Uniform Operational Consistent Query Answering

MARCO CALAUTTI, Department of Computer Science, University of Milan, Milano, Italy
ESTER LIVSHITS, School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland
ANDREAS PIERIS, School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland and Department of Computer Science, University of Cyprus, Nicosia, Cyprus
MARKUS SCHNEIDER, School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland

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.

CCS Concepts: • **Information systems → Data management systems;**

Additional Key Words and Phrases: Inconsistency, consistent query answering, operational semantics, functional dependencies, conjunctive queries, randomized approximation schemes

This work was supported by the EPSRC grants EP/S003800/1, EP/L016427/1 and EP/T022124/1, and by the European Union - Next Generation EU under the MUR PRIN-PNRR grant P2022KHTX7 "DISTORT".
Authors' Contact Information: Marco Calautti, Department of Computer Science, University of Milan, Milano, Italy; e-mail: marco.calautti@unimi.it; Ester Livshits, School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland; e-mail: ester.livshits@ed.ac.uk; Andreas Pieris (corresponding author), School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland and Department of Computer Science, University of Cyprus, Nicosia, Nicosia, Cyprus; e-mail: pieris.andreas@gmail.com; Markus Schneider, School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom of Great Britain and Northern Ireland; e-mail: m.schneider@ed.ac.uk.

ACM 0362-5915/2026/03-ART15
https://doi.org/10.1145/3774316

ACM Reference Format:
Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. 2026. Uniform Operational Consistent Query Answering. *ACM Trans. Datab. Syst.* 51, 3, Article 15 (March 2026), 65 pages. https://doi.org/10.1145/3774316

1 Introduction

Consistent query answering (**CQA**) is an elegant framework, introduced in the late 1990s by Arenas, Bertossi, and Chomicki [Arenas et al. 1999], that allows us to compute conceptually meaningful answers to queries posed over inconsistent databases, that is, databases that do not conform to their specifications given in the form of integrity constraints. 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 [Chomicki and Marcinkowski 2005]), there was a great effort on drawing the tractability boundary for CQA; see, e.g., [Fuxman et al. 2005; Fuxman and Miller 2007; Geerts et al. 2015; Koutris and Suciu 2014; Koutris and Wijsen 2015, 2021]. 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 [Calautti et al. 2018], 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 [Karp and Lipton 1980]. 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 [Maslowski and Wijsen 2013]. Indeed, for primary keys and conjunctive queries, one can approximate the relative frequency via an FPRAS; this is implicit in [Dalvi and Suciu 2007] and it has been made explicit in [Calautti et al. 2019]. Moreover, a recent experimental evaluation revealed that approximate CQA in the presence of primary keys and conjunctive queries is not unrealistic in practice [Calautti et al. 2021]. However, it seems that the simple case of primary keys is the limit of this approach. We have recently shown 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 [Calautti et al. 2022].

The above limitations of the classical CQA approach led to a new framework for CQA, based on revised definitions of repairs and consistent answers, that opens up the possibility of efficient approximations with error guarantees [Calautti et al. 2018]. The main idea 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 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 using an example from [Calautti et al. 2018]:

Example 1.1. Consider a data integration scenario that results in a database containing the facts $\mathrm{Emp}(1, \mathrm{Alice})$ and $\mathrm{Emp}(1, \mathrm{Tom})$ 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 $\mathrm{Emp}(1, \mathrm{Alice})$ or $\mathrm{Emp}(1, \mathrm{Tom})$. Therefore, we have three repairs, each with its probability: the empty repair with probability 0.25, and the repairs consisting of $\mathrm{Emp}(1, \mathrm{Alice})$ or $\mathrm{Emp}(1, \mathrm{Tom})$ with probability 0.375. Note that the standard CQA approach only allows the removal of one of the two facts (with equal probability 0.5). It somehow assumes that we trust at least one of the sources, even though they are in conflict.

The preliminary data complexity analysis of operational CQA performed in [Calautti et al. 2018] 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 [Calautti et al. 2018] is the following: *how can we choose the probabilities assigned to operations in a conceptually meaningful way and, at the same time, 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*; further justification of the above choices is given in Section 4, where the formal definitions underlying uniform operational CQA are given. 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, which, as said above, seems to be the limit of the classical CQA approach.

Roadmap. In Section 2, we give the very basics on relational databases, functional dependencies, and conjunctive queries. In Section 3, we recall the operational approach to consistent query answering from [Calautti et al. 2018] focussing on functional dependencies. In Section 4, we introduce uniform operational CQA, which is the main concern of the present work. The complexity analysis of uniform operational CQA is performed in Sections 5–8. We conclude with Section 9.

2 Preliminaries

We recall the basics on relational databases, functional dependencies, and conjunctive queries. In the rest of the article, we assume the disjoint countably infinite sets $\mathbf{C}$ and $\mathbf{V}$ of *constants* and *variables*, respectively. For $n > 0$, we write $[n]$ for the set of integers $\{1, \dots, n\}$.

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 $R(c_1, \dots, c_n)$, 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, that is, the value of the attribute A_i in the fact f.

Functional Dependencies. A *functional dependency* (FD) ϕ over a schema $\mathbf{S}$ is an expression

$$R : X \to Y,$$

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 $R : X \to Y$. A database D satisfies an FD $\phi = R : X \to Y$, 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. An *atom* α over a schema $\mathbf{S}$ is an expression $R(t_1, \dots, t_n)$, 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

$$\text{Ans}(\bar{x}) \text{ :- } R_1(\bar{y}_1), \dots, R_n(\bar{y}_n),$$

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 as the one above to a database D is a function $h : \text{var}(Q) \cup \text{const}(Q) \to \text{dom}(D)$, 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 $() \in Q(D)$.

