfils the probabilistic guarantees required for an FPRAS. To confirm the desired running time of A' , note that there are at most $n = |V|$ connected components of G , which can be computed in polynomial time via any textbook algorithm. Thus, since A is an FPRAS for $\#\text{IS}_\Delta^{\text{con}}$, for each $i \in [n]$, the random variable $A(\text{CC}_i, \frac{\epsilon}{2n}, \frac{\delta}{2n})$ can be computed in polynomial time w.r.t. CC_i , $\frac{|V|}{\epsilon}$, and $\frac{|V|}{\delta}$. Since A' multiplies at most $|V|$ such random variables, A' is an FPRAS for $\#\text{IS}_\Delta$, which contradicts Proposition B.4. \square

With Lemma B.5 in place, we can now prove Proposition 5.5.

PROOF OF PROPOSITION 5.5. Consider a non-trivially connected undirected graph $G = (V, E)$ with degree $\Delta = 6$. Let S be the schema consisting of the single relation name $\{R/\Delta+1\}$ with $(A_1, \dots, A_{\Delta+1})$ being the tuple of attributes of R , and $\Sigma_K = \{\phi_1, \dots, \phi_{\Delta+1}\}$ a set of keys over S , where $\phi_i = R : A_i \rightarrow \text{att}(R)$ for each $i \in [\Delta + 1]$. We show that, unless $\text{RP} = \text{NP}$, we can construct a non-trivially Σ_K -connected database D_G in polynomial time in $\|G\|$ such that $|\text{IS}(G)| = |\text{CORep}(D_G, \Sigma_K)|$. Hence, the existence of an FPRAS for $\#\text{CORep}^{\text{con}}(\Sigma_K)$ would imply the existence of an FPRAS for $\#\text{IS}_{\Delta}^{\text{con}}$, which in turn contradicts Lemma B.5.

The key property that D_G should enjoy is the following: there exists a bijection $\mu : V \rightarrow D_G$ from the set of nodes of G to the facts of D_G such that $(u, v) \in E$ iff $\{\mu(u), \mu(v)\} \not\models \Sigma_K$. The latter immediately implies that $|\text{IS}(G)| = |\text{IS}(\text{CG}(D_G, \Sigma_K))|$, which, together with the fact that G is non-trivially connected, and hence, D_G is non-trivially Σ_K -connected, implies that $|\text{IS}(G)| = |\text{CORep}(D_G, \Sigma_K)|$ by Lemma 5.4. The formal construction of D_G follows.

The Database D_G . It is known that the edges of G are $(\Delta + 1)$ -colourable, and such a coloring can be constructed in polynomial time in $\|G\|$ [20]. Therefore, we are able to efficiently assign the colours $C = \{c_1, \dots, c_{\Delta+1}\}$ to the edges of G in such a way that none of the nodes belongs to two distinct edges of the same colour. Let $M : E \rightarrow C$ be such a coloring. The database D_G is such that $\text{dom}(D_G) = E \cup F$, where F is a finite set of constants with $E \cap F = \emptyset$, and has the following facts:

- (1) for each node $v \in V$, we add to D_G a fact of the form $R(a_1^v, \dots, a_{\Delta+1}^v)$, and
- (2) the constants of such facts are defined as follows:
 - (a) for every edge $e = \{u, v\} \in E$, assuming that $M(e) = c_i$, we let $a_i^u = a_i^v = e$, and
 - (b) for every a_i^v not defined in the above step, we let $a_i^v = f$ for some constant $f \in F$ only to be used once.

We use $R(\bar{a}^v)$ to denote $R(a_1^v, \dots, a_{\Delta+1}^v)$, for short. Note that every a_i^v is well-defined since v has at most one edge of colour c_i , and thus, a_i^v is uniquely defined by either the edge with colour c_i , or in case there is no such edge, by a fresh constant $f \in F$. Let us also stress that we can build D_G in polynomial time in $\|G\|$. We can now prove the following crucial property of D_G :

LEMMA B.6. *Consider two nodes u, v of G . Then, $\{u, v\}$ is an edge in G iff $\{R(\bar{a}^u), R(\bar{a}^v)\} \not\models \Sigma_K$.*

PROOF. Consider an edge $e = \{u, v\}$ in G with $M(e) = c_i$, for some $i \in [\Delta + 1]$. By construction of D_G , it is clear that there will be exactly two facts in D_G such that the constant e appears at position i of those facts. These two facts will be precisely $R(a_1^u, \dots, a_{\Delta+1}^u)$ and $R(a_1^v, \dots, a_{\Delta+1}^v)$, having $a_i^u = a_i^v = e$. As there are no multiple edges between two vertices, the constants at the other positions will be pairwise different, i.e., $a_j^u \neq a_j^v$, for all $j \neq i$. Hence, the two facts together violate ϕ_i , and thus, $\{R(\bar{a}^u), R(\bar{a}^v)\} \not\models \Sigma_K$.

Consider now two facts $R(a_1^u, \dots, a_{\Delta+1}^u)$ and $R(a_1^v, \dots, a_{\Delta+1}^v)$ that together violate $\phi_i = R : A_i \rightarrow \text{att}(R)$ for some $i \in [\Delta + 1]$. Thus, the same constant appears at position i in both facts, i.e., $a_i^u = a_i^v$. By construction of D_G , the only reason why $a_i^u = a_i^v$ is because $\{u, v\}$ is an edge of G , and the claim follows. \square

By Lemma B.6, we get that $|\text{IS}(G)| = |\text{IS}(\text{CG}(D_G, \Sigma_K))|$, and that D_G is non-trivially Σ -connected. Hence, by Lemma 5.4, $|\text{IS}(G)| = |\text{IS}(\text{CG}(D_G, \Sigma_K))| = |\text{CORep}(D_G, \Sigma_K)|$, and the claim follows. \square

Step 2: Transferring FPRAS. We proceed with the second and last step of the proof of item (3) of Theorem 5.1, which corresponds to Lemma 5.6. The formal statement, already given in the main body, and its proof follow. Note that the statement of Lemma 5.6 given below is more compact than the one given in the main body since we explicitly use the name of the problem $\#\text{CORep}^{\text{con}}(\Sigma_K)$, where Σ_K is the set of keys provided by Proposition 5.5.

LEMMA 5.6. *Assume that $\text{RRFreq}(\Sigma, Q)$ admits an FPRAS, for every set Σ of FDs and CQ Q . Then, $\#\text{CORep}^{\text{con}}(\Sigma_K)$ admits an FPRAS.*

PROOF. By Proposition 5.5, unless $\text{RP} = \text{NP}$, Σ_K is a set of keys over a schema $\{R/n\}$ such that $\#\text{CORep}^{\text{con}}(\Sigma_K)$ does not admit an FPRAS. Let $S = \{R'/m\}$, where $m = n + 2$, and, assuming that (A_1, \dots, A_n) is the tuple of attributes of R , let (A, B, A_1, \dots, A_n) be the tuple of attributes of R' . We first show that there exist a set Σ_F of FDs over S , and a Boolean CQ Q_F over S such that, for every non-trivially Σ_K -connected database D over $\{R\}$, we can construct in polynomial time in $\|D\|$ a database D_F over S such that

$$\text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) = \frac{1}{|\text{CORep}(D, \Sigma_K)| + 1}. \quad (*)$$

By exploiting the above equation, the fact that D_F can be constructed in polynomial time, and the FPRAS for $\text{RRFreq}(\Sigma_F, Q_F)$ (which exists by hypothesis), we will then explain how to devise an FPRAS for the problem $\#\text{CORep}^{\text{con}}(\Sigma_K)$.

We start by explaining how Σ_F and D_F are defined in a way that

$$|\text{CORep}(D_F, \Sigma_F)| = |\text{CORep}(D, \Sigma_K)| + 1.$$

We define the set Σ_F of FDs over S as follows:

$$\{R' : X \rightarrow Y \mid R : X \rightarrow Y \in \Sigma_K\} \cup \{R' : A \rightarrow B\}.$$

Note that each key ϕ of Σ_K over R becomes an FD ϕ' over R' ; indeed, ϕ' is not a key since R' has two additional attributes. Now, given a non-trivially Σ_K -connected database D over $\{R\}$, we define the database D_F as follows with a, b being constants not in $\text{dom}(D)$:

$$\{R'(a, b, a_1, \dots, a_n) \mid R(a_1, \dots, a_n) \in D\} \cup \{R'(a, a, \dots, a)\}.$$

For brevity, we will write f^* for the fact $R'(a, a, \dots, a)$. It is not difficult to verify that the number $|\text{CORep}(D_F, \Sigma_F)|$ is the sum

$$\begin{aligned} & |\{D' \in \text{CORep}(D_F, \Sigma_F) \mid f^* \in D'\}| + \\ & |\{D' \in \text{CORep}(D_F, \Sigma_F) \mid f^* \notin D'\}|, \end{aligned}$$

that is, the sum of the number of candidate repairs containing f^* and the number of candidate repairs not containing f^* . It is not difficult to see that $\{f^*\}$ is the only candidate repair containing f^* . This is because f^* is in a conflict with every other fact of D_F due to the FD $R' : A \rightarrow B$. Moreover, one can easily devise a sequence $s \in \text{CRS}(D_F, \Sigma_F)$ removing all facts in $D_F \setminus \{f^*\}$ in an arbitrary order, and therefore obtaining $\{f^*\}$.

Regarding the number of candidate repairs not containing f^* , observe that since D is non-trivially Σ_K -connected, and since f^* is in a conflict with every other fact of D_F , then D_F is non-trivially Σ_F -connected. Therefore, by Lemma 5.4, $|\text{CORep}(D_F, \Sigma_F)| = |\text{IS}(\text{CG}(D_F, \Sigma_F))|$. Since $\{f^*\}$ is the only candidate repair of D_F

containing f^* , and thus, the only independent set of $\text{CG}(D_F, \Sigma_F)$ containing f^* , the set of candidate repairs without f^* , i.e., $\{D' \in \text{CORep}(D_F, \Sigma_F) \mid f^* \notin D'\}$ coincides with $\text{IS}(\text{CG}(D_F \setminus \{f^*\}, \Sigma_F))$. Note that, by construction of D_F and Σ_F , since D is non-trivially Σ_K -connected, $D_F \setminus \{f^*\}$ is non-trivially Σ_F -connected, and thus, by Lemma 5.4, $\text{IS}(\text{CG}(D_F \setminus \{f^*\}, \Sigma_F)) = \text{CORep}(D_F \setminus \{f^*\}, \Sigma_F)$.

Finally, by construction of D_F and Σ_F , we have that $|\text{CORep}(D_F \setminus \{f^*\}, \Sigma_F)| = |\text{CORep}(D, \Sigma_K)|$. In fact, it suffices to observe that two facts $R(a_1, \dots, a_n), R(b_1, \dots, b_n) \in D$ violate Σ_K iff the corresponding facts $R'(a, b, a_1, \dots, a_n), R'(a, b, b_1, \dots, b_n) \in D_F \setminus \{f^*\}$ violate Σ_F . Hence, we conclude that

$$|\text{CORep}(D_F, \Sigma_F)| = |\text{CORep}(D, \Sigma_K)| + 1.$$

Let us now define the Boolean CQ Q_F in such a way that the equation (*) holds. We define Q_F as the Boolean CQ

$$\text{Ans}() :- R'(x, x, \dots, x).$$

In simple words, Q_F asks whether there exists a fact such that all the attributes have the same value. Clearly, the only candidate repair of $\text{CORep}(D_F, \Sigma_F)$ that satisfies the query Q_F is $\{f^*\}$, i.e.,

$$\text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) = \frac{1}{|\text{CORep}(D_F, \Sigma_F)|}.$$

Since, as shown above, $|\text{CORep}(D_F, \Sigma_F)| = |\text{CORep}(D, \Sigma_K)| + 1$, we get that the equation (*) holds, as needed.

Building the FPRAS. We proceed to devise an FPRAS for the problem $\#\text{CORep}^{\text{con}}(\Sigma_K)$ by exploiting the equation (*), the fact that D_F can be constructed in polynomial time, and the FPRAS A' for $\text{RRFreq}(\Sigma_F, Q_F)$ (which exists by hypothesis).

Given a non-trivially Σ_K -connected database D , $\epsilon > 0$, and $0 < \delta < 1$, we define A as the following randomized procedure:

- (1) Compute D_F from D ;
- (2) Let $\epsilon' = \frac{\epsilon}{2+\epsilon}$;
- (3) Let $r = \max\left\{\frac{1-\epsilon'}{2 \cdot (1+2^{|D|})}, A'(D_F, (), \epsilon', \delta)\right\}$;
- (4) Output $\frac{1}{r} - 1$.

We proceed to show that A is an FPRAS for $\#\text{CORep}^{\text{con}}(\Sigma_K)$. Since D_F can be constructed in polynomial time in $\|D\|$, $A(D, \epsilon, \delta)$ runs in polynomial time in $\|D\|$, $1/\epsilon$ and $\log(1/\delta)$ by definition. We now discuss the probabilistic guarantees. By assumption,

$$\begin{aligned} \Pr\left((1 - \epsilon') \cdot \text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) \leq A'(D_F, (), \epsilon', \delta)\right) \\ \leq (1 + \epsilon') \cdot \text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) \geq 1 - \delta. \end{aligned}$$

Thus, it suffices to show that the left-hand side of the above inequality is bounded from above by

$$\begin{aligned} \Pr((1 - \epsilon) \cdot |\text{CORep}(D, \Sigma_K)| \leq A(D, \epsilon, \delta) \leq \\ (1 + \epsilon) \cdot |\text{CORep}(D, \Sigma_K)|). \end{aligned}$$

To this end, by equation (*), we get that

$$\begin{aligned} \Pr\left((1 - \epsilon') \cdot \text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) \leq A'(D_F, (), \epsilon', \delta)\right) \\ \leq (1 + \epsilon') \cdot \text{rrfreq}_{\Sigma_F, Q_F}(D_F, ()) = \Pr(E), \end{aligned}$$

where E is the event

$$\frac{1 - \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|} \leq A'(D_F, (), \epsilon', \delta) \leq \frac{1 + \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|}.$$

Note that $|\text{CORep}(D, \Sigma_K)| \leq 2^{|D|}$, i.e., $|\text{CORep}(D, \Sigma_K)|$ is at most the number of all possible subsets of D . Hence,

$$\frac{1 - \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|} \geq \frac{1 - \epsilon'}{1 + 2^{|D|}}.$$

Thus, for E to hold is necessary that the output of $A'(D_F, (), \epsilon', \delta)$ is no smaller than $\frac{1 - \epsilon'}{1 + 2^{|D|}}$. Hence, for any number $p < \frac{1 - \epsilon'}{1 + 2^{|D|}}$, E coincides with the event

$$\begin{aligned} \frac{1 - \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|} \leq \max\{p, A'(D_F, (), \epsilon', \delta)\} \leq \\ \frac{1 + \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|}. \end{aligned}$$

Hence, with $p = \frac{1 - \epsilon'}{2 \cdot (1 + 2^{|D|})} < \frac{1 - \epsilon'}{1 + 2^{|D|}}$, we conclude that

$$\Pr(E) = \Pr\left(\frac{1 - \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|} \leq \max\{p, A'(D_F, (), \epsilon', \delta)\} \leq \frac{1 + \epsilon'}{1 + |\text{CORep}(D, \Sigma_K)|}\right).$$

Since the random variable $\max\{p, A'(D_F, (), \epsilon', \delta)\}$ always outputs a rational strictly larger than 0, the latter probability coincides with

$$\Pr\left(\frac{1 + |\text{CORep}(D, \Sigma_K)|}{1 + \epsilon'} \leq \frac{1}{\max\{p, A'(D_F, (), \epsilon', \delta)\}} \leq \frac{1 + |\text{CORep}(D, \Sigma_K)|}{1 - \epsilon'}\right).$$

For short, let X be the random variable $\frac{1}{\max\{p, A'(D_F, (), \epsilon', \delta)\}}$. Since $\frac{1}{1 - \epsilon'} = 1 + \frac{\epsilon'}{1 - \epsilon'}$ and $\frac{1}{1 + \epsilon'} = 1 - \frac{\epsilon'}{1 + \epsilon'} \geq 1 - \frac{\epsilon'}{1 - \epsilon'}$, the probability above is less or equal than

$$\begin{aligned} \Pr\left(\left(1 - \frac{\epsilon'}{1 - \epsilon'}\right) \cdot (1 + |\text{CORep}(D, \Sigma_K)|) \leq X \leq \right. \\ \left. \left(1 + \frac{\epsilon'}{1 - \epsilon'}\right) \cdot (1 + |\text{CORep}(D, \Sigma_K)|)\right). \end{aligned}$$

If we subtract 1 from all sides of the inequality, then the above probability coincides with

$$\begin{aligned} \Pr\left(\left(1 - \frac{\epsilon'}{1 - \epsilon'}\right) \cdot (1 + |\text{CORep}(D, \Sigma_K)|) - 1 \leq X - 1 \leq \right. \\ \left. \left(1 + \frac{\epsilon'}{1 - \epsilon'}\right) \cdot (1 + |\text{CORep}(D, \Sigma_K)|) - 1\right). \end{aligned}$$

By expanding the products in the above expression, we obtain

$$\begin{aligned} \Pr\left(|\text{CORep}(D, \Sigma_K)| - \frac{\epsilon'}{1 - \epsilon'} - \frac{\epsilon'}{1 - \epsilon'} \cdot |\text{CORep}(D, \Sigma_K)| \leq \right. \\ \left. X - 1 \leq \right. \\ \left. |\text{CORep}(D, \Sigma_K)| + \frac{\epsilon'}{1 - \epsilon'} + \frac{\epsilon'}{1 - \epsilon'} \cdot |\text{CORep}(D, \Sigma_K)|\right). \end{aligned}$$

Finally, since $|\text{CORep}(D, \Sigma_K)| \geq 1$, we have that

$$\frac{\epsilon'}{1 - \epsilon'} \leq \frac{\epsilon'}{1 - \epsilon'} \cdot |\text{CORep}(D, \Sigma_K)|.$$

Thus, the above probability is less or equal than

$$\Pr \left(\left| |\text{CORep}(D, \Sigma_K)| - 2 \cdot \frac{\epsilon'}{1 - \epsilon'} \cdot |\text{CORep}(D, \Sigma_K)| \right| \leq X - 1 \leq |\text{CORep}(D, \Sigma_K)| + 2 \cdot \frac{\epsilon'}{1 - \epsilon'} \cdot |\text{CORep}(D, \Sigma_K)| \right),$$

which coincides with

$$\Pr \left(\left(1 - 2 \cdot \frac{\epsilon'}{1 - \epsilon'} \right) \cdot |\text{CORep}(D, \Sigma_K)| \leq X - 1 \leq \left(1 + 2 \cdot \frac{\epsilon'}{1 - \epsilon'} \right) \cdot |\text{CORep}(D, \Sigma_K)| \right).$$

Recalling that $\epsilon' = \frac{\epsilon}{2 + \epsilon}$, one can verify that $2 \cdot \frac{\epsilon'}{1 - \epsilon'} = \epsilon$. Moreover, $X - 1$ is $A(D, \epsilon, \delta)$. Hence, the above probability coincides with

$$\Pr \left((1 - \epsilon) \cdot |\text{CORep}(D, \Sigma_K)| \leq A(D, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{CORep}(D, \Sigma_K)| \right).$$

Consequently, A is an FPRAS for $\#\text{CORep}^{\text{con}}(\Sigma_K)$, as needed. \square

It is now straightforward to see that from Proposition 5.5 and Lemma 5.6, we can conclude item (3) of Theorem 5.1.