3 Operational CQA

We proceed to recall the operational approach to consistent query answering introduced in [Calautti et al. 2018]. 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 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. We write $\mathcal{P}(S)$ for the powerset of a set S.

Definition 3.1 (Operation). For a database D over a schema $\mathbf{S}$, a *D-operation* is a function $op : \mathcal{P}(D) \to \mathcal{P}(D)$ such that, for a non-empty set $F \subseteq D$, 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 $-\{f\}$. 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, 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 $\mathbf{S}$, a *D-violation* of an FD $\phi = R : X \to Y$ over $\mathbf{S}$ is a set $\{f, g\} \subseteq D$ such that $\{f, g\} \not\models \phi$. We denote the set of D-violations of ϕ by $\mathsf{V}(D, \phi)$. For a set Σ of FDs, we denote by $\mathsf{V}(D, \Sigma)$ the set $\{(\phi, v) \mid \phi \in \Sigma \text{ and } v \in \mathsf{V}(D, \phi)\}$.

Therefore, a pair $(\phi, \{f, g\}) \in \mathsf{V}(D, \Sigma)$ 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. Now, 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 $\mathbf{S}$ and Σ a set of FDs over $\mathbf{S}$. For a database $D' \subseteq D$, a D-operation $-F$ is called *(D', Σ)-justified* if there exists $(\phi, \{f, g\}) \in \mathsf{V}(D', \Sigma)$ such that $F \subseteq \{f, g\}$.

Note that justified operations do not try to minimize the number of facts that need to be removed. As argued in [Calautti et al. 2018], a set of facts that collectively contributes to a violation is 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 scenarios.

Repairing Sequences. As said above, the main idea of the operational approach is to iteratively apply justified operations, starting from an inconsistent database, until we reach a consistent one. This is formalized via the notion of repairing sequence. Consider a database D and a set Σ of FDs. Given a sequence $s = (op_i)_{i \in [n]}$ of D-operations, let

$$D_0^s = D \qquad \text{and} \qquad D_i^s = op_i(D_{i-1}^s)$$

for $i \in [n]$, that is, D_i^s is obtained by applying to D the first i operations of s.

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

It is clear that the length of a (D, Σ)-repairing sequence is linear in the size of D. It is also clear that the set $\mathsf{RS}(D, \Sigma)$ is finite. For a (D, Σ)-repairing sequence $s = (op_i)_{i \in [n]}$, we define its *result* as the database $s(D) = D_n^s$ and call it *complete* if $s(D) \models \Sigma$, i.e., it leads to a consistent database. Let $\mathsf{CRS}(D, \Sigma)$ 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 $s \in \mathsf{CRS}(D, \Sigma)$. We write $\mathsf{CORep}(D, \Sigma)$ for the set of all candidate repairs of D w.r.t. Σ. Although every database of $\mathsf{CORep}(D, \Sigma)$ corresponds to a conceptually meaningful way of repairing the database D, we would like to have a mechanism

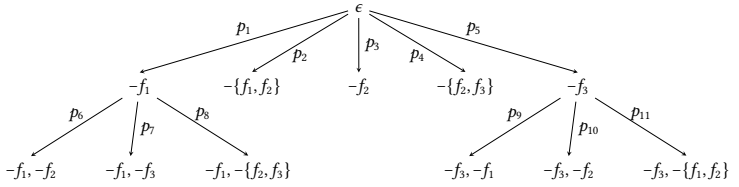


Fig. 1. Repairing markov chain.

that allows us to choose which candidate repairs should be really 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 $s \cdot op_1, \ldots, s \cdot op_k$ of a repairing sequence s, we assign probabilities $p_1, \ldots, p_k$ 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 $\mathrm{Ops}_s(D, \Sigma)$ for the set of (D, Σ)-repairing sequences $\{s' \in \mathrm{RS}(D, \Sigma) \mid s' = s \cdot op \text{ for some } D\text{-operation } op\}$.

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 $T = (V, E, \mathbf{P})$, where $V = \mathrm{RS}(D, \Sigma)$, $E \subseteq V \times V$ and $\mathbf{P} : E \to [0, 1]$, such that:

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

Let us clarify that, for a database D and a set Σ of FDs, we assume that a (D, Σ)-repairing Markov chain $(V, E, \mathbf{P})$ is compactly represented as a function $f : \mathrm{RS}(D, \Sigma) \times \mathrm{RS}(D, \Sigma) \to [0, 1] \cup \{\bot\}$ such that, for every pair $(s, s') \in \mathrm{RS}(D, \Sigma) \times \mathrm{RS}(D, \Sigma)$, $f(s, s') = \mathbf{P}(s, s')$ if $(s, s') \in E$; otherwise, $f(s, s') = \bot$. 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 $\mathbf{S} = \{R/3\}$, where

$$f_1 = R(a_1, b_1, c_1) \qquad f_2 = R(a_1, b_2, c_2) \qquad f_3 = R(a_2, b_1, c_2).$$

Consider also the set $\Sigma = \{\phi_1, \phi_2\}$ of FDs over $\mathbf{S}$, where

$$\phi_1 = R : A \to B \qquad \text{and} \qquad \phi_2 = R : C \to B,$$

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 $\mathsf{V}(D, \Sigma) = \{(\phi_1, \{f_1, f_2\}), (\phi_2, \{f_2, f_3\})\}$. Observe that for the edge-labeled rooted tree $T = (V, E, \mathbf{P})$ in Figure 1, $V = \mathrm{RS}(D, \Sigma)$, for a non-leaf node s the set of its children is $\mathrm{Ops}_s(D, \Sigma)$, and the set of leaves coincides with $\mathrm{CRS}(D, \Sigma)$. 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 operational approach to CQA also provides a mechanism that allows us to define a family of repairing Markov chains independently of the input database. In other words, whenever the input database D changes, the desired repairing Markov chain should be simply extracted from D via a procedure (parameterized by the underlying set Σ of FDs) that takes as input D. This mechanism is formalized via the notion of repairing Markov chain generator.

Definition 3.7 (Repairing Markov Chain Generator). A *repairing Markov chain generator* w.r.t. a set Σ of FDs is a function M_Σ assigning to every database D a (D, Σ)-repairing Markov chain.

Coming back to our running example, given a repairing Markov chain generator M_Σ w.r.t. Σ, $M_\Sigma(D)$ is essentially a rooted tree as the one in Figure 1 such that $p_1 + p_2 + p_3 + p_4 + p_5 = 1$, $p_6 + p_7 + p_8 = 1$ and $p_9 + p_{10} + p_{11} = 1$. Thus, given two different repairing Markov chain generators M_Σ and M'_Σ w.r.t. Σ, $M_\Sigma(D)$ and $M'_\Sigma(D)$ are both rooted trees as in Figure 1, but with different labeling functions, i.e., they calculate differently the probabilities that label the edges.

Now, an operational repair is a candidate operational repair 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 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), ..., (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, $\pi(s) = \mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \cdots \cdot \mathbf{P}(s_{n-1}, s_n)$. The set of *reachable leaves* of T, denoted $\mathsf{RL}(T)$, is the set of leaves of T that have non-zero probability according to the leaf distribution of T.

Definition 3.8 (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 $D' \in \mathsf{CORep}(D, \Sigma)$ with $D' = s(D)$ for $s \in \mathsf{RL}(M_\Sigma(D))$. Let $\mathsf{ORep}(D, M_\Sigma)$ be the set of all repairs of D w.r.t. M_Σ.

At this point, it should be clear that an operational repair is a consistent subset of the database that need not be maximal, unlike a classical database repair as defined in [Arenas et al. 1999]. Let us stress that the deviation of the operational approach to consistent query answering, as defined in [Calautti et al. 2018], from the principle of applying minimal changes during the repairing process was a deliberate design choice. The rationale underlying this choice is that, although the principle that a repair should preserve as much information as possible perfectly applies to applications where the data is the most valuable asset, there are applications where it is important to remove as much inconsistency as possible at the expense of loosing more data during the repairing process. For example, such an application is healthcare, where highly critical decisions are made based on the available data, and thus, we cannot afford to make decisions based on conflicting data. Consequently, since the participation of a fact in a conflict is a strong indication that it might carry wrong information, for such applications it would be preferable to deviate from the principle of maximal repairs, and remove as many facts that participate in a conflict as possible. This, in turn, allows us to provide even more reliable answers to queries. The operational approach to CQA allows us to specify the preferred repairs between the two extremes discussed above (i.e., subset maximal repairs vs. repairs that minimize the uncertainty) by adopting the respective Markov chain generator. The above discussion is illustrated by the following simple example.

Example 3.9. Consider the schema $\mathbf{S}$ mentioning only the relation Patients with attributes

$$(\text{Id}, \text{Name}, \text{Age}, \text{InsuranceLevel}, \text{BloodType}, \text{BloodTypeForTransfusion}).$$

Each tuple in Patients stores the id of a patient, together with his/her name, age, insurance level, blood type, as well as the blood type that has been chosen for a planned blood transfusion of the

patient. Assume now that $\Sigma = \{\phi_1, \phi_2, \phi_3\}$ is the set of functional dependencies over **S** with

$$\begin{aligned} \phi_1 &= \text{Patients} : \text{Id} \rightarrow \text{Name}, \\ \phi_2 &= \text{Patients} : \text{Age} \rightarrow \text{InsuranceLevel}, \\ \phi_3 &= \text{Patients} : \text{BloodType} \rightarrow \text{BloodTypeForTransfusion}. \end{aligned}$$

The FD ϕ_1 states that the patient name is uniquely determined by his/her id, while ϕ_2 specifies that the insurance level of the patient depends on how old the patient is. Finally, ϕ_3 specifies that the blood type determines which blood type must be used for the transfusion. Consider now the following database D over **S**, which is inconsistent w.r.t. Σ (attribute names are shortened):

Id	Name	Age	IL	BT	BTFT	
1	James	38	High	AB^-	O^-	(f_1)
1	Andrea	36	Medium	O^+	O^+	(f_2)
2	Mary	38	Low	A^-	A^-	(f_3)
3	Laura	18	Medium	AB^-	O^+	(f_4)

For each tuple $\bar{t}$, we give the name of the corresponding fact f_i of the form Patients($\bar{t}$). The set $\mathsf{V}(D, \Sigma)$ is $\{(\phi_1, \{f_1, f_2\}), (\phi_2, \{f_1, f_3\}), (\phi_3, \{f_1, f_4\})\}$, that is, f_1 and f_2 are in a conflict because of ϕ_1, f_1 and f_3 are in a conflict because of ϕ_2, and f_1 and f_4 are in a conflict because of ϕ_3.

Towards constructing a consistent subset of D, one might consider removing the fact f_1. This can be done for multiple reasons, e.g., because we are uncertain who the patient with id 1 is (ϕ_1), or because we are uncertain whether people at age 38 should have a "High" or a "Low" insurance level (ϕ_2), or even further, because we are not sure whether for blood type AB^-, blood type O^- or O^+ is the one to be used for the transfusion (ϕ_3). No matter the reason, assume we consider a repair where f_1 is removed. Hence, there exists only one maximally consistent subset of D w.r.t. Σ, i.e., a classical database repair as defined in [Arenas et al. 1999], which does not contain f_1: the database $D' = \{f_2, f_3, f_4\}$. In particular, note how the maximality of D' forces us to keep f_3 and f_4. However, being forced to keep f_3 and f_4 means that we are committing to a combination of possibly wrong data, i.e., patients at age 38 only deserve a low insurance level *and* the patient Laura will have a blood transfusion using O^+ as the blood type. The amount of such possibly wrong data could increase as the number of FDs and attributes increase in the relation. Our framework is flexible enough to consider also other consistent subsets of the database as repairs that go beyond the maximally consistent ones of [Arenas et al. 1999]. This allows to take into account other scenarios where less uncertainty is preferred in the resulting repair. For example, the candidate operational repair $D'' = \{f_2, f_3\}$ describes a scenario where not only f_1 is assumed to be wrong, but also f_4.