C PROOFS OF SECTION 6

In this section, we prove the main result of Section 6, which we recall here for the sake of readability:

- THEOREM 6.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us}}, Q)$ is $\#\text{P}$ -hard.*
(2) *For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us}}, Q)$ admits an FPRAS.*

As discussed in Section 6, we actually need to prove the above result for the problem $\text{SRFreq}(\Sigma, Q)$.

C.1 Proof of Item (1) of Theorem 6.1

Let Σ and Q be the singleton set of primary keys and the Boolean CQ, respectively, for which $\text{RRFreq}(\Sigma, Q)$ is $\#\text{P}$ -hard; Σ and Q are obtained from the proof of item (1) of Theorem 5.1. We show that also $\text{SRFreq}(\Sigma, Q)$ is $\#\text{P}$ -hard via a polynomial-time Turing reduction from $\#\text{H-Coloring}$, where H is the graph employed in the proof of item (1) of Theorem 5.1. Actually, we can exploit the same construction as in the proof of item (1) of Theorem 5.1. Assuming that, for an undirected graph G , D_G is the database that the construction in the proof of item (1) of Theorem 5.1 builds, we show that

$$\text{rrfreq}_{\Sigma, Q}(D_G, ()) = \text{srfreq}_{\Sigma, Q}(D_G, ()),$$

which implies that the polynomial-time Turing reduction from $\#\text{H-Coloring}$ to $\text{RRFreq}(\Sigma, Q)$ is also a polynomial-time Turing reduction from $\#\text{H-Coloring}$ to $\text{SRFreq}(\Sigma, Q)$. Recall that

$$\text{srfreq}_{\Sigma, Q}(D_G, ()) = \frac{|\{s \in \text{CRS}(D_G, \Sigma) \mid s(D) \models Q\}|}{|\text{CRS}(D_G, \Sigma)|}.$$

By construction of D_G , each candidate repair $D \in \text{CORep}(D_G, \Sigma)$ can be obtained via $|V_G|!$ different complete sequences of $\text{CRS}(D_G, \Sigma)$. Therefore, $\text{srfreq}_{\Sigma, Q}(D_G, ())$ is

$$\frac{|\text{CORep}(D_G, \Sigma, Q)| \cdot |V_G|!}{|\text{CORep}(D_G, \Sigma)| \cdot |V_G|!} = \frac{|\text{CORep}(D_G, \Sigma, Q)|}{|\text{ORep}(D_G, \Sigma)|}.$$

The latter expression is precisely $\text{rrfreq}_{\Sigma, Q}(D_G, ())$, as needed.

C.2 Proof of Item (2) of Theorem 6.1

We prove that, for a set Σ of primary keys, and a CQ Q , the problem $\text{SRFreq}(\Sigma, Q)$ admits an FPRAS. As for item (2) of Theorem 5.1, the proof consists of two steps: (1) existence of an efficient sampler, and (2) provide a polynomial lower bound for $\text{srfreq}_{\Sigma, Q}(D, \bar{c})$.

Step 1: Efficient Sampler. To establish that we can efficiently sample elements of $\text{CRS}(D, \Sigma)$ uniformly at random, we first need to show that the number of complete repairing sequences can be computed in polynomial time in the case of primary keys.

LEMMA C.1. *Consider a set Σ of primary keys. For every database D , $|\text{CRS}(D, \Sigma)|$ can be computed in polynomial time in $\|D\|$.*

PROOF. Consider a database D . As in the proof of Lemma 5.2, let B_1, \dots, B_n be the blocks of D w.r.t. Σ that contain at least two facts. For a block B of size $m \geq 2$ and for $0 \leq i \leq \lfloor \frac{m}{2} \rfloor$, we denote by $S_m^{\text{ne}, i}$ the number of sequences $s \in \text{CRS}(B, \Sigma)$ such that $s(B) \neq \emptyset$ (hence, $s(B)$ contains a single fact) and precisely i of the operations of s are pair removals. In the case where m is an even number, we cannot obtain a non-empty repair with $\frac{m}{2}$ pair removals; hence, we will have that $S_m^{\text{ne}, \frac{m}{2}} = 0$. In any other case, we have that:

$$\begin{aligned} S_m^{\text{ne}, i} &= m \times \left[\binom{m-1}{2i} \times \frac{(2i)!}{2^i \cdot i!} \times (m-i-1)! \right] \\ &= m \times \frac{(m-1)!}{(2i)! \cdot (m-2i-1)!} \times \frac{(2i)!}{2^i \cdot i!} \times (m-i-1)! \\ &= \frac{m! \cdot (m-i-1)!}{2^i \cdot i! \cdot (m-2i-1)!} \end{aligned}$$

where:

- m is the number of ways to select the single fact of $s(B)$,
- $\binom{m-1}{2i}$ is the number of ways to select $2i$ facts out of the remaining $m-1$ facts (these are the facts removed in pairs),
- $\frac{(2i)!}{2^i \cdot i!}$ is the number of ways to split $2i$ facts into i pairs, and
- $(m-i-1)!$ is the number of permutations of the $m-i-1$ operations in the sequence ($m-2i-1$ singleton removals, and i pair removals).

Similarly, we denote by $S_m^{\text{e}, i}$ the number of sequences $s \in \text{CRS}(B, \Sigma)$ such that $s(B) = \emptyset$ and s has precisely i pair removals. As we cannot obtain an empty repair without pair removals, it holds that $S_m^{\text{e}, 0} = 0$. For $i \geq 1$, the following holds:

$$\begin{aligned} S_m^{\text{e}, i} &= \binom{m}{2} \times \left[\binom{m-2}{2i-2} \times \frac{(2i-2)!}{2^{i-1} \cdot (i-1)!} \times (m-i-1)! \right] \\ &= \frac{m!}{2! \cdot (m-2)!} \times \frac{(m-2)!}{(2i-2)! \cdot (m-2i)!} \\ &\quad \times \frac{(2i-2)!}{2^{i-1} \cdot (i-1)!} \times (m-i-1)! \\ &= \frac{m! \cdot (m-i-1)!}{2^i \cdot (i-1)! \cdot (m-2i)!} \end{aligned}$$

where:

- $\binom{m}{2}$ is the number of ways to select the last pair that will be removed in the sequence,

- $\binom{m-2}{2i-2}$ is the number of ways to select $2(i-1)$ facts out of the remaining $m-2$ facts (these are the facts removed in pairs),
- $\frac{(2i-2)!}{2^{i-1} \cdot (i-1)!}$ is the number of ways to split $2(i-1)$ facts into $i-1$ pairs, and
- $(m-i-1)!$ is the number of permutations of the $m-i-1$ operations in the sequence excluding the last pair removal ($m-2i$ singleton removals, and $i-1$ pair removals).

Since there are no conflicts among facts from different blocks, the repairing sequences for different blocks are independent (in the sense that an operation over the facts of one block has no impact on the justified operations over the facts of another block). Hence, every complete repairing sequence $s \in \text{CRS}(D, \Sigma)$ is obtained by interleaving sequences for the individual blocks. We can compute this number of sequences in polynomial time using dynamic programming. We denote by $P_j^{k,i}$ the number of sequences $s \in \text{CRS}(B_1 \cup \dots \cup B_j, \Sigma)$ with precisely i pair removals such that $s(D) \cap B_\ell \neq \emptyset$ for k of the blocks of B_1, \dots, B_j (hence, $0 \leq k \leq j$). For $k < 0$ or $k > j$, we have $P_j^{k,i} = 0$. Then, it holds that:

$$\text{CRS}(D, \Sigma) = \sum_{k=0}^n \sum_{i=0}^{\lfloor \frac{|B_1|}{2} \rfloor + \dots + \lfloor \frac{|B_n|}{2} \rfloor} P_n^{k,i}.$$

Clearly, for every $i \in \left\{0, \dots, \lfloor \frac{|B_1|}{2} \rfloor\right\}$, we have that:

$$\begin{aligned} P_1^{0,i} &= S_{|B_1|}^{e,i} \\ P_1^{1,i} &= S_{|B_1|}^{ne,i}. \end{aligned}$$

For $j > 1$, it holds that:

$$\begin{aligned} P_j^{k,i} &= \sum_{\substack{0 \leq i_1 \leq \lfloor \frac{|B_1|}{2} \rfloor + \dots + \lfloor \frac{|B_{j-1}|}{2} \rfloor \\ 0 \leq i_2 \leq \lfloor \frac{|B_j|}{2} \rfloor \\ i_1 + i_2 = i}} \left[P_{j-1}^{k,i_1} \times S_{|B_j|}^{e,i_2} \times \right. \\ &\quad \frac{(|B_1 \cup \dots \cup B_j| - i_1 - i_2 - k)!}{(|B_1 \cup \dots \cup B_{j-1}| - i_1 - k)! \times (|B_j| - i_2)!} + \\ &\quad \left. P_{j-1}^{k-1,i_1} \times S_{|B_j|}^{ne,i_2} \times \right. \\ &\quad \left. \frac{(|B_1 \cup \dots \cup B_j| - i_1 - i_2 - k)!}{(|B_1 \cup \dots \cup B_{j-1}| - i_1 - k + 1)! \times (|B_j| - i_2 - 1)!} \right], \end{aligned}$$

where the last expression is the number of ways to interleave a sequence of $\text{CRS}(B_1 \cup \dots \cup B_{j-1}, \Sigma)$ that has i_1 pair removals with a sequence of $\text{CRS}(B_j, \Sigma)$ that has i_2 pair removals. Note that if for a block B_ℓ , the sequence s has i pair removals over the facts of B_ℓ and it holds that $s(D) \cap B_\ell \neq \emptyset$, then s contains $|B_\ell| - i - 1$ operations over the facts of B_ℓ (as we keep one of its facts in the repair). If $s(D) \cap B_\ell = \emptyset$, then s contains $|B_\ell| - i$ operations over the facts of B_ℓ . Hence, $|B_1 \cup \dots \cup B_j| - i_1 - i_2 - k$ is the total number of operations in the combined sequence, $|B_1 \cup \dots \cup B_{j-1}| - i_1 - k$ (or $|B_1 \cup \dots \cup B_{j-1}| - i_1 - k + 1$) is the number of operations over the facts of the first $j-1$ blocks, and $|B_j| - i_2$ (or $|B_j| - i_2 - 1$) is the number of operations over the facts of the j th block. \square

We give an example that illustrates the algorithm described in the proof of Lemma C.1.

Example C.2. Consider again the database D depicted in Figure 2, and the set $\Sigma = \{R : A_1 \rightarrow A_2\}$ consisting of a single key. The complete repairing sequences over the facts of the first block (that consists of the facts $f_{1,1}, f_{1,2}, f_{1,3}$) are:

$$\begin{aligned} &-f_{1,1}, -f_{1,2} & -f_{1,1}, -f_{1,3} & -f_{1,1}, -\{f_{1,2}, f_{1,3}\} \\ &-f_{1,2}, -f_{1,1} & -f_{1,2}, -f_{1,3} & -f_{1,2}, -\{f_{1,1}, f_{1,3}\} \\ &-f_{1,3}, -f_{1,1} & -f_{1,3}, -f_{1,2} & -f_{1,3}, -\{f_{1,1}, f_{1,2}\} \\ &-\{f_{1,1}, f_{1,2}\} & -\{f_{1,1}, f_{1,3}\} & -\{f_{1,2}, f_{1,3}\} \end{aligned}$$

There are no repairing sequences over the facts of the second block, as it only contains a single fact $f_{2,1}$. The complete repairing sequences over the facts of the third block (with facts $f_{3,1}, f_{3,2}$) are:

$$-f_{3,1} \quad -f_{3,2} \quad -\{f_{3,1}, f_{3,2}\}$$

Every complete repairing sequence over D is obtained by interleaving the complete repairing sequences over the different blocks. For example, the following is one possible complete repairing sequence:

$$-f_{1,2}, -\{f_{3,1}, f_{3,2}\}, -f_{1,1}$$

and it has one pair removal.

In this case, we have that:

$$\begin{aligned} S_3^{ne,0} &= \frac{3! \cdot (3-0-1)!}{2^0 \cdot 0! \cdot (3-2 \times 0-1)!} = \frac{12}{2} = 6 \\ S_3^{ne,1} &= \frac{3! \cdot (3-1-1)!}{2^1 \cdot 1! \cdot (3-2 \times 1-1)!} = \frac{6}{2} = 3 \\ S_3^{e,0} &= 0 \\ S_3^{e,1} &= \frac{3! \cdot (3-1-1)!}{2^1 \cdot (1-1)! \cdot (3-2 \times 1)!} = \frac{6}{2} = 3 \end{aligned}$$

Indeed, there are 12 repairing sequences over the facts of the first block that contains three facts—six of them have no pair removals, three have a single pair removal and result in a non-empty repair, and three have a single pair removal and result in an empty repair.

For the third block, that has two facts, it holds that:

$$\begin{aligned} S_2^{ne,0} &= \frac{2! \cdot (2-0-1)!}{2^0 \cdot 0! \cdot (2-2 \times 0-1)!} = \frac{2}{1} = 2 \\ S_2^{ne,1} &= 0 \\ S_2^{e,0} &= 0 \\ S_2^{e,1} &= \frac{2! \cdot (2-1-1)!}{2^1 \cdot (1-1)! \cdot (2-2 \times 1)!} = \frac{1}{2} = 1 \end{aligned}$$

Indeed, there are two sequences with no pair removals (that result in non-empty repairs) and a single sequence with one pair removal (that results in an empty repair).