Now, 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 such that $D' = s(D)$.

Definition 3.10 (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

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

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 $[\![D]\!]_{M_\Sigma} = \{(D', \mathsf{P}_{D,M_\Sigma}(D')) \mid D' \in \mathsf{ORep}(D, M_\Sigma)\}$.

Table 1. Basic Notation of Operational CQA

$\mathsf{RS}(D, \Sigma)$	(D, Σ)-repairing sequences
$\mathsf{CRS}(D, \Sigma)$	Complete (D, Σ)-repairing sequences s – i.e., $s(D) \models \Sigma$
$\mathsf{RL}(M_\Sigma(D))$	Reachable leaves s of $M_\Sigma(D)$ – i.e., $s \in \mathsf{CRS}(D, \Sigma)$ with $\pi(s) > 0$
$\mathsf{CORep}(D, \Sigma)$	Candidate repairs D' of D w.r.t. Σ – i.e., $D' = s(D)$ for some $s \in \mathsf{CRS}(D, \Sigma)$
$\mathsf{ORep}(D, M_\Sigma)$	Operational repairs D' of D w.r.t. M_Σ – i.e., $D' \in \mathsf{CORep}(D, \Sigma)$ such that $D' = s(D)$ for some $s \in \mathsf{RL}(M_\Sigma(D))$

Operational CQA. We now have all the necessary notions to recall the operational approach to CQA 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, the probability of $\bar{c}$ being an answer to Q over some operational repair of D is defined as

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

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 $\left\{\left(\bar{c}, \mathsf{P}_{M_\Sigma, Q}(D, \bar{c})\right) \mid \bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}\right\}$. 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, and asks for the probability $\mathsf{P}_{M_\Sigma, Q}(D, \bar{c})$. 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 :	$\mathsf{OCQA}(\Sigma, M_\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$.
OUTPUT :	$\mathsf{P}_{M_\Sigma, Q}(D, \bar{c})$.

The non-standard notation introduced above is summarized in Table 1. Until now, a repairing Markov chain generator is a general function from databases to repairing Markov chains. We now discuss 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 the set of candidate operational repairs, the set of complete repairing sequences, and 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:

– $\mathsf{ORep}(D, M_\Sigma^{\mathsf{ur}}) = \mathsf{CORep}(D, \Sigma)$, and for each $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{ur}})$,

$$\mathsf{P}_{D, M_\Sigma^{\mathsf{ur}}}(D') = \frac{1}{|\mathsf{ORep}(D, M_\Sigma^{\mathsf{ur}})|}.$$

– For every $s \in \mathsf{CRS}(D, \Sigma)$, assuming that π is the leaf distribution of $M_\Sigma^{\mathsf{us}}(D)$,

$$\pi(s) = \frac{1}{|\mathsf{CRS}(D, \Sigma)|}.$$

— For every $s, s' \in \mathsf{RS}(D, \Sigma)$, assuming that $M^{\mathsf{uo}}_{\Sigma}(D) = (V, E, \mathbf{P})$, $s' \in \mathsf{Ops}_s(D, \Sigma)$ implies

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

Before we give the formal definitions of the above Markov chain generators, let us first briefly discuss their conceptual relevance. As said in Section 3, candidate operational repairs are consistent subsets of the database that need not be maximal, which in essence means that the operational approach to CQA allows for the selection of candidate repairs that fall between the two extremes of subset maximal repairs and repairs that minimize the uncertainty. Now, the task of classifying some candidate operational repairs as more relevant than others is left to the Markov chain generator, while the choice of the Markov chain generator is application-centric. The three Markov chain generators discussed above adhere to the following principles: (i) M^{ur}_{Σ} does not express any particular preference, (ii) M^{us}_{Σ} gives priority to repairs with less atoms that may carry incorrect information, and (iii) M^{uo}_{Σ} favours repairs with minimal change (i.e., the opposite of M^{us}_{Σ}). Let us elaborate a bit more on those Markov chain generators:

Uniform Repairs. As said above, not removing a fact during the repairing process for the sake of keeping as much information as possible in an operational repair could be undesirable for certain critical applications. Consider, for example, the database $\{f, g_1, \ldots, g_n\}$, and assume that the facts $g_1, \ldots, g_n$ are all in conflict with f. This is a strong indication that all those facts might contain incorrect information, and thus, there is no deep reason to believe that *all* facts $g_1, \ldots, g_n$ are absolutely correct simply because we removed f during the repairing process. Under this perspective, it is natural to consider any consistent subset of the database as a conceptually valid repair in which both information and facts that may contain incorrect information are preserved to some extent. The Markov chain generator M^{ur}_{Σ} gives no preference to any of those candidate operational repairs.

Uniform Sequences. Differently from M^{ur}_{Σ}, the Markov chain generator M^{us}_{Σ} instead favours candidate operational repairs in which the number of atoms that may contain incorrect information is reduced as much as possible. Under this perspective, the empty repair, or, more generally, the database containing only isolated nodes in the underlying conflict graph, which contains an edge among conflicting facts, is the most desirable candidate operational repair in the sense that no fact that may carry erroneous information is left at the expense of sacrificing more data than other candidate operational repairs. This is actually achieved by taking into account the repairing process that leads to the candidate operational repairs. In other words, M^{us}_{Σ} assigns probabilities to candidate operational repairs by taking into account the support in terms of the repairing process. This is completely neglected by M^{ur}_{Σ} since a candidate operational repair obtained by very few repairing sequences is considered equally important as a candidate operational repair obtained by several repairing sequences. This is illustrated by the following example that builds on Example 3.9.

Example 4.1. Consider the database D and the set Σ of FDs from Example 3.9. The set of all complete (D, Σ)-repairing sequences $\mathsf{CRS}(D, \Sigma)$ contains the sequence $-f_1$, the sequences $-\{f_1, f_i\}$ with $i \in \{2, 3, 4\}$ and, for each permutation (i, j, k) of the numbers in $\{2, 3, 4\}$, the sequences of the form $-f_i, -f_j, -f_k$, of the form $-f_i, -\{f_1, f_j\}$, and of the form $-f_i, -f_j, -\{f_1, f_k\}$. In total, $\mathsf{CRS}(D, \Sigma)$ contains twenty two (D, Σ)-repairing sequences. The above sequences together lead to nine different candidate operational repairs, that is, the databases $D_1 = \emptyset$, $D_2 = \{f_1\}$, $D_3 = \{f_2\}$, $D_4 = \{f_3\}$, $D_5 = \{f_4\}$, $D_6 = \{f_2, f_3\}$, $D_7 = \{f_2, f_4\}$, $D_8 = \{f_3, f_4\}$, and $D_9 = \{f_2, f_3, f_4\}$. The sequences leading to the empty repair D_1 are the ones with three operations where the last one removes a pair of facts. There are six such sequences, and according to the repairing Markov chain generator M^{us}_{Σ},

the probability of D_1 is $6/22 \approx 27\%$, which is the highest probability among the probabilities of candidate operational repairs, according to M_Σ^{us}. Conversely, the repair D_9, which contains all facts except for f_1, is obtained only via the sequence $-f_1$, and thus, its probability is $1/22 \approx 4.5\%$, which is the lowest probability among the probabilities of the candidate operational repairs. Hence, in the case of the database D and set Σ of FDs of Example 3.9 dealing with the health domain, the repairing Markov chain generator M_Σ^{us} can be seen as an appropriate choice since it prioritizes repairs where more facts participating in a conflict are removed.

Uniform Operations. The Markov chain M_Σ^{uo}, as said above, should be understood as the opposite of M_Σ^{us} in the sense that it favours candidate operational repairs that keep as much information as possible. This is achieved by considering some repairing sequences more important than others depending on their structure. The key structural property here is the length of a sequence, which in turn translates into the number of removals applied to the given database. Since the intention of M_Σ^{uo} is to keep as much information as possible, it classifies shorter sequences as more important than longer ones. In essence, M_Σ^{uo} assigns probabilities to repairs by consolidating two parameters: the number of sequences that lead to a repair and the number of modifications applied to the given database; this is reflected in Example 3.6 and it is further illustrated by the following example.

Example 4.2. Consider again the database D and the set Σ of FDs from Example 3.9, and recall the (D, Σ)-repairing sequences and candidate operational repairs discussed in Example 4.1. We first focus on the empty repair D_1. For this, recall that the only (D, Σ)-repairing sequences leading to D_1 are the ones of the form $-f_i, -f_j, -\{f_1, f_k\}$, with i, j, k all distinct and different from 1. All such sequences have the same probability, according to M_Σ^{uo}, which we discuss below. Observe that there are seven possible (D, Σ)-justified operations, i.e., removing either one of the four facts in D, or one of the pairs $\{f_1, f_2\}, \{f_1, f_3\}, \{f_1, f_4\}$. After removing a single fact f_i different from f_1, we obtain the database $D' = D \setminus \{f_i\}$, and thus, we are left with five (D', Σ)-justified operations, that is, remove one of the three facts left in D', or one of the two pairs $\{f_1, f_j\}$, with $j \neq 1$ and $j \neq i$. After removing again a single fact f_j from D', we obtain the database D'' containing only two facts, one of which is f_1. Hence, we are only left with three (D'', Σ)-justified operations: remove one of the two facts or both. We conclude that the probability of each sequence of the form $-f_i, -f_j, -\{f_1, f_k\}$, with i, j, k all distinct and different from 1 is $1/7 \times 1/5 \times 1/3 = 1/105$. Since there are six such sequences, the probability of D_1 is $6/105 \approx 5.7\%$, according to M_Σ^{uo}, which is the lowest among the probabilities of the candidate operational repairs according to M_Σ^{uo}. If we consider instead the candidate operational repair $D_9 = \{f_2, f_3, f_4\}$, which contains the largest amount of facts from D, $-f_1$ is the only sequence leading to D_9, whose probability is $1/7$, and thus, the probability of D_9 is $1/7 \approx 14.2\%$, which is the largest probability among the probabilities of the candidate operational repairs, according to M_Σ^{uo}. Hence, in the case of the database D and set Σ of FDs of Example 3.9 dealing with the health domain, the repairing Markov chain generator M_Σ^{uo} might not be appropriate as it prioritizes repairs where the highest number of facts is kept, potentially carrying wrong information, such as the wrong blood type to be used for a transfusion.

We now provide the formal definitions of the "uniform" repairing Markov chain generators discussed above, and show that they indeed capture our intention. In what follows, for brevity, given a database D, a set Σ of FDs, and a sequence $s = op_1, \ldots, op_n \in \mathsf{RS}(D, \Sigma)$, we write s_0 for the empty sequence ε, and s_i for the sequence $op_1, \ldots, op_i$, for each $i \in [n]$.