Finally, we denote the block with three facts by B_1 , and the block with two facts by B_2 . We have that:

$$\begin{aligned} P_1^{0,0} &= S_3^{e,0} = 0 & P_1^{1,0} &= S_3^{ne,0} = 6 \\ P_1^{0,1} &= S_3^{e,1} = 3 & P_1^{1,1} &= S_3^{ne,1} = 3 \end{aligned}$$

Next,

$$P_2^{0,0} = P_1^{0,0} \times S_2^{e,0} \times \frac{(5-0)!}{(3-0)! \times (2-0)!} = 0 \times 0 \times 10 = 0$$

Indeed, if s has zero pair removals, then $s(D) \cap B_1 \neq \emptyset$ and $s(D) \cap B_2 \neq \emptyset$; hence, $P_2^{0,k} > 0$ only for $k = 2$. Thus,

$$\begin{aligned} P_2^{1,0} &= P_1^{0,0} \times S_2^{ne,0} \times \frac{(5-1)!}{(3-0)! \times (2-1)!} \\ &\quad + P_1^{1,0} \times S_2^{e,0} \times \frac{(5-1)!}{(3-1)! \times (2-0)!} \\ &= 0 \times 0 \times 4 + 6 \times 0 \times 6 = 0 \end{aligned}$$

And:

$$P_2^{2,0} = P_1^{1,0} \times S_2^{ne,0} \times \frac{(5-2)!}{(3-1)! \times (2-1)!} = 6 \times 2 \times 3 = 36$$

Similarly, we compute:

$$\begin{aligned} P_2^{0,1} &= P_1^{0,0} \times S_2^{e,1} \times \frac{(5-1)!}{(3-0)! \times (2-1)!} \\ &\quad + P_1^{0,1} \times S_2^{e,0} \times \frac{(5-1)!}{(3-1)! \times (2-0)!} \\ &= 0 \times 1 \times 4 + 3 \times 0 \times 6 = 0 \end{aligned}$$

$$\begin{aligned} P_2^{1,1} &= P_1^{1,1} \times S_2^{e,0} \times \frac{(5-2)!}{(3-2)! \times (2-0)!} \\ &\quad + P_1^{0,1} \times S_2^{ne,0} \times \frac{(5-2)!}{(3-1)! \times (2-1)!} \\ &\quad + P_1^{1,0} \times S_2^{e,1} \times \frac{(5-2)!}{(3-1)! \times (2-1)!} \\ &\quad + P_1^{0,0} \times S_2^{ne,1} \times \frac{(5-2)!}{(3-0)! \times (2-2)!} \\ &= 3 \times 0 \times 3 + 3 \times 2 \times 3 + 6 \times 1 \times 3 + 0 \times 0 \times 1 = 36 \end{aligned}$$

$$\begin{aligned} P_2^{2,1} &= P_1^{1,0} \times S_2^{ne,1} \times \frac{(5-3)!}{(3-1)! \times (2-2)!} \\ &\quad + P_1^{1,1} \times S_2^{ne,0} \times \frac{(5-3)!}{(3-2)! \times (2-1)!} \\ &= 6 \times 0 \times 1 + 3 \times 2 \times 2 = 12 \end{aligned}$$

And, finally:

$$P_2^{0,2} = P_1^{0,1} \times S_2^{e,1} \times \frac{(5-2)!}{(3-1)! \times (2-1)!} = 3 \times 1 \times 3 = 9$$

$$\begin{aligned} P_2^{1,2} &= P_1^{1,1} \times S_2^{e,1} \times \frac{(5-3)!}{(3-2)! \times (2-1)!} \\ &\quad + P_1^{0,1} \times S_2^{ne,1} \times \frac{(5-3)!}{(3-1)! \times (2-2)!} \\ &= 3 \times 1 \times 2 + 3 \times 0 \times 1 = 6 \end{aligned}$$

$$P_2^{2,2} = P_1^{1,1} \times S_2^{ne,1} \times \frac{(5-4)!}{(3-2)! \times (2-2)!} = 3 \times 0 \times 1 = 0$$

We conclude that:

$$|\text{CRS}(D, \Sigma)| = 0 + 0 + 36 + 0 + 36 + 12 + 9 + 6 + 0 = 99$$

That is, there are 99 complete repairing sequences of D w.r.t. Σ . ■

Having Lemma C.1 in place, we can establish the existence of an efficient sampler. The formal statement, already given in the main body of the paper, and its proof follow:

Input: A database D and a set Σ of primary keys over a schema S .

Output: $s \in \text{CRS}(D, \Sigma)$ with probability $\frac{1}{|\text{CRS}(D, \Sigma)|}$.

```

1 return Sample( $D, \Sigma, \epsilon$ )
2 Function Sample( $D, \Sigma, s$ ):
3   if  $s(D) \models \Sigma$  then
4     return  $s$ ;
5   else
6     Select a  $(s(D), \Sigma)$ -justified operation  $op$  with
       probability  $\frac{|\text{CRS}(op(s(D)), \Sigma)|}{|\text{CRS}(s(D), \Sigma)|}$ 
7     return Sample( $D, \Sigma, s \cdot op$ )

```

Algorithm 1: An algorithm SampleSeq for sampling elements of $\text{CRS}(D, \Sigma)$ uniformly at random.

LEMMA 6.2. For a database D , and a set Σ of primary keys, we can sample elements of $\text{CRS}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.

PROOF. The algorithm SampleSeq, depicted in Algorithm 1, is a recursive algorithm that returns a sequence $s \in \text{CRS}(D, \Sigma)$ with probability $\frac{1}{|\text{CRS}(D, \Sigma)|}$. The algorithm starts with the empty sequence ϵ , and, at each step, extends the sequence by selecting one of the justified operations at that point. That is, if the current sequence is s , then we select one of the $(s(D), \Sigma)$ -justified operations. The probability of selecting an operation op is:

$$\frac{|\text{CRS}(op(s(D)), \Sigma)|}{|\text{CRS}(s(D), \Sigma)|}.$$

Hence, the probability of returning a sequence $s = op_1, \dots, op_n$ of $\text{CRS}(D, \Sigma)$ is:

$$\begin{aligned} &\frac{|\text{CRS}(op_1(D), \Sigma)|}{|\text{CRS}(\epsilon(D), \Sigma)|} \times \frac{|\text{CRS}(op_2(op_1(D)), \Sigma)|}{|\text{CRS}(op_1(D), \Sigma)|} \times \dots \\ &\times \frac{|\text{CRS}(op_n(\dots D \dots), \Sigma)|}{|\text{CRS}(op_{n-1}(\dots D \dots), \Sigma)|} \\ &= \frac{|\text{CRS}(op_n(\dots D \dots), \Sigma)|}{|\text{CRS}(\epsilon(D), \Sigma)|} = \frac{1}{|\text{CRS}(D, \Sigma)|} \end{aligned}$$

Most of the terms in the product cancel each other, and

$$|\text{CRS}(op_n(\dots D \dots), \Sigma)| = |\text{CRS}(s(D), \Sigma)| = 1$$

since $s(D) \models \Sigma$; hence, there is a single complete repairing sequence for $s(D)$ w.r.t. Σ —the empty sequence.

Since the length of a sequence is bounded by $|D| - 1$, the number of justified operations at each step is polynomial in $\|D\|$ (as this is the number of facts involved in violations of the constraints plus the number of conflicting pairs of facts), and, by Lemma C.1, for a set Σ of primary keys, we can compute $|\text{CRS}(D, \Sigma)|$ in polynomial time in $\|D\|$ for any database D , we get that the total running time of the algorithm is also polynomial in $\|D\|$, as needed. □

Step 2: Polynomial Lower Bound. Now that we have an efficient sampler for the complete repairing sequences, we show that there is a polynomial lower bound on $\text{srfreq}_{\Sigma, Q}(D, \bar{c})$. The formal statement, already given in the main body of the paper, and its proof follow:

LEMMA 6.3. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{srfreq}_{\Sigma, Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot \|D\|)^{\|Q\|}}$$

whenever $\text{srfreq}_{\Sigma, Q}(D, \bar{c}) > 0$.

PROOF. The proof is very similar to the proof of Lemma 5.3, except that here we reason about sequences rather than repairs. Recall that we treat a query Q as the set $\{R_i(\bar{y}_i) \mid i \in [n]\}$ of atoms on the right-hand side of $:-$, and, for a database D and a homomorphism h from Q to D , we denote by $h(Q)$ the set $\{R_i(h(\bar{y}_i)) \mid i \in [n]\}$. Here, we denote by $S_{D, \Sigma, h(Q)}^e$ the set of sequences $s \in \text{CRS}(D, \Sigma)$ such that $s(D) \cap B_j = \emptyset$ for at least one block of $\{B_1, \dots, B_m\}$ (recall that these are the blocks that contains the facts of $h(Q)$), and by $S_{D, \Sigma, h(Q)}^{\text{ne}}$ the set of sequences $s \in \text{CRS}(D, \Sigma)$ such that $s(D) \cap B_j \neq \emptyset$ for every block of $\{B_1, \dots, B_m\}$.

Now, for every sequence $s \in S_{D, \Sigma, h(Q)}^e$, and for every block B_j such that $s(D) \cap B_j = \emptyset$, the last operation of s over the facts of B_j must remove a pair $\{f, g\}$ of facts. We map each sequence $s \in S_{D, \Sigma, h(Q)}^e$ to a sequence $s' \in S_{D, \Sigma, h(Q)}^{\text{ne}}$ by replacing the last operation of s over each such $B_j \in \{B_1, \dots, B_m\}$ with an operation that removes only one of the facts of the pair—either f or g . Hence, if $s(D) \cap B_j = \emptyset$ for precisely ℓ of the blocks of $\{B_1, \dots, B_m\}$, the sequence s is mapped to 2^ℓ distinct sequences of $S_{D, \Sigma, h(Q)}^{\text{ne}}$.

Similarly to the proof of Lemma 5.3, for every sequence $s' \in S_{D, \Sigma, h(Q)}^{\text{ne}}$, there are $2^m - 1$ sequences $s \in S_{D, \Sigma, h(Q)}^e$ that are mapped to it. This is because the sequence s' determines all the operations over the blocks outside $\{B_1, \dots, B_m\}$, and for each block $B_j \in \{B_1, \dots, B_m\}$, it determines all the operations over B_j except for the last one. If $s'(D) \cap B_j = \{f\}$ and the last operation of s' over B_j removes the fact g , then the last operation of s over B_j either also removes g or removes the pair $\{f, g\}$. If the last operation of s' over B_j removes a pair $\{g, h\}$ of facts, then the last operation of s over B_j must also remove the same pair of facts. Hence, there are at most two possible cases for each block of $\{B_1, \dots, B_m\}$ and 2^m possibilities in total. And, again, we have to disregard the possibility that is equivalent to s' itself.

Therefore, we have that:

$$\left| S_{D, \Sigma, h(Q)}^e \right| \leq (2^m - 1) \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right|$$

and

$$\begin{aligned} |\text{CRS}(D, \Sigma)| &= \left| S_{D, \Sigma, h(Q)}^e \right| + \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right| \\ &\leq (2^m - 1) \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right| + \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right| \\ &= 2^m \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right|. \end{aligned}$$

As said above, each sequence s of $S_{D, \Sigma, h(Q)}^e$ can be mapped to 2^ℓ distinct sequences of $S_{D, \Sigma, h(Q)}^{\text{ne}}$, where ℓ is the number of blocks in $\{B_1, \dots, B_m\}$ for which $E \cap B_j = \emptyset$. Moreover, there are sequences $s' \in S_{D, \Sigma, h(Q)}^{\text{ne}}$ such that no sequence $s \in S_{D, \Sigma, h(Q)}^e$ is mapped to s' .

These are the sequences s' where the last operation of s' over every block of $\{B_1, \dots, B_m\}$ is a pair removal (but s' keeps some fact of each B_j). Hence, $(2^m - 1) \times |S_{D, \Sigma, h(Q)}^{\text{ne}}|$ is only an upper bound on $|S_{D, \Sigma, h(Q)}^e|$. Since all the facts of a single block are symmetric,

$$|\{s \in \text{CRS}(D, \Sigma) \mid h(Q) \subseteq s(D)\}| = \frac{1}{|B_1| \times \dots \times |B_m|} \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right|$$

and, we conclude that

$$\begin{aligned} \frac{|\{s \in \text{CRS}(D, \Sigma) \mid h(Q) \subseteq s(D)\}|}{|\text{CRS}(D, \Sigma)|} &\geq \frac{\frac{1}{|B_1| \times \dots \times |B_m|} \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right|}{2^m \times \left| S_{D, \Sigma, h(Q)}^{\text{ne}} \right|} \\ &= \frac{1}{|B_1| \times \dots \times |B_m| \times 2^m} \\ &\geq \frac{1}{|D|^m \times 2^m} \\ &\geq \frac{1}{(2\|D\|)^{\|Q\|}} \geq \frac{1}{(2\|D\|)^{\|Q\|}} \end{aligned}$$

Since all the sequences of $\{s \in \text{CRS}(D, \Sigma) \mid h(Q) \subseteq s(D)\}$ are such that $h(Q) \subseteq s(D)$ and so $\bar{c} \in s(D)$, this concludes our proof. \square

We give an example that illustrates the argument given in the proof of Lemma 6.3.

Example C.3. Consider the database D , the set Σ of keys, the query Q , and the homomorphism h from Example B.3. Recall that $h(Q) = \{R(a_1, b_1)\}$. The set $S_{D, \Sigma, h(Q)}^e$ contains, for example,

$$-f_{1,2}, -f_{3,1}, -\{f_{1,1}, f_{1,3}\}$$

as the resulting database $s(D)$ contains no fact from the block of $R(a_1, b_1)$. According to the mapping defined in the proof of Lemma 6.3, this sequence is mapped to the following two sequences:

$$-f_{1,2}, -f_{3,1}, -f_{1,1}$$

$$-f_{1,2}, -f_{3,1}, -f_{1,3}$$

that replace the last pair removal over the block of $R(a_1, b_1)$ with a singleton removal. In this case, we have that

$$|\{s \in \text{CRS}(D, \Sigma) \mid h(Q) \subseteq s(D)\}| = 24.$$

These are all the sequences obtained by interleaving the following operations over the facts of the first block (that do not remove $f_{1,1}$), with any of the three operations over the facts of the third block:

$$-f_{1,2}, -f_{1,3} \quad -f_{1,3}, -f_{1,2} \quad -\{f_{1,2}, f_{1,3}\}$$

Moreover, as we have seen in Example C.2, we have that

$$|\text{CRS}(D, \Sigma)| = 99$$

Indeed, it holds that

$$\frac{24}{99} \geq \frac{1}{12} = \frac{1}{(2\|D\|)^{\|Q\|}}$$

as claimed. \blacksquare

D PROOFS OF SECTION 7

In this section, we prove the main result of Section 7, which we recall here for the sake of readability:

- THEOREM 7.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ is $\sharp\text{P}$ -hard.*
(2) *For a set Σ of keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ admits an FPRAS.*

D.1 Proof of Item (1) of Theorem 7.1

As we did for item (1) of Theorem 6.1, we reuse the construction underlying the proof of item (1) of Theorem 5.1. In particular, assuming that Σ and Q are the singleton set of primary keys and the Boolean CQ, respectively, for which $\text{RRFreq}(\Sigma, Q)$ is $\sharp\text{P}$ -hard (Σ and Q are extracted from the proof of item (1) of Theorem 5.1), we show that $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ is $\sharp\text{P}$ -hard via a polynomial-time Turing reduction from $\sharp\text{H}$ -Coloring by reusing the construction in the proof of item (1) of Theorem 5.1; H is the same undirected graph employed in that proof. Assuming that, for an undirected graph G , D_G is the database that the construction in the proof of item (1) of Theorem 5.1 builds, we show that

$$\text{rrfreq}_{\Sigma, Q}(D_G, ()) = P_{M_\Sigma^{\text{uo}}, Q}(D_G, ()),$$

which implies that the polynomial-time Turing reduction from $\sharp\text{H}$ -Coloring to $\text{RRFreq}(\Sigma, Q)$ is also a polynomial-time Turing reduction from $\sharp\text{H}$ -Coloring to $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$.

In the proof of item (1) of Theorem 6.1, we have shown that $\text{rrfreq}_{\Sigma, Q}(D_G, ()) = \text{srfreq}_{\Sigma, Q}(D_G, ())$. Thus, it suffices to show

$$\text{srfreq}_{\Sigma, Q}(D_G, ()) = P_{M_\Sigma^{\text{uo}}, Q}(D_G, ()).$$

Let $M_\Sigma^{\text{uo}}(D_G) = (V, E, \mathbf{P})$. Note that each node u of G induces a violation $\{V(u, 0), V(u, 1)\}$ in D_G that can be resolved using one of the following three operations: remove the first, the second, or both facts. Hence, every complete sequence in $\text{CRS}(D_G, \Sigma)$ is of length precisely $|V_G|$, and for every non-leaf node $s \in V$, $|\text{Ops}_s(D_G, \Sigma)| = 3 \cdot (|V_G| - |s|)$. Hence, by Definition A.5, for each $(s, s') \in E$,

$$P(s, s') = \frac{1}{|\text{Ops}_s(D_G, \Sigma)|} = \frac{1}{3 \cdot (|V_G| - |s|)}.$$

We conclude that, with π being the leaf distribution of M_Σ^{uo} , for each $s = op_1, \dots, op_n \in \text{CRS}(D_G, \Sigma)$,

$$\pi(s) = P(s_0, s_1) \cdots P(s_{n-1}, s_n) = \frac{1}{3^{|V_G|} \cdot |V_G|!}.$$

Since each sequence $s \in \text{CRS}(D_G, \Sigma)$ is assigned the same non-zero probability, π is the uniform distribution over $\text{CRS}(D_G, \Sigma)$. The latter implies that $\text{srfreq}_{\Sigma, Q}(D_G, ()) = P_{M_\Sigma^{\text{uo}}, Q}(D_G, ()),$ as needed.

D.2 Proof of Item (2) of Theorem 7.1

We prove that, for a set Σ of keys, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}}, Q)$ admits an FPRAS. As for item (2) of Theorems 5.1 and 6.1, the proof consists of the usual two steps: (1) existence of an efficient sampler, and (2) provide a polynomial lower bound for $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c})$.

Step 1: Efficient Sampler. Given a database D , the definition of M_Σ^{uo} immediately implies the existence of an efficient sampler that returns a sequence $s \in \text{RL}(\Sigma, M_\Sigma^{\text{uo}}(D))$ with probability $\pi(s)$, where π is the leaf distribution of $M_\Sigma^{\text{uo}}(D)$. The algorithm is very similar to

Algorithm 1, except that if the current sequence is s , the probability to select a $(s(D), \Sigma)$ -justified operation is

$$\frac{1}{|\text{Ops}_s(D, \Sigma)|}.$$

Hence, we immediately obtain the following result, already given in the main body of the paper:

LEMMA 7.2. *Given a database D , and a set Σ of keys, we can sample elements of $\text{RL}(M_\Sigma^{\text{uo}}(D))$ according to the leaf distribution of $M_\Sigma^{\text{uo}}(D)$ in polynomial time in $\|D\|$.*

Step 2: Polynomial Lower Bound. The rest of the section is devoted to showing that there is a polynomial lower bound on $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c})$.

PROPOSITION 7.3. *Consider a set Σ of keys, and a CQ $Q(\bar{x})$. There is a polynomial pol such that, for every database D , and $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) \geq \frac{1}{\text{pol}(\|D\|)}$$

whenever $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) > 0$.

As usual, we treat the CQ Q as the set $\{R_i(\bar{y}_i) \mid i \in [n]\}$ of atoms occurring on the right-hand side of $\bar{\cdot}$. Moreover, for a database D and a homomorphism h from Q to D , we write $h(Q)$ for the set $\{R_i(h(\bar{y}_i)) \mid i \in [n]\}$. Clearly, if there is no homomorphism h from Q to D with $h(Q) \models \Sigma$ and $h(\bar{x}) = \bar{c}$, then $P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) = 0$. Assume now that such a homomorphism h exists. We first prove the claim for the case where $|h(Q)| = 1$, and then generalize it to the case where $|h(Q)| = m$ for some $m \in [|Q|]$.

The Case $|h(Q)| = 1$

Let f be the single fact of $h(Q)$, and

$$P_{D, M_\Sigma^{\text{uo}}, Q}(h) = \sum_{D' \in \text{ORep}(D, M_\Sigma^{\text{uo}}) \text{ and } h(Q) \subseteq D'} P_{D, M_\Sigma^{\text{uo}}}(D').$$

Note that since $h(\bar{x}) = \bar{c}$, it holds that

$$P_{M_\Sigma^{\text{uo}}, Q}(D, \bar{c}) \geq P_{D, M_\Sigma^{\text{uo}}, Q}(h).$$

Hence, it suffices to show that there is a polynomial pol such that $P_{D, M_\Sigma^{\text{uo}}, Q}(h) \geq \frac{1}{\text{pol}(\|D\|)}$. Let S_f and $S_{\neg f}$ be the sets of sequences of $\text{RL}(M_\Sigma^{\text{uo}}(D))$ that keep f and remove f , respectively, i.e.,

$$\begin{aligned} S_f &= \{s \in \text{RL}(M_\Sigma^{\text{uo}}(D)) \mid f \in s(D)\} \\ S_{\neg f} &= \{s \in \text{RL}(M_\Sigma^{\text{uo}}(D)) \mid f \notin s(D)\}. \end{aligned}$$

With π being the leaf distribution of $M_\Sigma^{\text{uo}}(D)$,

$$P_{D, M_\Sigma^{\text{uo}}, Q}(h) = \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}},$$

where

$$\Lambda_f = \sum_{s \in S_f} \pi(s) \quad \text{and} \quad \Lambda_{\neg f} = \sum_{s \in S_{\neg f}} \pi(s).$$

Therefore, to get the desired lower bound $\frac{1}{\text{pol}(\|D\|)}$ for $P_{D, M_\Sigma^{\text{uo}}, Q}(h)$, it suffices to show that there exists a polynomial pol' such that $\Lambda_{\neg f} \leq \text{pol}'(\|D\|) \cdot \Lambda_f$. Indeed, in this case we can conclude that

$$P_{D, M_\Sigma^{\text{uo}}, Q}(h) = \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}}$$

$$\begin{aligned} &\geq \frac{\Lambda_f}{\Lambda_f + \text{pol}'(|D|) \cdot \Lambda_f} \\ &= \frac{1}{1 + \text{pol}'(|D|)}, \end{aligned}$$

and the claim follows with $\text{pol}(|D|) = 1 + \text{pol}'(|D|)$.

We proceed to show that a polynomial pol' such that $\Lambda_{-f} \leq \text{pol}'(|D|) \cdot \Lambda_f$ exists. To this end, we establish an involved technical lemma that relates the sequences of S_{-f} with the sequences of S_f ; as usual, we write π for the leaf distribution of $M_{\Sigma}^{\text{uo}}(D)$:

LEMMA D.1. *There exists a function $F : S_{-f} \rightarrow S_f$ such that:*

- (1) *There exists a polynomial pol'' such that, for every $s \in S_{-f}$,*

$$\pi(s) \leq \text{pol}''(|D|) \cdot \pi(F(s)).$$
- (2) *For every $s' \in S_f$, $|\{s \in S_{-f} \mid F(s) = s'\}| \leq 2 \cdot |D| - 1$.*

PROOF. The bulk of the proof is devoted to showing item (1), whereas item (2) is shown via a simple combinatorial argument.

Item (1). Let $s \in \text{RL}(M_{\Sigma}^{\text{uo}}(D))$ be a repairing sequence that removes f , i.e., $s \in S_{-f}$. We transform s into a repairing sequence $s' \in \text{RL}(M_{\Sigma}^{\text{uo}}(D))$ that does not remove f , i.e., $s' \in S_f$, by deleting or replacing the operation that removes f , and adding additional operations at the end of the sequence as follows. Assume that

$$s = \text{op}_1, \text{op}_2, \dots, \text{op}_{i-1}, \text{op}_i, \text{op}_{i+1}, \dots, \text{op}_n$$

where $\text{op}_i = -f$. Then, we define the sequence

$$s' = \text{op}_1, \text{op}_2, \dots, \text{op}_{i-1}, \text{op}_{i+1}, \dots, \text{op}_n, \text{op}'_1, \dots, \text{op}'_{\ell}$$

where $\text{op}'_1, \dots, \text{op}'_{\ell}$ are new operations that we will describe later. If op_i is of the form $-\{f, g\}$, then

$$s' = \text{op}_1, \text{op}_2, \dots, \text{op}_{i-1}, \text{op}_i^*, \text{op}_{i+1}, \dots, \text{op}_n, \text{op}'_1, \dots, \text{op}'_{\ell}$$

where op_i^* is the operation $-g$, i.e., it removes only the fact g .

An important observation here is that since the sequence s removes f , the repair $s(D)$ might contain facts that conflict with f , but at most k such facts, where k is the number of keys in Σ over the relation name of f . This is a property of keys. Indeed, if $s(D)$ contains $k+1$ facts that conflict with f , then it contains two facts g_1, g_2 that violate the same key with f , in which case g_1, g_2 also jointly violate this key and cannot appear in the same repair. Therefore, at the end of the sequence s' we add ℓ new operations (for some $\ell \leq k$) that remove the facts of $s(D)$ that conflict with f , in some arbitrary order. Note that the sequence s' is a valid repairing sequence, as an additional fact (the fact f) cannot invalidate a justified repairing operation, and we can remove the ℓ conflicting facts at the end in any order, as they are all in conflict with f . Here is a simple example illustrating the construction of s' :

Example D.2. Consider again the database D depicted in Figure 2, and the set $\Sigma = \{R : A_1 \rightarrow A_2, R : A_2 \rightarrow A_1\}$ of keys. Consider also the query Q and homomorphism h from Example B.3. Recall that $h(Q) = \{R(a_1, b_1)\}$. The following sequence is a sequence that removes the fact $R(a_1, b_1)$:

$$s_1 = -f_{1,2}, -f_{1,1}, -f_{3,1}$$

Note that $s(D)$ contains the facts $f_{1,3}$ and $f_{2,1}$ that conflict with $f_{1,1}$. This sequence is mapped to the following sequence s' :

$$s'_1 = -f_{1,2}, -f_{3,1}, -f_{1,3}, -f_{2,1}$$

where we delete the operation $-f_{1,1}$ that removes the fact of $h(Q)$, and add, at the end of the sequence, the operations $-f_{1,3}$ and $-f_{2,1}$ that remove the facts of $s(D)$ that conflict with $f_{1,1}$.

As another example, the sequence:

$$s_2 = -f_{3,1}, -\{f_{1,1}, f_{1,2}\}$$

is mapped to the sequence:

$$s'_2 = -f_{3,1}, -f_{1,2}, -f_{2,1}, -f_{1,3}.$$

Here, the pair removal $-\{f_{1,1}, f_{1,2}\}$ is replaced by the singleton removal $-f_{1,2}$, and, at the end of the sequence, we again add two additional operations that remove (in some arbitrary order) the facts $f_{1,3}$ and $f_{2,1}$ that conflict with $f_{1,1}$. ■

Now, according to the definition of M_{Σ}^{uo} , we have that

$$\pi(s) = \frac{1}{N_1} \times \frac{1}{N_2} \times \dots \times \frac{1}{N_{i-1}} \times \frac{1}{N_i} \times \frac{1}{N_{i+1}} \times \dots \times \frac{1}{N_n}$$

where N_j is the total number of (D_{j-1}^s, Σ) -justified repairing operations before applying the operation op_j of the sequence (recall that D_{j-1}^s is the database obtained from D by applying the first $j-1$ operations of s). Hence,

$$\mathbf{P}((\text{op}_1, \dots, \text{op}_{j-1}), (\text{op}_1, \dots, \text{op}_j)) = \frac{1}{N_j}.$$

Then,

$$\begin{aligned} \pi(s') &= \frac{1}{N_1} \times \frac{1}{N_2} \times \dots \times \frac{1}{N_{i-1}} \times \left[\frac{1}{N_i} \right] \times \frac{1}{N'_{i+1}} \times \dots \times \frac{1}{N'_n} \times \\ &\quad \frac{1}{2\ell+1} \times \frac{1}{2(\ell-1)+1} \times \frac{1}{3}. \end{aligned}$$

The probability $\mathbf{P}((\text{op}_1, \dots, \text{op}_{j-1}), (\text{op}_1, \dots, \text{op}_j))$, for $2 \leq j \leq i-1$, is not affected by the decision to remove or keep f at the i th step. The probability $\mathbf{P}((\text{op}_1, \dots, \text{op}_{j-1}), (\text{op}_1, \dots, \text{op}_j))$ for $i+2 \leq j \leq n$, on the other hand, might decrease in the sequence s' compared to the sequence s , because the additional fact f (that is removed by s but not by s') might be involved in violations with the remaining facts of the database and introduce additional justified operations, in which case $N_j \leq N'_j$. Similarly, the probability $\mathbf{P}((\text{op}_1, \dots, \text{op}_{i-1}), (\text{op}_1, \dots, \text{op}_{i-1}, \text{op}_{i+1}))$ (in the case where $\text{op}_i = -f$) or $\mathbf{P}((\text{op}_1, \dots, \text{op}_i^*), (\text{op}_1, \dots, \text{op}_i^*, \text{op}_{i+1}))$ (in the case where $\text{op}_i = -\{f, g\}$) can only decrease compared to the probability $\mathbf{P}((\text{op}_1, \dots, \text{op}_i), (\text{op}_1, \dots, \text{op}_i, \text{op}_{i+1}))$ in s ; hence, $N_{i+1} \leq N'_{i+1}$.

The term $\frac{1}{N_i}$ denotes the probability of op_i^* , and it only appears in the expression if the sequence s removes the fact f jointly with some other fact g (and the operation op_i^* removes g by itself). Since all the (D_{i-1}^s, Σ) -justified operations have the same probability to be selected, the probabilities $\mathbf{P}((\text{op}_1, \dots, \text{op}_{i-1}), (\text{op}_1, \dots, \text{op}_i))$ and $\mathbf{P}((\text{op}_1, \dots, \text{op}_{i-1}), (\text{op}_1, \dots, \text{op}_i^*))$ are the same. Finally, at the end of the sequence, the only remaining conflicts are those involving f . As said above, there are ℓ facts that conflict with f for some $\ell \leq k$ at that point, and each one of them violates a different key with f . Hence, there are $2\ell+1$ justified operations before applying op'_1 (removing one of the ℓ conflicting facts, removing one of these facts jointly with f , or removing f), there are $2(\ell-1)+1$ possible operations before applying op'_2 and so on.

Example D.3. We continue with Example D.2. For the sequence s_1 , we have that

$$\begin{aligned} \pi(s_1) &= \mathbf{P}(\varepsilon, (-f_{1,2})) \times \mathbf{P}((-f_{1,2}), (-f_{1,2}, -f_{1,1})) \\ &\quad \times \mathbf{P}((-f_{1,2}, -f_{1,1}), (-f_{1,2}, -f_{1,1}, -f_{3,1})) = \frac{1}{14} \times \frac{1}{10} \times \frac{1}{5}. \end{aligned}$$

This holds since, at first, all six facts are involved in violations of the keys, and there are eight conflicting pairs; hence, the total number of justified operations is 14. After removing the fact $f_{1,2}$, the number of justified operations reduces to 10, and after removing the fact $f_{1,1}$, this number is 5. Now, for the sequence s'_1 ,

$$\begin{aligned} \pi(s'_1) &= \mathbf{P}(\varepsilon, (-f_{1,2})) \times \mathbf{P}((-f_{1,2}), (-f_{1,2}, -f_{3,1})) \\ &\quad \times \mathbf{P}((-f_{1,2}, -f_{3,1}), (-f_{1,2}, -f_{3,1}, -f_{1,3})) \\ &\quad \times \mathbf{P}((-f_{1,2}, -f_{3,1}, -f_{1,3}), (-f_{1,2}, -f_{3,1}, -f_{1,3}, -f_{2,1})) \\ &= \frac{1}{14} \times \frac{1}{10} \times \frac{1}{5} \times \frac{1}{3}. \end{aligned}$$

Indeed, the probability of applying $-f_{1,2}$ (i.e., $\mathbf{P}(\varepsilon, (-f_{1,2}))$) is the same for both sequences ($\frac{1}{14}$), while the probability of applying the operation $-f_{3,1}$ in s'_1 (i.e., $\mathbf{P}((-f_{1,2}), (-f_{1,2}, -f_{3,1}))$) is smaller than the probability ($\mathbf{P}((-f_{1,2}, -f_{1,1}), (-f_{1,2}, -f_{1,1}, -f_{3,1}))$) of applying this operation in s_1 : $\frac{1}{10}$ compared to $\frac{1}{5}$. Finally, there are $\ell = 2$ facts in $s(D)$ that conflict with $f_{1,1}$ and we have that

$$\begin{aligned} \mathbf{P}((-f_{1,2}, -f_{3,1}), (-f_{1,2}, -f_{3,1}, -f_{1,3})) &= \frac{1}{2 \times 2 + 1} = \frac{1}{5} \\ \mathbf{P}((-f_{1,2}, -f_{3,1}, -f_{1,3}), (-f_{1,2}, -f_{3,1}, -f_{1,3}, -f_{2,1})) \\ &= \frac{1}{2 \times (2 - 1) + 1} = \frac{1}{3}. \end{aligned}$$

As for the sequence s_2 , it holds that

$$\pi(s_2) = \mathbf{P}(\varepsilon, (-f_{3,1})) \times \mathbf{P}((-f_{3,1}), (-f_{3,1}, -\{f_{1,1}, f_{1,2}\})) = \frac{1}{14} \times \frac{1}{10}$$

while for the sequence s'_2 , it holds that

$$\begin{aligned} \pi(s'_2) &= \mathbf{P}(\varepsilon, (-f_{3,1})) \times \mathbf{P}((-f_{3,1}), (-f_{3,1}, -f_{1,2})) \\ &\quad \times \mathbf{P}((-f_{3,1}, -f_{1,2}), (-f_{3,1}, -f_{1,2}, -f_{2,1})) \\ &\quad \times \mathbf{P}((-f_{3,1}, -f_{1,2}, -f_{2,1}), (-f_{3,1}, -f_{1,2}, -f_{2,1}, -f_{1,3})) \\ &= \frac{1}{14} \times \frac{1}{10} \times \frac{1}{5} \times \frac{1}{3}. \end{aligned}$$

Again, the probability of applying the operation $-f_{3,1}$ is the same in s_2 and s'_2 . The probability of applying $-\{f_{1,1}, f_{1,2}\}$ in s_2 is the same as the probability of applying the operation $-f_{1,2}$ in s'_2 , and the probability of the two additional operations is again $\frac{1}{5} \times \frac{1}{3}$. ■

For every $j \in \{i + 1, \dots, n\}$, we denote by r_j the difference between N_j and N'_j (that is, $N'_j = N_j + r_j$). Hence, it holds that

$$\begin{aligned} \pi(s) &= \pi(s') \times \left[\frac{1}{N_i} \right] \times \frac{1}{N_{i+1}} \times \dots \times \frac{1}{N_n} \times (N_{i+1} + r_{i+1}) \times \dots \times \\ &\quad \times (N_n + r_n) \times (2\ell + 1) \times \dots \times 3 \\ &\leq \pi(s') \times \frac{1}{N_{i+1}} \times \dots \times \frac{1}{N_n} \times (N_{i+1} + r_{i+1}) \times \dots \times \\ &\quad \times (N_n + r_n) \times (2\ell + 1) \times \dots \times 3 \end{aligned}$$

Note that here, the term $\frac{1}{N_i}$ only appears if the original sequence s removes f alone, in which case the term $\frac{1}{N_i}$ does not appear in the expression for $\pi(s')$. We will show that

$$\begin{aligned} \frac{1}{N_{i+1}} \times \dots \times \frac{1}{N_n} \times (N_{i+1} + r_{i+1}) \times \dots \times (N_n + r_n) \\ \times (2\ell + 1) \times \dots \times 3 \leq \text{pol}''(|D|) \end{aligned}$$

for some polynomial pol'' , or, equivalently,

$$\begin{aligned} (N_{i+1} + r_{i+1}) \times \dots \times (N_n + r_n) \times (2\ell + 1) \times \dots \times 3 \\ \leq \text{pol}''(|D|) \times N_{i+1} \times \dots \times N_n. \end{aligned}$$

Note that since $\ell \leq k$, and k is a constant (since we are interested in data complexity), $(2\ell + 1) \times \dots \times 3$ is bounded by a constant. From this point, we denote this value by c . Thus, we prove that

$$(N_{i+1} + r_{i+1}) \times \dots \times (N_n + r_n) \times c \leq \text{pol}''(|D|) \times N_{i+1} \times \dots \times N_n.$$

To show the above, we need to reason about the values r_j . For $j \in \{i + 1, \dots, n\}$, let N_j^f be the number of facts in the database that conflict with f after applying all the operations of s' that occur before op_j , and before applying the operation op_j . Moreover, for every $p \in \{1, \dots, k\}$, let n_j^p be the number of facts in the database that violate the p th key jointly with f at that point. Note that $n_j^1 + \dots + n_j^k \geq N_j^f$, as the same fact might violate several distinct keys jointly with f . If $n_j^p \geq 2$, then every fact that violates the p th key jointly with f participates in a violation of the constraints even if f is not present in the database (as all the facts that violate the same key with f also violate this key among themselves). Hence, for each one of these n_j^p facts, the operation that removes this fact is a justified repairing operation regardless of the presence or absence of f in the database, and it is counted as one of the N_j operations that can be applied at that point in the sequence s . The addition of f then adds n_j^p new justified operations (the removal of a pair of facts that includes f and one of the n_j^p conflicting facts).