4.1 Uniform Repairs

We start with the Markov chain generator based on the uniform probability distribution over the set of candidate operational repairs. Since multiple complete sequences can lead to the same database

(see, e.g., $-f_1, -\{f_2, f_3\}$ and $-f_3, -\{f_1, f_2\}$ in Example 3.6), 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. Given a database D and a set Σ of FDs, we say that a (D, Σ)-repairing sequence $s \in \mathsf{CRS}(D, \Sigma)$ is *canonical* if there is no $s' \in \mathsf{CRS}(D, \Sigma)$ such that $s(D) = s'(D)$ and $s' \prec s$ for some arbitrary ordering $\prec$ over the set $\mathsf{RS}(D, \Sigma)$. Let $\mathsf{CanCRS}(D, \Sigma)$ be the set of all sequences of $\mathsf{CRS}(D, \Sigma)$ that are canonical. Furthermore, for a sequence $s \in \mathsf{RS}(D, \Sigma)$, we write $\mathsf{CanCRS}_s(D, \Sigma)$ for the set of all sequences s' of $\mathsf{CanCRS}(D, \Sigma)$ that have s as a prefix, i.e., $s' = s \cdot s''$ for some (possibly empty) sequence s''. Hence, for $s \in \mathsf{RS}(D, \Sigma)$, $\mathsf{CanCRS}_s(D, \Sigma)$ is the set of canonical leaves of the subtree rooted at s, with $\mathsf{CanCRS}_\epsilon(D, \Sigma) = \mathsf{CanCRS}(D, \Sigma)$ being the set of canonical leaves of the tree. We are now ready to define the desired repairing Markov chain generator.

Definition 4.3 (Repairing Markov Chain Generator Based on Uniform Repairs). Consider a set Σ of FDs. Let M^{ur}_Σ be the function assigning to a database D the (D, Σ)-repairing Markov chain $(V, E, \mathbf{P})$, where, for each $(s, s') \in E$,

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

Notice that the ratio $\frac{|\mathsf{CanCRS}_{s'}(D,\Sigma)|}{|\mathsf{CanCRS}_s(D,\Sigma)|}$ is not defined if the subtree T_s rooted at s has no canonical leaves, that is, $\mathsf{CanCRS}_s(D, \Sigma) = \emptyset$. 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 as long as the sum of probabilities equals one; we have chosen $\frac{1}{|\mathsf{Ops}_s(D,\Sigma)|}$. The above Markov chain generator is well-defined since, for each $s \in \mathsf{RS}(D, \Sigma)$ that is not complete,

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

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

Proposition 4.4. *Consider a set Σ of FDs. For every database D:*

(1) $\mathsf{ORep}(D, M^{\mathsf{ur}}_\Sigma) = \mathsf{CORep}(D, \Sigma)$.

(2) For every $D' \in \mathsf{ORep}(D, M^{\mathsf{ur}}_\Sigma)$, $\mathsf{P}_{D,M^{\mathsf{ur}}_\Sigma}(D') = \frac{1}{|\mathsf{ORep}(D,M^{\mathsf{ur}}_\Sigma)|}$.

Proof. **Item (1).** It suffices to prove that $\mathsf{RL}(M^{\mathsf{ur}}_\Sigma(D)) = \mathsf{CanCRS}(D, \Sigma)$. Let $M^{\mathsf{ur}}_\Sigma(D) = (V, E, \mathbf{P})$, and assume that π is its leaf distribution. Recall that for a sequence $s = op_1, \ldots, op_n \in \mathsf{CRS}(D, \Sigma)$, $\pi(s) = \mathbf{P}(s_0, s_1) \times \cdots \times \mathbf{P}(s_{n-1}, s_n)$ assuming that (s_0, s_1), (s_1, s_2), ..., (s_{n-1}, s_n), where $n \geq 0$, $\varepsilon = s_0$ and $s = s_n$, is the unique path in $M^{\mathsf{ur}}_\Sigma(D)$ from ε to s.

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

($\subseteq$) Assume now that $s = op_1, \ldots, op_n \in \mathsf{RL}(M^{\mathsf{ur}}_\Sigma(D))$. By contradiction, assume that $s \notin \mathsf{CanCRS}(D, \Sigma)$. Since $s \in \mathsf{RL}(M^{\mathsf{ur}}_\Sigma(D))$, s must be complete. The fact that s is complete but not canonical implies that there exists $i \in \{0, \ldots, n\}$ such that $\mathsf{CanCRS}_{s_i}(D, \Sigma) = \emptyset$. In particular, let ℓ be the smallest integer in $\{0, 1, \ldots, n\}$ such that $\mathsf{CanCRS}_{s_\ell}(D, \Sigma) = \emptyset$. Clearly, $\ell > 0$ since $\mathsf{CanCRS}_\epsilon(D, \Sigma)$ is non-empty. Thus, by the first rule of the expression defining $\mathbf{P}$ in Definition 4.3, we have that $\mathbf{P}(s_{\ell-1}, s_\ell) = 0$. Hence, $\pi(s) = 0$, and thus, $s \notin \mathsf{RL}(M^{\mathsf{ur}}_\Sigma(D))$, which is a contradiction.

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

$$|\mathsf{ORep}(D, M^{\mathsf{ur}}_{\Sigma})| \;=\; |\mathsf{CanCRS}(D,\Sigma)| \;=\; |\mathsf{RL}(M^{\mathsf{ur}}_{\Sigma}(D))|.$$

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

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

Since $\mathsf{CanCRS}_{s_0}(D,\Sigma) = \mathsf{CanCRS}_{\epsilon}(D,\Sigma) = \mathsf{CanCRS}(D,\Sigma)$ and $\mathsf{CanCRS}_{s_n}(D,\Sigma) = \{s_n\}$, we get that $\pi(s) = \frac{1}{|\mathsf{CanCRS}(D,\Sigma)|}$ and the claim follows. □

Here is an example that illustrates repairing Markov chain generators based on uniform repairs.