On the other hand, if $n_j^p = 1$, then the single fact that violates the p th key jointly with f at that point might not participate in any violation once we remove f . In this case, the presence of f implies two additional justified operations in s' compared to s —the removal of this fact by itself and a pair removal that includes f and this fact. If $n_j^p = 0$, then clearly the p th key has no impact on the number of justified repairing operations w.r.t. f at that point. Now, assume, without loss of generality, that for some $1 \leq p_1 < p_2 \leq k$, it holds that $n_j^p \geq 2$ for all $p \leq p_1$, $n_j^p = 1$ for all $p_1 < p \leq p_2$, and $n_j^p = 0$ for all $p > p_2$. It then holds that

$$r_j \leq N_j^f + (p_2 - p_1) + 1$$

(N_j^f operations remove f jointly with one of its conflicting facts, at most $p_2 - p_1$ operations remove a fact that violates the p th key with f if $n_j^p = 1$, and one operation removes f itself.) Moreover,

$$\begin{aligned} N_j &\geq n_j^1 + \dots + n_j^{p_1} + \frac{n_j^1(n_j^1 - 1)}{2} + \dots + \frac{n_j^{p_1}(n_j^{p_1} - 1)}{2} \\ &= \frac{(n_j^1)^2 + \dots + (n_j^{p_1})^2 + n_j^1 + \dots + n_j^{p_1}}{2}. \end{aligned}$$

Because, as already said, for every p with $n_j^p \geq 2$, the n_j^p operations that remove the facts that violate the p th key with f are also justified operations at the j th step in s , and there are $\frac{n_j^p(n_j^{p-1})}{2}$ additional justified operations that remove a pair from these n_j^p facts, as each such pair of facts jointly violates the p th key.

Example D.4. We continue with Example D.3. Let

$$s_3 = -f_{3,1}, -f_{1,1}, -f_{1,2}$$

Before applying the operation $-f_{1,2}$ of s_3 , there are five justified operations:

$$-f_{1,2} \quad -f_{1,3} \quad -f_{3,2} \quad -\{f_{1,2}, f_{1,3}\} \quad -\{f_{1,2}, f_{3,2}\}$$

At this point, the database contains three facts that conflict with $f_{1,1}$. The facts $f_{1,2}$ and $f_{1,3}$ jointly violate with it the key $R : A_1 \rightarrow A_2$, while the fact $f_{2,1}$ jointly violates with it the key $R : A_2 \rightarrow A_1$.

Observe that the operations $-f_{1,2}, -f_{1,3}, -\{f_{1,2}, f_{1,3}\}$ are justified operations at this point, even though the fact $f_{1,1}$ no longer appears in the database, because $f_{1,2}$ conflict with $f_{1,3}$. If we bring $f_{1,1}$ back, we will have two additional justified operations that involve these fact (one for each fact): $-\{f_{1,1}, f_{1,2}\}$ and $-\{f_{1,1}, f_{1,3}\}$.

Contrarily, the fact $f_{2,1}$ is not involved in any violation of the constraints at this point (before applying the operation $-f_{1,2}$ of s_3); hence, removing this fact is not a justified operation. However, if we bring $f_{1,1}$ back, we will have two additional justified operations that involve this fact: $-f_{2,1}$ and $-\{f_{1,1}, f_{2,1}\}$.

Finally, the fact $f_{1,1}$ introduces another justified operation—the removal of this fact by itself ($-f_{1,1}$). Hence, in the sequence s'_3 that s_3 is mapped to

$$s'_3 = -f_{3,1}, -f_{1,2}, -f_{2,1}, -f_{1,3}$$

The number of justified operations before applying the operation $-f_{1,2}$ is ten, while the number of justified operations before applying this operation in s_3 is five. That is,

$$\mathbf{P}((-f_{3,1}, -f_{1,1}), (-f_{3,1}, -f_{1,1}, -f_{1,2})) = \frac{1}{5}$$

and

$$\mathbf{P}((-f_{3,1}), (-f_{3,1}, -f_{1,2})) = \frac{1}{5+5} = \frac{1}{10} \quad \blacksquare$$

According to the Cauchy–Schwarz inequality for n -dimensional euclidean spaces, it holds that

$$\left(\sum_{i=1}^v x_i y_i \right)^2 \leq \left(\sum_{i=1}^v x_i^2 \right) \times \left(\sum_{i=1}^v y_i^2 \right),$$

where $v \geq 1$ is an integer, and x_i, y_i for $i \in [v]$ are real numbers. By defining $y_i = 1$ for every $i \in [v]$, we then obtain that

$$(x_1 + \dots + x_v)^2 \leq v \times (x_1^2 + \dots + x_v^2).$$

Hence, we have that

$$\begin{aligned} N_j &\geq \frac{(n_j^1)^2 + \dots + (n_j^{p_1})^2 + n_j^1 + \dots + n_j^{p_1}}{2} \\ &\geq \frac{\frac{(n_j^1 + \dots + n_j^{p_1})^2}{p_1} + n_j^1 + \dots + n_j^{p_1}}{2} \end{aligned}$$

$$\begin{aligned} &= \frac{(n_j^1 + \dots + n_j^{p_1})^2 + p_1 \times (n_j^1 + \dots + n_j^{p_1})}{2p_1} \\ &\geq \frac{(N_j^f - (p_2 - p_1))^2 + p_1 \times [N_j^f - (p_2 - p_1)]}{2p_1}. \end{aligned}$$

Note that $N_j^f - (p_2 - p_1)$ is a lower bound on $n_j^1 + \dots + n_j^{p_1}$ because for every $p_2 \leq p$, there are no facts that violate the p th key with f , and for $p_1 < p \leq p_2$, there is a single fact that violates the p th key with f ; hence, $n_j^{p_1+1} + \dots + n_j^{p_2} \leq p_2 - p_1$ and $n_j^{p_2+1} + \dots + n_j^k = 0$.

As aforementioned, $n_j^1 + \dots + n_j^k \geq N_j^f$. Therefore,

$$\begin{aligned} n_j^1 + \dots + n_j^{p_1} &\geq N_j^f - (n_j^{p_1+1} + \dots + n_j^{p_2}) - (n_j^{p_2+1} + \dots + n_j^k) \\ &\geq N_j^f - (p_2 - p_1). \end{aligned}$$

We conclude that

$$r_j \leq N_j^f + (p_2 - p_1) + 1$$

and

$$N_j \geq \frac{(N_j^f - (p_2 - p_1))^2 + p_1 \times [N_j^f - (p_2 - p_1)]}{2p_1}.$$

Hence, it holds that

$$N_j \geq \frac{(r_j - 2(p_2 - p_1) - 1)^2 + p_1 \times [r_j - 2(p_2 - p_1) - 1]}{2p_1}.$$

If $r_j \geq 2(p_2 - p_1) + 1$, then $p_1 \times [r_j - 2(p_2 - p_1) - 1] \geq 0$ and

$$N_j \geq \frac{(r_j - 2(p_2 - p_1) - 1)^2}{2p_1}$$

and

$$\begin{aligned} r_j &\leq \sqrt{2p_1 N_j} + 2(p_2 - p_1) + 1 \leq \sqrt{2k N_j} + 2k + k \\ &\leq \sqrt{4k^2 N_j} + 3k\sqrt{N_j} = 5k\sqrt{N_j}. \end{aligned}$$

If $r_j < 2(p_2 - p_1) + 1$, then $r_j \leq 2k + k \leq 5k\sqrt{N_j}$. Thus, in both cases, we have that $r_j \leq 5k\sqrt{N_j}$.

Recall that our goal is to show that

$$(N_{i+1} + r_{i+1}) \times \dots \times (N_n + r_n) \times c \leq \text{pol}''(|D|) \times N_{i+1} \times \dots \times N_n.$$

We have that

$$(N_{i+1} + r_{i+1}) \times \dots \times (N_n + r_n) \leq (N_{i+1} + 5k\sqrt{N_{i+1}}) \times \dots \times (N_n + 5k\sqrt{N_n}).$$

Thus, it suffices to show that

$$(\sqrt{N_{i+1}} + 5k) \times \dots \times (\sqrt{N_n} + 5k) \times c \leq \text{pol}''(|D|) \times \sqrt{N_{i+1}} \times \dots \times \sqrt{N_n}.$$

For brevity, let $x_j = \sqrt{N_j}$. Moreover, we can clearly define $\text{pol}''(|D|)$ as $c \times \text{pol}'''(|D|)$ for some polynomial pol''' , and get rid of the constant c . Therefore, we now show that

$$(x_{i+1} + 5k) \times \dots \times (x_n + 5k) \leq \text{pol}'''(|D|) \times x_{i+1} \times \dots \times x_n$$

for some polynomial pol''' , or, equivalently,

$$\frac{x_{i+1} + 5k}{x_{i+1}} \times \dots \times \frac{x_n + 5k}{x_n} \leq \text{pol}'''(|D|).$$

Note that in the sequence s , there are $n - j + 1$ operations after the operation op_j (including the operation op_j). Since the number of justified operations can only decrease after applying a certain operation, this means that $N_j \geq n - j + 1$. Hence, we have that

$N_{i+1} \geq n-i$, $N_{i+2} \geq n-i-1$, and so on, which implies that $x_{i+1} \geq \sqrt{n-i}$, $x_{i+2} \geq \sqrt{n-i-1}$, etc. Now, an expression of the form $\frac{x+5k}{x}$ increases when the value of x decreases (because $\frac{x+5k}{x} = 1 + \frac{5k}{x}$); hence, we have that

$$\begin{aligned} & \frac{x_{i+1} + 5k}{x_{i+1}} \times \dots \times \frac{x_n + 5k}{x_n} \\ & \leq \frac{\sqrt{n-i} + 5k}{\sqrt{n-i}} \times \frac{\sqrt{n-i-1} + 5k}{\sqrt{n-i-1}} \times \dots \times \frac{1 + 5k}{1} \\ & \leq \frac{\lfloor \sqrt{n-i} \rfloor + 5k}{\lfloor \sqrt{n-i} \rfloor} \times \frac{\lfloor \sqrt{n-i-1} \rfloor + 5k}{\lfloor \sqrt{n-i-1} \rfloor} \times \dots \times \frac{1 + 5k}{1} \end{aligned}$$

Next, for every $m \geq 1$ it holds that

$$\sqrt{m-1} \geq \sqrt{m} - 1$$

and thus,

$$\lfloor \sqrt{m-1} \rfloor \geq \lfloor \sqrt{m} \rfloor - 1$$

We then obtain the following:

$$\begin{aligned} & \frac{\lfloor \sqrt{n-i} \rfloor + 5k}{\lfloor \sqrt{n-i} \rfloor} \times \frac{\lfloor \sqrt{n-i-1} \rfloor + 5k}{\lfloor \sqrt{n-i-1} \rfloor} \times \dots \times \frac{1 + 5k}{1} \\ & \leq \frac{\lfloor \sqrt{n-i} \rfloor + 5k}{\lfloor \sqrt{n-i} \rfloor} \times \frac{\lfloor \sqrt{n-i} \rfloor - 1 + 5k}{\lfloor \sqrt{n-i} \rfloor - 1} \times \dots \times \frac{1 + 5k}{1} \\ & = \frac{(\lfloor \sqrt{n-i} \rfloor + 5k)!}{(\lfloor \sqrt{n-i} \rfloor)! \times (5k)!} = \binom{\lfloor \sqrt{n-i} \rfloor + 5k}{5k} \\ & \leq \left(\frac{e(\lfloor \sqrt{n-i} \rfloor + 5k)}{5k} \right)^{5k} \leq \left(\frac{e(\lfloor \sqrt{n} \rfloor + 5k)}{5k} \right)^{5k} \\ & \leq \left(\frac{e}{5k} \right)^{5k} \times (\sqrt{|D|} + 5k)^{5k} \end{aligned}$$

(Observe that the maximal length n of a sequence is $|D| - 1$.) The claim follows with

$$\text{pol}'''(|D|) = \left(\frac{e}{5k} \right)^{5k} \times (\sqrt{|D|} + 5k)^{5k}.$$

Recall that $c = (2\ell + 1) \times \dots \times 3$, where ℓ is the number of facts that conflict with f and are not removed by the sequence s ; hence, $\ell \leq k$. Therefore, for every sequence s that removes f , there is some sequence s' that does not remove f such that

$$\pi(s) \leq (2k + 1)! \times \text{pol}'''(|D|) \times \pi(s'),$$

and item (1) of Lemma D.1 follows with

$$\text{pol}''(|D|) = (2k + 1)! \times \text{pol}'''(|D|).$$

Item (2). We now show that the function F from sequences that remove f to sequences that do not remove f , maps at most $2|D| - 1$ sequences of the first type to the same sequence of the second type. Given a sequence $s' \in S_f$, we can obtain this sequence either from a sequence $s \in \text{RL}(M_{\Sigma}^{\text{uo}}(D))$ that has one additional operation that removes f , or from a sequence s that removes f jointly with some other fact g , while s' removes the fact g by itself. (Some of the operations at the end of s' might not appear in s , as they remove facts that conflict only with f .) Since the length of the sequence s' is at most $|D| - 1$, there are at most $|D|$ possible ways to insert an additional operation that removes f , and $|D| - 1$ ways to add f to an existing operation. Hence, there are at most $|D| + |D| - 1$

sequences that remove f that are mapped to the sequence s' . Here is an example that illustrates the above combinatorial argument.

Example D.5. We continue with Example D.4. Consider again the sequence s'_3 . Recall that

$$s'_3 = -f_{3,1}, -f_{1,2}, -f_{2,1}, -f_{1,3}$$

This sequence can be obtained from any of the following sequences that have an additional operation that removes $f_{1,1}$:

$$\begin{aligned} & -f_{1,1}, -f_{3,1}, -f_{1,2} \\ & -f_{3,1}, -f_{1,1}, -f_{1,2} \\ & -f_{3,1}, -f_{1,2}, -f_{1,1} \end{aligned}$$

Note that the operations $-f_{2,1}, -f_{1,3}$ do not appear in these sequences, as after removing $f_{1,1}$ they are no longer involved in violations of the constraints.

The sequence s'_3 can also be obtained from the following sequences that replace an operation of s'_3 that removes a single fact with an operation that removes a pair of conflicting facts:

$$\begin{aligned} & -\{f_{1,1}, f_{3,1}\}, -f_{1,2} \\ & -f_{3,1}, -\{f_{1,1}, f_{1,2}\} \end{aligned} \quad \blacksquare$$

This completes the proof of Lemma D.1. \square

Having Lemma D.1 in place, it is now easy to establish the existence of the polynomial pol' such that $\Lambda_{-f} \leq \text{pol}'(|D|) \cdot \Lambda_f$. Indeed, with F and pol'' being the function and the polynomial, respectively, provided by Lemma D.1,

$$\begin{aligned} \Lambda_{-f} &= \sum_{s \in S_{-f}} \pi(s) \leq \sum_{s \in S_{-f}} \text{pol}''(|D|) \cdot \pi(F(s)) \\ &\leq \text{pol}''(|D|) \cdot (2 \cdot |D| - 1) \cdot \sum_{s \in S_f} \pi(s) \\ &= \text{pol}''(|D|) \cdot (2 \cdot |D| - 1) \cdot \Lambda_f, \end{aligned}$$

and the claim follows with $\text{pol}'(|D|) = \text{pol}''(|D|) \cdot (2 \cdot |D| - 1)$.

The Case $|h(Q)| \geq 1$

We now generalize the proof given above for the case $|h(Q)| = 1$ to the case $|h(Q)| = m$ for some $1 \leq m \leq |Q|$. As in the case where $|h(Q)| = 1$, we map sequences that remove at least one of the facts of $h(Q)$ to sequences that keep all these facts, by deleting or replacing every operation that removes a fact of $h(Q)$ and adding a constant number of operations at the end of the sequence that remove all the facts that conflict with some fact of $h(Q)$.

More formally, let $s \in \text{RL}(M_{\Sigma}^{\text{uo}}(D))$ be a repairing sequence that removes r of the facts of $h(Q)$ (for some $1 \leq r \leq m$):

$$s = op_1, \dots, op_{i_1}, \dots, op_{i_2}, \dots, op_{i_r}, \dots, op_n$$

where the operations $op_{i_1}, \dots, op_{i_r}$ remove these r facts. Note that there are no conflicts among the facts of $h(Q)$; hence, it cannot be the case that a single operation removes two of these facts. We transform s into a sequence $s' \in \text{RL}(M_{\Sigma}^{\text{uo}}(D))$ where each operation op_{i_j} that removes a single fact is deleted, and every operation op_{i_j} that removes a pair $\{f, g\}$ of facts where $f \in h(Q)$ and $g \notin h(Q)$, is replaced by the operation $op_{i_j}^*$ that removes only the fact g . At the end of the sequence s' , we add operations op'_1, \dots, op'_r that remove the facts that are in conflict with one of the facts of $h(Q)$ that appears in

$s(D)$. As we have explained before, for each such fact, the sequence s keeps at most k conflicting facts, where k is the maximal number of keys in Σ over the same relation R ; hence, the total number of conflicting facts that s does not remove is bounded by $m \times k$, and this is a bound on the number ℓ of additional operations (that remove these conflicting facts one by one in some arbitrary order). As in the case where $h(Q) = 1$, the probability of applying the additional ℓ operations at the end of the sequence is some constant that we denote by $\frac{1}{c}$. We provide below more details about this constant.

The probability $\mathbf{P}((op_1, \dots, op_{j-1}), (op_1, \dots, op_j))$, for $2 \leq j \leq i_1 - 1$, is not affected by the decision to remove or keep a certain fact at the i_1 th step. However, for $j \geq i_1$, the probability of applying the operation op_j might decrease in the sequence s' compared to the sequence s , because the additional facts of $h(Q)$ (that are removed by s but not by s') might be involved in violations with the remaining facts of the database and introduce additional justified repairing operations at each step. As we have already shown, if the number of (D_{j-1}^s, Σ) -justified operations before applying the operation op_j of s is N_j , then the addition of a fact can increase this number by at most $5k\sqrt{N_j}$. Hence, the addition of at most m facts (the facts of $h(Q)$) can increase this number by at most $5km\sqrt{N_j}$. We again denote by r_j the factor by which the number of operations increases, and we have that $r_j \leq 5km\sqrt{N_j}$.

Now, all the arguments for the case where $|h(Q)| = 1$ apply also in this case, with the only difference being the value of r_j . Therefore, we conclude that

$$\pi(s) \leq \text{pol}''(\|D\|) \times \pi(s')$$

with

$$\text{pol}''(\|D\|) = c \times \left(\frac{e}{5km}\right)^{5km} \times (\sqrt{\|D\|} + 5km)^{5km}.$$

Recall that $\frac{1}{c}$ is the probability of applying the additional operations at the end of the sequence, and r is the number of facts of $h(Q)$ that are removed by the sequence s . We would like to provide a lower bound on this probability (hence, an upper bound on c). Clearly, the lowest probability is obtained when the number of additional operations is the highest (as for each additional operation we need to multiply the probability by a number lower than one) and when the probability of each individual operation is the lowest. As mentioned above, for each one of the r facts of $h(Q)$ that are removed by s , there are at most k facts that conflict with it and are not removed by s . Hence, $r \times k$ is an upper bound on the number of additional operations. Moreover, the lowest probability of each operation is obtained when the number of justified operations at the point of applying it is the highest. When there are ℓ facts in a database D' that are involved in violations of the constraints, an upper bound on the number of (D', Σ) -justified operations (that is obtained when every fact is in conflict with every other fact) is

$$\ell + \frac{\ell(\ell-1)}{2} = \frac{\ell^2 + \ell}{2} = \frac{\ell(\ell+1)}{2} \leq \frac{(\ell+1)^2}{2} \leq (\ell+1)^2.$$

Therefore, we have that

$$\begin{aligned} \frac{1}{c} &\geq \frac{1}{(rk+r+1)^2} \times \frac{1}{(rk+r)^2} \times \frac{1}{(rk+r-1)^2} \times \dots \times \frac{1}{3} \\ &\geq \frac{1}{((rk+r+1)^2)!} \geq \frac{1}{((mk+m+1)^2)!} \end{aligned}$$

and

$$c \leq ((mk+m+1)^2)!$$

(Observe that $rk+r$ is the number of facts involved in violations if each of the r facts of $h(Q)$ that s removes conflicts with k facts of $s(D)$.) Now, it holds that

$$\begin{aligned} &((mk+m+1)^2)! \times \left(\frac{e}{5km}\right)^{5km} \times (\sqrt{\|D\|} + 5km)^{5km} \leq \\ &((\|Q\|\Sigma + |Q| + 1)^2)! \times e^{5|Q|\Sigma} \times (\sqrt{\|D\|} + 5|Q|\Sigma)^{5|Q|\Sigma} \end{aligned}$$

Hence, with

$$\text{pol}''(\|D\|) = ((\|Q\|\Sigma + |Q| + 1)^2)! \times e^{5|Q|\Sigma} \times (\sqrt{\|D\|} + 5|Q|\Sigma)^{5|Q|\Sigma}$$

we have that

$$\pi(s) \leq \text{pol}''(\|D\|) \times \pi(s'),$$

as needed.

Finally, we show that our mapping from sequences that remove at least one of the facts of $h(Q)$ to sequences that do not remove any of these facts maps at most polynomially many sequences of the first type to the same sequence of the second type. Given a sequence s' that does not remove any of the facts of $h(Q)$, we can obtain this sequence from any sequence s that has additional operations that remove some of the facts of $h(Q)$ individually or operations that remove these facts jointly with another fact (while s' removes only one of these facts). The sequence s can remove any number $1 \leq r \leq m$ of facts of $h(Q)$. And, in the case where it removes r of the facts of $h(Q)$, for every $\ell \leq r$ there are $\binom{r}{\ell}$ possible ways to choose a subset of size ℓ of $h(Q)$ of facts that will be removed by themselves (while the remaining $r - \ell$ facts will be removed jointly with another fact). Since the length of the sequence s is at most $|D| - 1$, there are at most $\binom{|D| + \ell - 1}{\ell}$ possible choices for the positions of the additional singleton deletions, and $\binom{|D| - 1}{r - \ell}$ possible choices for the individual fact removals that will become pair removals. Hence, the number of sequences that remove a fact of $h(Q)$ that are mapped to the sequence s' is at most

$$\begin{aligned} &\sum_{r=1}^m \sum_{\ell=0}^r \binom{r}{\ell} \times \binom{|D| + \ell - 1}{\ell} \times \binom{|D| - 1}{r - \ell} \\ &\leq \sum_{r=1}^m \sum_{\ell=0}^r \left(\frac{er}{\ell}\right)^\ell \times \left(\frac{e(|D| + \ell - 1)}{\ell}\right)^\ell \times \left(\frac{e(|D| - 1)}{r - \ell}\right)^{r - \ell} \\ &\leq \sum_{r=1}^{|Q|} \sum_{\ell=0}^{|Q|} (e|Q|)^\ell \times (e(|D| + \ell - 1))^\ell \times (e(|D| - 1))^{|Q| - \ell} \\ &\leq |Q| \times (|Q| + 1) \times (e|Q|)^{|Q|} \times (e(|D| + |Q| - 1))^{|Q|} \times \\ &\quad (e(|D| - 1))^{|Q|} \\ &\leq (e|Q|)^2 \times (e|Q|)^{|Q|} \times (e(|D| + |Q| - 1))^{|Q|} \times (e(|D| - 1))^{|Q|} \\ &= (e|Q|)^{|Q| + 2} \times (e(|D| + |Q| - 1))^{|Q|} \times (e(|D| - 1))^{|Q|}. \end{aligned}$$

This number is clearly polynomial in $\|D\|$. We denote this number by $\text{pol}'(\|D\|)$. Finally, similarly to the case where $|h(Q)| = 1$,

$$\mathbf{P}_{D, M_{\Sigma}^{\text{uo}}, Q}(h) \geq \frac{1}{1 + \text{pol}''(\|D\|) \times \text{pol}'(\|D\|)}.$$

With $\text{pol}(\|D\|) = 1 + \text{pol}''(\|D\|) \times \text{pol}'(\|D\|)$, we obtain that

$$\mathbf{P}_{M_{\Sigma}^{\text{uo}}, Q}(D, \bar{c}) \geq \mathbf{P}_{D, M_{\Sigma}^{\text{uo}}, Q}(h) \geq \frac{1}{\text{pol}(\|D\|)},$$

which concludes our proof.

D.3 The case of Functional Dependencies

Unlike the case of keys, in the case of FDs, there is no polynomial lower bound on the target probability, as we show next. This means that we cannot rely on Monte Carlo Sampling for devising an FPRAS. On the other hand, this does not preclude the existence of an FPRAS in the case of FDs, which remains an open problem.

PROPOSITION D.6. *Consider the FD set $\{R : A_1 \rightarrow A_2\}$ over the schema $\{R/3\}$, and the Boolean CQ $\text{Ans}() :- R(0, 0, 0)$. There exists a family $\{D_n\}_{n \geq 1}$ of databases such that*

$$0 < P_{M_\Sigma^{\text{uo}}, Q}(D_n, ()) \leq \frac{1}{2^{|D_n|-1}}.$$

PROOF. Let D_n be the database that contains the fact $R(0, 0, 0)$ and $n - 1$ additional facts $R(0, 1, i)$ for $i \in \{1, \dots, n - 1\}$. Observe that each fact $R(0, 1, i)$ is in conflict with $R(0, 0, 0)$, but there are no conflicts among two facts $R(0, 1, i)$ and $R(0, 1, j)$ for $i \neq j$. Clearly, it holds that $0 < P_{M_\Sigma^{\text{uo}}, Q}(D, ())$ as the operational repair that keeps the fact $R(0, 0, 0)$ entails Q . We prove by induction on n , the number of facts in the database, that for a database D that contains the fact $R(0, 0, 0)$ and $n - 1$ facts of the form $R(0, 1, i)$, it holds that:

$$P_{M_\Sigma^{\text{uo}}, Q}(D, ()) \leq \frac{1}{2^{n-1}}$$

Base Case. For $n = 1$, $D = \{R(0, 0, 0)\}$ and there are no violations of the FD. In this case, it is rather straightforward to see that

$$P_{M_\Sigma^{\text{uo}}, Q}(D, ()) = \frac{1}{2^{1-1}} = 1.$$

Inductive Step. We assume that the claim holds for $n = 1, \dots, p$ and prove that it holds for $n = p + 1$. Let D be such a database with $p + 1$ facts; that is, D contains the fact $R(0, 0, 0)$ and p facts of the form $R(0, 1, i)$. Whenever we have p facts of the form $R(0, 1, i)$ in the database, there are $1 + 2p$ justified operations: (1) the removal of $R(0, 0, 0)$, (2) the removal of a fact of the form $R(0, 1, i)$, or (3) the removal of a pair $\{R(0, 0, 0), R(0, 1, i)\}$. Only the p operations of type (2) keep the fact $R(0, 0, 0)$ in the database. We denote these operations by op_1, \dots, op_p . For every $i \in \{1, \dots, p\}$, we have that

$$P(\varepsilon, (op_i)) = \frac{1}{1 + 2p}.$$

After removing a fact of the form $R(0, 1, i)$ from the database, we have $p - 1$ such facts left, regardless of which specific fact we remove. For every $i \in \{1, \dots, p\}$, we denote by D_i the database $op_i(D)$. By the inductive hypothesis, we have that

$$P_{M_\Sigma^{\text{uo}}, Q}(D_i, ()) \leq \frac{1}{2^{p-1}}.$$

Every sequence $s \in \text{RL}(M_\Sigma^{\text{uo}}(D))$ with $R(0, 0, 0) \in s(D)$ is of the form $op_i \cdot s_i$ for some $i \in [p]$ and $s_i \in \text{RL}(M_\Sigma^{\text{uo}}(D_i))$ with $R(0, 0, 0) \in s_i(D)$. The probability $P_{M_\Sigma^{\text{uo}}, Q}(D, ())$ can then be written as

$$P_{M_\Sigma^{\text{uo}}, Q}(D, ()) = \sum_{\substack{s \in \text{RL}(M_\Sigma^{\text{uo}}(D)) \\ () \in Q(s(D))}} \pi(s)$$

$$= \sum_{i=1}^p \left(P(\varepsilon, (op_i)) \times \sum_{\substack{s_i \in \text{RL}(M_\Sigma^{\text{uo}}(D_i)) \\ () \in Q(s_i(D_i))}} \pi(s_i) \right).$$

As said above, for every $i \in \{1, \dots, p\}$,

$$P_{M_\Sigma^{\text{uo}}, Q}(D_i, ()) = \sum_{\substack{s_i \in \text{RL}(M_\Sigma^{\text{uo}}(D_i)) \\ () \in Q(s_i(D_i))}} \pi(s_i) \leq \frac{1}{2^{p-1}}.$$

Therefore, we conclude that

$$\begin{aligned} P_{M_\Sigma^{\text{uo}}, Q}(D, ()) &\leq \sum_{i=1}^p \left(\frac{1}{1 + 2p} \times \frac{1}{2^{p-1}} \right) = \frac{p}{(1 + 2p) \times 2^{p-1}} \\ &= \frac{p}{2^{p-1} + p \times 2^p} \leq \frac{p}{p \times 2^p} = \frac{1}{2^p} \end{aligned}$$

and the claim follows. \square

D.4 Proof of Theorem 7.5

We now show that if only singleton removals are allowed, then we can devise an FPRAS even for arbitrary FDs. For a database D and a set Σ of FDs, we denote by $\text{RS}^1(D, \Sigma)$ the set of sequences in $\text{RS}(D, \Sigma)$ mentioning only operations of the form $-f$, i.e., removing a single fact. Similarly, we denote $\text{Ops}_s^1(D, \Sigma) = \{s' \in \text{RS}^1(D, \Sigma) \mid s' = s \cdot op \text{ for some } D\text{-operation } op\}$. Then, we define the Markov chain generator $M_\Sigma^{\text{uo}, 1}$ such that for every $s, s' \in \text{RS}^1(D, \Sigma)$, assuming that $M_\Sigma^{\text{uo}, 1}(D) = (V, E, P)$, if $s' \in \text{Ops}_s^1(D, \Sigma)$ then

$$P(s, s') = \frac{1}{|\text{Ops}_s^1(D, \Sigma)|}.$$

Observe, however, that the Markov chain generator $M_\Sigma^{\text{uo}, 1}$ is defined over all the sequences of $\text{RS}(D, \Sigma)$. If $s \in \text{RS}^1(D, \Sigma)$ but $s' \in \text{RS}(D, \Sigma) \setminus \text{RS}^1(D, \Sigma)$ (and $s' \in \text{Ops}_s(D, \Sigma)$), then we define $P(s, s') = 0$. If $s \in \text{RS}(D, \Sigma) \setminus \text{RS}^1(D, \Sigma)$, none of the leaves of the subtree T_s is reachable with non-zero probability, and thus, $P(s, s')$, for any $s' \in \text{Ops}_s(D, \Sigma)$, can get an arbitrary probability (as long as the sum of probabilities equals one), e.g., $\frac{1}{|\text{Ops}_s(D, \Sigma)|}$.

We can now show that, assuming singleton removals, for FDs the problem of interest admits an FPRAS. The formal statement, already given in the main body of the paper, and its proof follow:

THEOREM 7.5. *For a set Σ of FDs, and a CQ Q , $\text{OCQA}(\Sigma, M_\Sigma^{\text{uo}, 1}, Q)$ admits an FPRAS.*

The proof consists of the usual two steps: (1) existence of an efficient sampler, and (2) provide a polynomial lower bound on the target probability.

Step 1: Efficient Sampler. We can sample elements of $\text{RL}(M_\Sigma^{\text{uo}, 1}(D))$ according to the leaf distribution of $M_\Sigma^{\text{uo}, 1}(D)$ in polynomial time in $\|D\|$. This is done by employing the same iterative algorithm as the one used to sample elements of $\text{RL}(M_\Sigma^{\text{uo}}(D))$, but with the difference that only justified operations that consist of singleton removals are considered. In particular, at each step, the algorithm

extends the current sequence s by selecting one of the $(s(D), \Sigma)$ -justified operations of the form $-f$ with probability

$$\frac{1}{|\text{Ops}_\Sigma^1(D, \Sigma)|}.$$

Hence, we immediately obtain the following result:

LEMMA D.7. *Given a database D , and a set Σ of keys, we can sample elements of $\text{RL}(M_\Sigma^{\text{uo},1}(D))$ according to the leaf distribution of $M_\Sigma^{\text{uo},1}(D)$ in polynomial time in $\|D\|$.*

Step 2: Polynomial Lower Bound. It remains to show that there exists a polynomial lower bound on the target probability.