Example 4.5. 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, illustrating what the repairing Markov chain generator M^{ur}_{Σ} does boils down to explaining how the probabilities $p_1, \ldots, p_{11}$ are calculated by M^{ur}_{Σ}. Assuming that for $s, s' \in \mathsf{RS}(D,\Sigma)$, $s \prec s'$ iff s comes before s' in a depth-first traversal of the tree depicted in Figure 1, we have that $\mathsf{CanCRS}(D,\Sigma)$ consists of the sequences

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

Therefore, we get that

$$|\mathsf{CanCRS}_{\epsilon}(D,\Sigma)| \;=\; 5 \qquad |\mathsf{CanCRS}_{-f_1}(D,\Sigma)| \;=\; 3$$
$$|\mathsf{CanCRS}_{-f_2}(D,\Sigma)| \;=\; |\mathsf{CanCRS}_{-\{f_2,f_3\}}(D,\Sigma)| \;=\; 1$$
$$|\mathsf{CanCRS}_{-\{f_1,f_2\}}(D,\Sigma)| \;=\; |\mathsf{CanCRS}_{-f_3}(D,\Sigma)| \;=\; 0.$$

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, $\mathsf{RL}(M^{\mathsf{ur}}_{\Sigma}(D)) = \mathsf{CanCRS}(D,\Sigma)$ and $\pi(s) = \frac{1}{5}$, for $s \in \mathsf{RL}(M^{\mathsf{ur}}_{\Sigma}(D))$, with π being the leaf distribution of $M^{\mathsf{ur}}_{\Sigma}(D)$. Hence, $\mathsf{ORep}(D, M^{\mathsf{ur}}_{\Sigma}) = \{\emptyset, \{f_1\}, \{f_2\}, \{f_3\}, \{f_1, f_3\}\}$ with $\mathsf{P}_{D,M_{\Sigma}}(D') = \frac{1}{5}$, for $D' \in \mathsf{ORep}(D, M^{\mathsf{ur}}_{\Sigma})$.

4.2 Uniform Sequences

We now proceed to define the repairing 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 complete sequences.

Definition 4.6 (Repairing Markov Chain Generator Based on Uniform Sequences). Consider a set Σ of FDs. Let M^{us}_{Σ} be the function assigning to a database D the (D,Σ)-repairing Markov chain $(V, E, \mathbf{P})$, where, for each $(s, s') \in E$,

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

The above Markov chain generator is well-defined since, for $s \in \mathsf{RS}(D,\Sigma)$ that is not complete,

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

and thus, for a non-leaf node $s \in V$, $\sum_{t \in \{s' \mid (s,s') \in E\}} \mathbf{P}(s,t) = 1$. Indeed, M^{us}_{Σ} captures our intention:

Proposition 4.7. *Consider a set Σ of FDs. For every database D:*
(1) $\mathsf{RL}(M^{\mathsf{us}}_\Sigma(D)) = \mathsf{CRS}(D,\Sigma)$.
(2) For every $s \in \mathsf{CRS}(D,\Sigma)$, assuming that π is the leaf distribution of $M^{\mathsf{us}}_\Sigma(D)$, $\pi(s) = \frac{1}{|\mathsf{CRS}(D,\Sigma)|}$.

Proof. **Item (1).** This item follows from the fact that each $s \in \mathsf{RL}(M^{\mathsf{us}}_\Sigma(D))$ is complete by definition, and each $s = op_1, \ldots, op_n \in \mathsf{CRS}(D,\Sigma)$ is such that $\mathsf{CRS}_{s_i}(D,\Sigma) \neq \emptyset$, for $i \in \{0,\ldots,n\}$, and thus $\pi(s) > 0$, where π is the leaf distribution of $M^{\mathsf{us}}_\Sigma(D)$.
Item (2). It is shown via a proof similar to the one of item (2) of Proposition 4.4. □

Here is an example of a repairing Markov chain generator based on uniform sequences.

Example 4.8. Let D and Σ be the database and the set of FDs, respectively, from Example 3.6. We proceed to explain how the probabilities $p_1, \ldots, p_{11}$ that label the edges of the tree depicted in Figure 1 are calculated by M^{us}_Σ. Observe that

$$|\mathsf{CRS}_\epsilon(D,\Sigma)| = 9$$
$$|\mathsf{CRS}_{-f_1}(D,\Sigma)| = |\mathsf{CRS}_{-f_3}(D,\Sigma)| = 3$$
$$|\mathsf{CRS}_{-\{f_1,f_2\}}(D,\Sigma)| = |\mathsf{CRS}_{-f_2}(D,\Sigma)| = |\mathsf{CRS}_{-\{f_2,f_3\}}(D,\Sigma)| = 1.$$

Hence, $p_1 = p_5 = \frac{3}{9}$, $p_2 = p_3 = p_4 = \frac{1}{9}$, $p_6 = p_7 = p_8 = \frac{1}{3}$, and $p_9 = p_{10} = p_{11} = \frac{1}{3}$. Thus, $\mathsf{RL}(M^{\mathsf{us}}_\Sigma(D)) = \mathsf{CRS}(D,\Sigma)$ and $\pi(s) = \frac{1}{9}$, for $s \in \mathsf{RL}(M^{\mathsf{us}}_\Sigma(D))$; π is the leaf distribution of $M^{\mathsf{us}}_\Sigma(D)$.

4.3 Uniform Operations

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

Definition 4.9 (Repairing Markov Chain Generator Based on Uniform Operations). Consider a set Σ of FDs. Let M^{uo}_Σ be the function assigning to a database D the (D,Σ)-repairing Markov chain $(V, E, \mathbf{P})$, where, for each $(s, s') \in E$,

$$\mathbf{P}(s,s') = \frac{1}{|\mathsf{Ops}_s(D,\Sigma)|}$$

It is easy to see that M^{uo}_Σ captures our intention; in fact, the following holds by definition:

Proposition 4.10. *Consider a set Σ of FDs. For every database D:*
(1) $\mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D)) = \mathsf{CRS}(D,\Sigma)$.
(2) Assuming that $M^{\mathsf{uo}}_\Sigma(D) = (V,E,\mathbf{P})$, $(s,s') \in E$ implies $\mathbf{P}(s,s') = \frac{1}{|\mathsf{Ops}_s(D,\Sigma)|}$.