LEMMA D.8. *Consider a set Σ of keys, and a CQ $Q(\bar{x})$. For every database D , and $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$P_{M_\Sigma^{\text{uo},1}, Q}(D, \bar{c}) \geq \frac{1}{(e \cdot \|D\|)^{\|Q\|}}$$

whenever $P_{M_\Sigma^{\text{uo},1}, Q}(D, \bar{c}) > 0$.

PROOF. Consider a database D . If there is no homomorphism h from Q to D such that $h(Q) \models \Sigma$ and $h(\bar{x}) = \bar{c}$, then clearly $P_{M_\Sigma^{\text{uo},1}, Q}(D, \bar{c}) = 0$. We now focus on the case where such a homomorphism h exists. Assume that $|h(Q)| = m$ for some $m \leq |Q|$. We prove by induction on n , that is, the number of facts in $D \setminus h(Q)$ that are involved in violations of the FDs (i.e., the facts $f \in (D \setminus h(Q))$ such that $\{f, g\} \not\models \Sigma$ for some $g \in D$), the following:

$$P_{D, M_\Sigma^{\text{uo},1}, Q}(h) \geq \frac{1}{\binom{n+m}{m}}.$$

Base Case. For $n = 0$, since $h(Q) \models \Sigma$, there are no violations of the FDs in D , and D has a single operational repair, which is D itself. In this case, the probability of obtaining an operational repair that contains all the facts of $h(Q)$ is $1 = \frac{1}{\binom{0+m}{m}}$, as needed.

Inductive Step. We now assume that the claim holds for databases where $n = 0, \dots, k-1$, and we prove that it holds for databases D where $n = k$. Every repairing sequence $s \in \text{RL}(M_\Sigma^{\text{uo},1}(D))$ for which $h(Q) \subseteq s(D)$ is such that the first operation of s removes a fact of $D \setminus h(Q)$ that is involved in violations of the FDs. Let f_1, \dots, f_k be these facts of $D \setminus h(Q)$, and for each $i \in \{1, \dots, k\}$, let op_i be the operation that removes the fact f_i . We then have that

$$P(\varepsilon, (op_i)) \geq \frac{1}{k+m}.$$

This is because the probability of removing a certain fact is $\frac{1}{k+p}$, where p is the number of facts involved in violations among the facts of $h(Q)$. Since $p \leq m$, we get that $\frac{1}{k+m} \leq \frac{1}{k+p}$.

After removing a conflicting fact of $D \setminus h(Q)$ from the database, we have at most $k-1$ such facts left, regardless of which specific fact we remove. For every $i \in \{1, \dots, p\}$, we denote by D_i the database $op_i(D)$ and by n_i the number of facts of $D_i \setminus h(Q)$ that are involved in violations of the FDs; hence, we have that $n_i \leq k-1$. By the inductive hypothesis, we have that

$$P_{D_i, M_\Sigma^{\text{uo},1}, Q}(h) \geq \frac{1}{\binom{n_i+m}{m}} \geq \frac{1}{\binom{k-1+m}{m}}.$$

Clearly, every sequence $s \in \text{RL}(M_\Sigma^{\text{uo}}(D))$ with $h(Q) \subseteq s(D)$ is of the form $op_i \cdot s_i$ for some $i \in \{1, \dots, k\}$ and $s_i \in \text{RL}(M_\Sigma^{\text{uo}}(D_i))$ with $h(Q) \subseteq s_i(D)$. Now, the following holds

$$P_{D, M_\Sigma^{\text{uo},1}, Q}(h) = \sum_{i=1}^k \left(P(\varepsilon, (op_i)) \times P_{D_i, M_\Sigma^{\text{uo},1}, Q}(h) \right).$$

Therefore, we can conclude that

$$\begin{aligned} P_{D, M_\Sigma^{\text{uo},1}, Q}(h) &\geq \sum_{i=1}^k \left(\frac{1}{k+m} \times \frac{1}{\binom{k-1+m}{m}} \right) \\ &= \frac{k}{k+m} \times \frac{1}{\binom{k-1+m}{m}} \\ &= \frac{k}{k+m} \times \frac{1}{\frac{(k-1+m)!}{m! \times (k-1)!}} \\ &= \frac{m! \times k!}{(k+m)!} \\ &= \frac{1}{\binom{k+m}{m}}. \end{aligned}$$

Finally, it is well known that $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$. We conclude that, for a database D such that $D \setminus h(Q)$ contains n facts that are involved in violations of the FDs, we have that

$$\begin{aligned} P_{D, M_\Sigma^{\text{uo},1}, Q}(h) &\geq \frac{1}{\binom{n+m}{n}} \\ &\geq \frac{1}{\left(\frac{e(n+m)}{n}\right)^m} \\ &= \frac{m^m}{e^m} \times \frac{1}{(n+m)^m} \\ &\geq \left(\frac{m}{e}\right)^m \times \frac{1}{|D|^m} \\ &\geq \left(\frac{1}{e}\right)^{|Q|} \times \frac{1}{|D|^{|Q|}}. \end{aligned}$$

Since $h(Q) \subseteq D'$ implies $\bar{c} \in Q(D')$, it holds that

$$P_{M_\Sigma^{\text{uo},1}, Q}(D, \bar{c}) \geq P_{D, M_\Sigma^{\text{uo},1}, Q}(h) \geq \frac{1}{(e|D|)^{|Q|}} \geq \frac{1}{(e\|D\|)^{\|Q\|}},$$

which concludes our proof. \square

E SINGLETON OPERATIONS

As mentioned in the main body of the paper (see the last paragraph of Section 7), focusing on singleton operations does not affect Theorem 5.1, Theorem 6.1, and item (1) of Theorem 7.1 that deals with exact query answering. In this section, we formally prove the above statements. But let us first briefly discuss the Markov chain generators based on uniform repairs and sequences that consider only singleton operations. The version of the Markov chain generator based on uniform operations that considers only singleton operations has been already discussed in the previous section.

Given a database D and a set Σ of FDs, we write $\text{CRS}^1(D, \Sigma)$ for the set of sequences in $\text{CRS}(D, \Sigma)$ mentioning only operations of the form $-f$, i.e., removing a single fact. Similarly, we define $\text{CORep}^1(D, \Sigma) = \{D' \in \text{CORep}(D, \Sigma) \mid s(D) = D' \text{ for some } s \in$

$\text{CRS}^1(D, \Sigma)$. Our intention is to focus on the repairing Markov chain generators $M_{\Sigma}^{\text{ur},1}$ and $M_{\Sigma}^{\text{us},1}$ enjoying the following:

- (1) $\text{ORep}(D, M_{\Sigma}^{\text{ur},1}) = \text{CORep}^1(D, \Sigma)$, and for every repair $D' \in \text{ORep}(D, M_{\Sigma}^{\text{ur},1})$, $P_{D, M_{\Sigma}^{\text{ur},1}}(D') = \frac{1}{|\text{ORep}(D, M_{\Sigma}^{\text{ur},1})|}$.
- (2) For every $s \in \text{CRS}^1(D, \Sigma)$, assuming that π is the leaf distribution of $M_{\Sigma}^{\text{us},1}(D)$, $\pi(s) = \frac{1}{|\text{CRS}^1(D, \Sigma)|}$.

It is not difficult to adapt Definitions A.1 and A.3 in order to obtain the Markov chain generators $M_{\Sigma}^{\text{ur},1}$ and $M_{\Sigma}^{\text{us},1}$ with the above properties. We proceed with our results about singleton operations.

E.1 Uniform Repairs

In this section, we prove the version of Theorem 5.1 that considers singleton operations:

- THEOREM E.1.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{ur},1}, Q)$ is $\#P$ -hard.*
- (2) *For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{ur},1}, Q)$ admits an FPRAS.*
- (3) *Unless $\text{RP} = \text{NP}$, there exist a set Σ of FDs, and a CQ Q such that there is no FPRAS for $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{ur},1}, Q)$.*

As for Theorem 5.1, we can conveniently restate the problem of interest as the problem of computing a “relative frequency” ratio. Indeed, for a database D , a set Σ of FDs, a CQ $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$, $P_{M_{\Sigma}^{\text{ur},1}, Q}(D, \bar{c}) = \text{rrfreq}_{\Sigma, Q}^1(D, \bar{c})$, where

$$\text{rrfreq}_{\Sigma, Q}^1(D, \bar{c}) = \frac{|\{D' \in \text{CORep}^1(D, \Sigma) \mid \bar{c} \in Q(D')\}|}{|\text{CORep}^1(D, \Sigma)|}.$$

Hence, $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{ur},1}, Q(\bar{x}))$ coincides with the following problem, which is independent from the Markov chain generator $M_{\Sigma}^{\text{ur},1}$:

PROBLEM : $\text{RRFreq}^1(\Sigma, Q(\bar{x}))$
INPUT : A database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$.
OUTPUT : $\text{rrfreq}_{\Sigma, Q}^1(D, \bar{c})$.

We proceed to establish Theorem E.1 by directly considering the problem $\text{RRFreq}^1(\Sigma, Q)$ instead of $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{ur},1}, Q)$.

Proof of Item (1) of Theorem E.1. We provide a polynomial-time Turing reduction from the $\#P$ -hard problem $\#P\text{os2DNF}$ [21]. A positive 2DNF formula is a Boolean formula of the form $\varphi = C_1 \vee \dots \vee C_n$, where each C_i is a conjunction of two variables occurring positively in C_i . Let $\text{var}(\varphi)$ be the set of Boolean variables occurring in φ . An assignment for φ is a function $\mu : \text{var}(\varphi) \rightarrow \{0, 1\}$. Such an assignment is satisfying for φ if $\mu(\varphi) = 1$, i.e., the formula obtained after replacing each variable x of φ with $\mu(x)$, evaluates to 1. We write $\text{sat}(\varphi)$ for the set of satisfying assignments for φ , i.e., the assignments for φ that evaluate φ to 1. The problem in question is

PROBLEM : $\#P\text{os2DNF}$
INPUT : A positive 2DNF formula φ .
OUTPUT : The number $|\text{sat}(\varphi)|$.

Consider the schema $S = \{V/2, C/2, T/1\}$, and let (A, B) be the tuple of attributes of V . We define the set Σ over S consisting of

$$V : A \rightarrow B$$

and the (constant-free) Boolean CQ Q over S

$$\text{Ans}() :- C(x, y), V(x, z), V(y, z), T(z).$$

Our goal is to show that $\text{RRFreq}^1(\Sigma, Q)$ is $\#P$ -hard via a polynomial-time Turing reduction from $\#P\text{os2DNF}$. Given a positive 2DNF formula $\varphi = C_1 \vee \dots \vee C_n$, we define the database

$$D_{\varphi} = \{V(c_x, 0), V(c_x, 1) \mid x \in \text{var}(\varphi)\} \cup \underbrace{\{C(c_x, c_y) \mid C_i = (x \wedge y) \text{ for some } i \in [n]\} \cup \{T(1)\}}_{D_c},$$

where, for each $x \in \text{var}(\varphi)$, c_x is a constant, which essentially encodes φ . We now define the algorithm SAT , which accepts as input a positive 2DNF formula φ , and performs the following steps:

- (1) Construct the database D_{φ} .
- (2) Compute the number $r = \text{rrfreq}_{\Sigma, Q}^1(D_{\varphi}, ())$.
- (3) Output the number $2^{|\text{var}(\varphi)|} \cdot r$.

It is clear that the above algorithm runs in polynomial time in $\|\varphi\|$. Hence, to prove that it is indeed a Turing reduction from $\#P\text{os2DNF}$ to $\text{RRFreq}^1(\Sigma, Q)$, it suffices to prove that

$$\text{rrfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) = \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}}.$$

Since we consider only single fact removals, a database D is an operational repair of $\text{CORep}^1(D_{\varphi}, \Sigma)$ iff it is of the form

$$\{V(c_x, \star) \mid x \in \text{var}(\varphi) \text{ and } \star \in \{0, 1\}\} \cup D_c,$$

which keeps precisely one fact $V(c_x, \star)$, for each variable x in φ . Hence, $|\text{CORep}^1(D_{\varphi}, \Sigma)| = 2^{|\text{var}(\varphi)|}$. Thus, with $\text{CORep}^1(D_{\varphi}, \Sigma, Q)$ being the set of repairs D in $\text{CORep}^1(D_{\varphi}, \Sigma)$ such that $D \models Q$, it is easy to see that $|\text{CORep}^1(D_{\varphi}, \Sigma, Q)| = |\text{sat}(\varphi)|$. Consequently,

$$\text{rrfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) = \frac{|\text{CORep}^1(D_{\varphi}, \Sigma, Q)|}{|\text{CORep}^1(D_{\varphi}, \Sigma)|} = \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}},$$

and the claim follows.

Proof of Item (2) of Theorem E.1. We can employ a proof similar to the one underlying item (2) of Theorem 5.1, which consists of two steps: (1) existence of an efficient sampler, and (2) provide a polynomial lower bound for the target ratio. The key difference is that now we focus on the set of repairs $\text{CORep}^1(D, \Sigma)$, rather than $\text{CORep}(D, \Sigma)$. Thus, each repair in $\text{CORep}^1(D, \Sigma)$ is obtained by keeping from D precisely one fact from each block of D .

We first show the existence of an efficient sampler.

LEMMA E.2. *Given a database D , and a set Σ of primary keys, we can sample elements of $\text{CORep}^1(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

PROOF. Let B_1, \dots, B_n be the blocks of D w.r.t. Σ . That is, for every relation name R of the schema with $R : X \rightarrow Y \in \Sigma$, we split the set of facts of D over R into blocks of facts that agree on the values of all the attributes in X . If there is no such key in Σ , then every fact is a separate block. As aforementioned, every repair of $\text{CORep}^1(D, \Sigma)$ contains one fact of each block. Hence,

$$|\text{CORep}^1(D, \Sigma)| = |B_1| \times \dots \times |B_n|.$$

In order to sample an element of $\text{CORep}^1(D, \Sigma)$ with probability

$$\frac{1}{|B_1| \times \cdots \times |B_n|}$$

we simply need to select, for each block B_i , one of its $|B_i|$ possible outcomes (one of its facts that will appear in the repair), uniformly at random, namely with probability $\frac{1}{|B_i|}$. \square

It remains to show that there exists a polynomial lower bound on the target ratio.

LEMMA E.3. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{rrfreq}_{\Sigma, Q}^1(D, \bar{c}) \geq \frac{1}{(|D|)^{|\bar{x}|} |Q|}$$

whenever $\text{rrfreq}_{\Sigma, Q}^1(D, \bar{c}) > 0$.

PROOF. Let D be a database. If there is no homomorphism h from Q to D such that $h(Q) \models \Sigma$ and $h(\bar{x}) = \bar{c}$, then it clearly holds that

$$\text{rrfreq}_{\Sigma, Q}^1(D, \bar{c}) = 0.$$

We now consider the case where such a homomorphism h exists. Assume that $|h(Q)| = m$ for some $1 \leq m \leq |Q|$. As in the proof of Lemma E.2, let B_1, \dots, B_n be the blocks of D w.r.t. Σ . Assume, without loss of generality, that the facts of $h(Q)$ belong to the blocks B_1, \dots, B_m . Note that no two facts of $h(Q)$ belong to the same block, as two facts that belong to the same block always jointly violate the corresponding key, and it holds that $h(Q) \models \Sigma$.

Since all the facts of a block are symmetric to each other, each of these facts has an equal chance to appear in a repair. In particular, every operational repair contains one fact from each block in $\{B_1, \dots, B_m\}$, and precisely

$$\frac{1}{|B_1| \times \cdots \times |B_m|}$$

repairs of $\text{CORep}^1(D, \Sigma)$ contain all the facts of $h(Q)$. Hence,

$$\begin{aligned} \text{rrfreq}_{\Sigma, Q}^1(D, \bar{c}) &\geq \frac{|\{E \in \text{CORep}^1(D, \Sigma) \mid h(Q) \subseteq E\}|}{|\text{CORep}^1(D, \Sigma)|} \\ &\geq \frac{\frac{1}{|B_1| \times \cdots \times |B_m|} \times |\text{CORep}^1(D, \Sigma)|}{|\text{CORep}^1(D, \Sigma)|} \\ &= \frac{1}{|B_1| \times \cdots \times |B_m|} \\ &\geq \frac{1}{|D|^m} \\ &\geq \frac{1}{|D|^{|Q|}} \\ &\geq \frac{1}{(|D|)^{|\bar{x}|} |Q|}, \end{aligned}$$

and the claim follows. \square

Proof of Item (3) of Theorem E.1. The proof of this item proceeds similarly to the one used to prove item (3) of Theorem 5.1. Here we highlight the key differences.

We first need to prove a result analogous to Lemma 5.4, but for the setting of singleton operations. For an undirected graph G , $\text{IS}_{\neq \emptyset}(G)$ denotes the set of all *non-empty* independent sets of G .

LEMMA E.4. *Consider a non-trivially Σ -connected database D , where Σ is a set of FDs. Then, $|\text{CORep}^1(D, \Sigma)| = |\text{IS}_{\neq \emptyset}(\text{CG}(D, \Sigma))|$.*

PROOF. (\subseteq) Consider a candidate repair $D' \in \text{CORep}^1(D, \Sigma)$. By definition, D' is consistent w.r.t. Σ , i.e., there are no two facts f, g of D' such that $\{f, g\} \not\models \Sigma$. Thus, by definition of the conflict graph of D w.r.t. Σ , we get that no two facts of D' are connected via an edge in $\text{CG}(D, \Sigma)$. Hence, D' is an independent set of $\text{CG}(D, \Sigma)$. It remains to show that $D' \neq \emptyset$. Since $D' \in \text{CORep}^1(D, \Sigma)$, there is a sequence $s = op_1, \dots, op_n \in \text{CRS}^1(D, \Sigma)$ such that $s(D) = D'$. Since op_n must be $(s_{n-1}(D), \Sigma)$ -justified, there must be a violation $(\phi, \{f, g\}) \in \text{V}(s_{n-1}(D), \Sigma)$, for some FD $\phi \in \Sigma$. Moreover, since $s(D) \models \Sigma$, this is the only violation. Hence, $op_n = -f$, and then $g \in s(D)$, or $op_n = -g$, and then $f \in s(D)$. Thus, $s(D) = D' \neq \emptyset$.

(\supseteq) Consider now an independent set $D' \in \text{IS}_{\neq \emptyset}(\text{CG}(D, \Sigma))$, which is by definition non-empty. We have already shown in the proof of Lemma 5.4 that one can construct a sequence $s \in \text{CRS}(D, \Sigma)$ such that $s(D) = D'$. In particular, by inspecting that proof, we can see that indeed s uses only operations of the form $-f$, and thus, $s \in \text{CRS}^1(D, \Sigma)$. Hence, $D' \in \text{CORep}^1(D, \Sigma)$. \square

The rest of the proof proceeds in two steps. We first prove the following result, which is analogous to Proposition 5.5. We write $\#\text{CORep}^{\text{con}, 1}(\Sigma)$ for the problem of computing $|\text{CORep}^1(D, \Sigma)|$, given a non-trivially Σ -connected database D .

PROPOSITION E.5. *Unless $\text{RP} = \text{NP}$, there exists a set Σ of keys over $\{R\}$ such that $\#\text{CORep}^{\text{con}, 1}(\Sigma)$ does not admit an FPRAS.*

PROOF. We provide a reduction from the problem of counting *non-empty* independent sets of non-trivially connected graphs of bounded degree. With $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$, for some integer $\Delta \geq 0$, being the problem of computing $|\text{IS}_{\neq \emptyset}(G)|$, given a non-trivially connected graph G with degree Δ , we first need to prove that:

LEMMA E.6. *Unless $\text{RP} = \text{NP}$, $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$ has no FPRAS, for all $\Delta \geq 6$.*

PROOF. By contradiction, assume that $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$ admits an FPRAS, for some $\Delta \geq 6$. We then show that $\#\text{IS}_{\Delta}^{\text{con}}$ admits an FPRAS, contradicting Lemma B.5. Assume that A is an FPRAS for $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$. Let A' be the randomized algorithm that, given a non-trivially connected undirected graph G , $\epsilon > 0$ and $0 < \delta < 1$, is such that $A'(G, \epsilon, \delta) = A(G, \epsilon, \delta) + 1$. We proceed to show that A' is an FPRAS for $\#\text{IS}_{\Delta}^{\text{con}}$. Since A is an FPRAS for $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$,

$$\Pr((1 - \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| \leq A(G, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)|) \geq 1 - \delta.$$

By adding 1 in all sides of the inequality, we obtain that

$$\Pr((1 - \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 \leq A'(G, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1) \geq 1 - \delta.$$

Since

$$(1 - \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 \geq (1 - \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 - \epsilon$$

$$(1 + \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 \leq (1 + \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 + \epsilon,$$

by factorizing the terms in the two inequalities, we obtain that

$$\begin{aligned} (1 - \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 &\geq (1 - \epsilon) \cdot (|\text{IS}_{\neq \emptyset}(G)| + 1) \\ (1 + \epsilon) \cdot |\text{IS}_{\neq \emptyset}(G)| + 1 &\leq (1 + \epsilon) \cdot (|\text{IS}_{\neq \emptyset}(G)| + 1). \end{aligned}$$

Since $|\text{IS}_{\neq \emptyset}(G)| + 1 = |\text{IS}(G)|$, we conclude that

$$\Pr((1 - \epsilon) \cdot |\text{IS}(G)| \leq A'(G, \epsilon, \delta) \leq (1 + \epsilon) \cdot |\text{IS}(G)|) \geq 1 - \delta,$$

and the claim follows. \square

With the above lemma in place, we establish our main claim by showing that there exists a set Σ_K of keys such that, given a non-trivially connected undirected graph G , we can construct a non-trivially Σ_K -connected database D_G in polynomial time in $\|G\|$ such that $|\text{IS}_{\neq \emptyset}(G)| = |\text{COREP}^1(D_G, \Sigma_K)|$. Hence, unless $\text{RP} = \text{NP}$, the existence of an FPRAS for $\#\text{COREP}^{\text{con},1}(\Sigma_K)$ would imply an FPRAS for $\#\text{IS}_{\Delta, \neq \emptyset}^{\text{con}}$, contradicting Lemma E.6.

The set Σ_K and the database D_G are defined in exactly the same way as in the proof of Proposition 5.5. We recall that D_G and Σ_K are such that $|\text{IS}(G)| = |\text{IS}(\text{CG}(D_G, \Sigma_K))|$. Hence, $|\text{IS}_{\neq \emptyset}(G)| = |\text{IS}_{\neq \emptyset}(\text{CG}(D_G, \Sigma_K))|$. Since D_G is non-trivially Σ_K -connected, by Lemma E.4, $|\text{IS}_{\neq \emptyset}(\text{CG}(D_G, \Sigma_K))| = |\text{COREP}^1(D_G, \Sigma_K)|$. Hence, $|\text{IS}_{\neq \emptyset}(G)| = |\text{COREP}^1(D_G, \Sigma_K)|$, as needed. \square

It remains to prove a result analogous to Lemma 5.6. Let Σ_K be the set of keys provided by Proposition E.5.

LEMMA E.7. *Assume that $\text{RRFREQ}^1(\Sigma, Q)$ admits an FPRAS, for every set Σ of FDs and CQ Q . $\#\text{COREP}^{\text{con},1}(\Sigma_K)$ admits an FPRAS.*

PROOF. The proof of this claim proceeds in the same way as the one of Lemma 5.6. The key difference is that now, given a non-trivially Σ_K -connected database D , we must show that for the set Σ_F of FDs and the Boolean CQ Q_F as defined in that proof, the database D_F obtained from D is such that

$$\text{rrfreq}_{\Sigma_F, Q_F}^1(D_F, ()) = \frac{1}{|\text{COREP}^1(D, \Sigma_K)| + 1}.$$

This is done using the same argument as in the proof of Lemma 5.6, with the difference that we exploit Lemma E.4, instead of Lemma 5.4, to prove that $|\text{COREP}^1(D_F, \Sigma_F)| = |\text{COREP}^1(D, \Sigma_K)| + 1$. \square

It is now straightforward to see that from Proposition E.5 and Lemma E.7, we can conclude item (3) of Theorem E.1.

E.2 Uniform Sequences

In this section, we prove the version of Theorem 6.1 that considers singleton operations:

- THEOREM E.8.** (1) *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us},1}, Q)$ is $\#\text{P}$ -hard.*
(2) *For a set Σ of primary keys, and a CQ Q , $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us},1}, Q)$ admits an FPRAS.*

As for Theorem 6.1, we can conveniently restate the problem of interest as the problem of computing a “relative frequency” ratio.

Indeed, for a database D , a set Σ of FDs, a CQ $Q(\bar{x})$, and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$, $P_{M_{\Sigma}^{\text{us},1}, Q}(D, \bar{c}) = \text{srfreq}_{\Sigma, Q}^1(D, \bar{c})$, where

$$\text{srfreq}_{\Sigma, Q}^1(D, \bar{c}) = \frac{|\{s \in \text{CRS}^1(D, \Sigma) \mid \bar{c} \in Q(s(D))\}|}{|\text{CRS}^1(D, \Sigma)|}.$$

Hence, $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us},1}, Q(\bar{x}))$ coincides with the following problem, which is independent from the Markov chain generator $M_{\Sigma}^{\text{us},1}$:

PROBLEM : $\text{SRFREQ}^1(\Sigma, Q(\bar{x}))$
INPUT : A database D , and a tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$.
OUTPUT : $\text{srfreq}_{\Sigma, Q}^1(D, \bar{c})$.

We proceed to establish Theorem E.8 by directly considering the problem $\text{SRFREQ}^1(\Sigma, Q(\bar{x}))$ instead of $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{us},1}, Q)$.

Proof of Item (1) of Theorem E.8. We provide a polynomial-time Turing reduction from $\#\text{Pos2DNF}$. In fact, the reduction is identical to the one used to prove item (1) of Theorem E.1. We only need to argue that, given a positive 2DNF formula φ ,

$$\text{srfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) = \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}},$$

where D_{φ} , Σ and Q are as in the proof of item (1) of Theorem E.1.

A database D is a repair in $\text{COREP}^1(D_{\varphi}, \Sigma)$ iff it keeps from D_{φ} precisely one fact $V(c_x, \star)$, for each variable x of φ . Hence, $|\text{COREP}^1(D_{\varphi}, \Sigma)| = 2^{|\text{var}(\varphi)|}$. Moreover, since no two violations in $V(D_{\varphi}, \Sigma)$ share a fact, each such a repair is the result of precisely $|\text{var}(\varphi)|!$ sequences of $\text{CRS}^1(D_{\varphi}, \Sigma)$ (i.e., operations can be applied in any order). Hence, $|\text{CRS}^1(D_{\varphi}, \Sigma)| = 2^{|\text{var}(\varphi)|} \cdot |\text{var}(\varphi)|!$. Thus, with $\text{CRS}^1(D_{\varphi}, \Sigma, Q)$ being the set of sequences s of $\text{CRS}^1(D_{\varphi}, \Sigma)$ such that $s(D_{\varphi}) \models Q$, it is straightforward to see that $|\text{CRS}^1(D_{\varphi}, \Sigma, Q)| = |\text{sat}(\varphi)| \cdot |\text{var}(\varphi)|!$. Therefore,

$$\begin{aligned} \text{srfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) &= \frac{|\text{CRS}^1(D_{\varphi}, \Sigma, Q)|}{|\text{CRS}^1(D_{\varphi}, \Sigma)|} \\ &= \frac{|\text{sat}(\varphi)| \cdot |\text{var}(\varphi)|!}{2^{|\text{var}(\varphi)|} \cdot |\text{var}(\varphi)|!} \\ &= \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}}, \end{aligned}$$

and the claim follows.

Proof of Item (2) of Theorem E.8. As for item (2) of Theorem 6.1, the proof consists of two steps: (1) existence of an efficient sampler, and (2) provide a polynomial lower bound on the target ratio.

We first show that an efficient sampler exists.

LEMMA E.9. *Given a database D , and a set Σ of primary keys, we can sample elements of $\text{CRS}^1(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

PROOF. The algorithm `SampleSeq` (Algorithm 1) that is used to sample elements of $\text{CRS}(D, \Sigma)$ can be used to sample elements of $\text{CRS}^1(D, \Sigma)$ as well. The only difference lies on the set of $(s(D), \Sigma)$ -justified operations that, in the case of $\text{CRS}(D, \Sigma)$ consists of both single-fact removals and pair removals, while in the case of $\text{CRS}^1(D, \Sigma)$ it consists only of single-fact removals. \square

We now show the polynomial lower bound on the target ratio.

LEMMA E.10. *Consider a set Σ of primary keys, and a CQ $Q(\bar{x})$. For every database D , and tuple $\bar{c} \in \text{dom}(D)^{|\bar{x}|}$,*

$$\text{srfreq}_{\Sigma, Q}^1(D, \bar{c}) \geq \frac{1}{(|D|)^{|\bar{x}|}}$$

whenever $\text{srfreq}_{\Sigma, Q}^1(D, \bar{c}) > 0$.

PROOF. Let D be a database. If there is no homomorphism h from Q to D such that $h(Q) \models \Sigma$ and $h(\bar{x}) = \bar{c}$, then clearly it holds that

$$\text{srfreq}_{\Sigma, Q}^1(D, \bar{c}) = 0.$$

We now consider the case where such a homomorphism h exists. Assume that $|h(Q)| = m$ for some $1 \leq m \leq |Q|$. As in the proof of Lemma E.3, let B_1, \dots, B_n be the blocks of D w.r.t. Σ . Assume, w.l.o.g., that the facts of $h(Q)$ belong to the blocks B_1, \dots, B_m .

Since all the facts of a block are symmetric to each other, if for some $f \in B_i$, there are m sequences s in $\text{CRS}^1(B_i, \Sigma)$ such that $f \in s(B_i)$, then the same holds for every fact $g \in B_i$. Moreover, since every operational repair of $\text{RL}(M_{\Sigma}^{\text{uo}, 1})$ keeps precisely one fact from each block, and the blocks are independent (in the sense that an operation over some block has no impact on the justified operations of another block), we can conclude that precisely

$$\frac{1}{|B_1| \times \dots \times |B_m|}$$

of the sequences s in $\text{CRS}^1(D, \Sigma)$ are such that $h(Q) \subseteq s(D)$ (i.e., the sequence s keeps the fact $B_i \cap h(Q)$ for every $B_i \in \{B_1, \dots, B_m\}$).

We then have that

$$\begin{aligned} \text{srfreq}_{\Sigma, Q}^1(D, \bar{c}) &\geq \frac{|\{s \in \text{CRS}^1(D, \Sigma) \mid h(Q) \subseteq s(D)\}|}{|\text{CRS}^1(D, \Sigma)|} \\ &\geq \frac{\frac{1}{|B_1| \times \dots \times |B_m|} \times |\text{CRS}^1(D, \Sigma)|}{|\text{CRS}^1(D, \Sigma)|} \\ &= \frac{1}{|B_1| \times \dots \times |B_m|} \\ &\geq \frac{1}{|D|^m} \\ &\geq \frac{1}{|D|^{|Q|}} \\ &\geq \frac{1}{(|D|)^{|\bar{x}|}}, \end{aligned}$$

and the claim follows. \square

E.3 Uniform Operations

In this last section, we prove that item (1) of Theorem 7.1 holds also in the case of singleton operations.

THEOREM E.11. *There exist a set Σ of primary keys, and a CQ Q such that $\text{OCQA}(\Sigma, M_{\Sigma}^{\text{uo}, 1}, Q)$ is $\#P$ -hard.*

PROOF. We use the reduction from the proof of Theorem E.1(1). We only need to argue that, given a positive 2DNF formula φ ,

$$P_{M_{\Sigma}^{\text{uo}, 1}, Q}(D_{\varphi}, ()) = \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}},$$

where D_{φ}, Σ and Q are as in the proof of item (1) of Theorem E.1.

Let $M_{\Sigma}^{\text{uo}, 1}(D_{\varphi}) = (V, E, \mathbf{P})$. By the definition of the Markov chain generator, $\text{RL}(M_{\Sigma}^{\text{uo}, 1}(D_{\varphi})) = \text{CRS}^1(D_{\varphi}, \Sigma)$. Moreover, we note that each variable x of φ induces a violation $\{V(c_x, 0), V(c_x, 1)\}$ in D_{φ} , which can be resolved with one of two operations removing a single fact. Hence, every complete sequence in $\text{CRS}^1(D_{\varphi}, \Sigma)$ is of length precisely $|\text{var}(\varphi)|$, and for every non-leaf node $s \in V$ that is also in $\text{Ops}_{D_{\varphi}}^1(\Sigma)$, $|\text{Ops}_s^1(D_{\varphi}, \Sigma)| = 2 \cdot (|\text{var}(\varphi)| - |s|)$. Hence, by Definition of $M_{\Sigma}^{\text{uo}, 1}$, with π being the leaf distribution of $M_{\Sigma}^{\text{uo}, 1}(D_{\varphi})$, for each $s = op_1, \dots, op_n \in \text{CRS}^1(D_{\varphi}, \Sigma) = \text{RL}(M_{\Sigma}^{\text{uo}, 1}(D_{\varphi}))$,

$$\pi(s) = \mathbf{P}(s_0, s_1) \cdots \mathbf{P}(s_{n-1}, s_n) = \frac{1}{2^{|\text{var}(\varphi)|} \cdot |\text{var}(\varphi)!}.$$

This means that each sequence $s \in \text{CRS}^1(D_{\varphi}, \Sigma) = \text{RL}(M_{\Sigma}^{\text{uo}, 1}(D_{\varphi}))$ is assigned the same non-zero probability, i.e., π is the uniform distribution over $\text{CRS}^1(D_{\varphi}, \Sigma)$. The latter implies that $P_{M_{\Sigma}^{\text{uo}, 1}, Q}(D_{\varphi}, ()) = \text{srfreq}_{\Sigma, Q}^1(D_{\varphi}, ())$. As we have already seen that

$$\text{rrfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) = \text{srfreq}_{\Sigma, Q}^1(D_{\varphi}, ()) = \frac{|\text{sat}(\varphi)|}{2^{|\text{var}(\varphi)|}}$$

the claim follows. \square