Here is an example of a repairing Markov chain generator based on uniform operations.

Example 4.11. Let D and Σ be the database and the set of FDs, respectively, from Example 3.6. We proceed to explain how the probabilities $p_1, \ldots, p_{11}$ that label the edges of the tree depicted in Figure 1 are calculated by M^{uo}_Σ. It is clear that $p_1 = p_2 = p_3 = p_4 = p_5 = \frac{1}{5}$, $p_6 = p_7 = p_8 = \frac{1}{3}$, and $p_9 = p_{10} = p_{11} = \frac{1}{3}$. Thus, both items of Proposition 4.10 are satisfied, as needed.

Notice that, unlike the repairing Markov chain generators M^{ur}_Σ and M^{us}_Σ discussed above, M^{uo}_Σ 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^{uo}_Σ has a significant impact on operational CQA when it comes to approximations.

At this point, we would like to stress that there is an interesting connection between M^{uo}_Σ and the notion of *repair-key*, a construct introduced in [Koch 2008] towards a query algebra for probabilistic

databases. It turns out that one can achieve the same as M^{uo}_{Σ} by exploiting the repair-key construct. However, this is not useful for our main objective, which is to analyze the complexity of uniform operational CQA, since we cannot inherit any (in)approximability results from the repair-key literature. Concerning positive results, we can only inherit an additive error approximation, which we easily get from our results due to the existence of an efficient sampler, but not a relative error approximation, which is the main target, since the main task underlying repair-keys does to admit one. Concerning negative results, the non-existence of a relative error approximation for repair-keys is shown via the NP-hardness of the underlying decision problem, while in our case the underlying decision problem (i.e., checking whether the probability of a tuple is positive) when we consider M^{uo}_{Σ} is tractable. Thus, an approximation preserving reduction from the main problem for repair-keys to our problem would imply NP = BPP, which is unlikely to be the case.

4.4 Our Main Objective

The data complexity of OCQA for *arbitrary* Markov chain generators has been already studied in [Calautti et al. 2018], showing that it is, in general, intractable. In particular:

Theorem 4.12 ([Calautti et al. 2018]). *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* $\mathsf{OCQA}(\Sigma, M_\Sigma, Q)$ *is* $\sharp$P*-hard.*

The above result in [Calautti et al. 2018] is stated for arbitrary keys, but a simple inspection of the underlying proof reveals that it holds even for primary keys. Let us also stress that this result is not directly applicable to the Markov chain generators considered in this article; hence, we provide an independent proof of $\sharp$P-hardness. With the above intractability result in place, the authors of [Calautti et al. 2018] asked whether $\mathsf{OCQA}(\Sigma, M_\Sigma, Q(\bar{x}))$ is approximable, i.e., whether the target probability can be approximated via a FPRAS. An FPRAS for $\mathsf{OCQA}(\Sigma, M_\Sigma, Q(\bar{x}))$ is a randomized algorithm A that takes as input a database D, a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, $\epsilon > 0$, and $0 < \delta < 1$, runs in polynomial time in $\|D\|$, $\|\bar{c}\|$,[1] $1/\epsilon$ and $\log(1/\delta)$, and produces a random variable $\mathsf{A}(D, \bar{c}, \epsilon, \delta)$ such that

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

It was shown that the problem in question does not admit an FPRAS, under the widely accepted complexity assumption that RP $\neq$ 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") [Arora and Barak 2009].

Theorem 4.13 ([Calautti et al. 2018]). *Unless* RP = 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* $\mathsf{OCQA}(\Sigma, M_\Sigma, Q)$.

Again, the above result in [Calautti et al. 2018] is stated for arbitrary keys, but it was actually shown for primary keys. 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 4.12 and 4.13 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 can be summarized as follows (unless stated otherwise, the results hold for all three Markov chain generators):

(1) The complexity of exact operational CQA remains $\sharp$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.

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

Table 2. The Complexity of (Exact ; Approximate) OCQA for the Markov Chain Generators M^{ur}_Σ, M^{us}_Σ and M^{uo}_Σ

	Primary Keys	Keys	Functional Dependencies
$\mathrm{OCQA}(\Sigma, M^{\mathsf{ur}}_\Sigma, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; ?	♯P-hard ; no FPRAS
$\mathrm{OCQA}(\Sigma, M^{\mathsf{us}}_\Sigma, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; ?	♯P-hard ; ?
$\mathrm{OCQA}(\Sigma, M^{\mathsf{uo}}_\Sigma, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; FPRAS	♯P-hard ; FPRAS[1] and ?

By FPRAS we mean that the problem in question admits and FPRAS. By FPRAS[1] we mean that it admits an FPRAS assuming that only operations that remove a single fact (not a pair of facts) are considered, whereas no FPRAS means that it does not admit an FPRAS (unless RP = NP). The symbol ? refers to the fact that it remains open whether the problem in question admits an FPRAS in general.

(3) In the case of arbitrary keys and FDs, although the repairing 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 article is devoted to discussing the above results that are summarized in Table 2.

5 Uniform Repairs

We start our complexity analysis by considering the repairing Markov chain generator based on uniform repairs and establish the next result:

Theorem 5.1. *The following hold:*

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

Notice that item (3) 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 a strong indication that item (3) holds even in the case of arbitrary keys. We now discuss how the three items of Theorem 5.1 are 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$,

$$\mathrm{P}_{M^{\mathsf{ur}}_\Sigma, Q}(D, \bar{c}) \quad = \quad \frac{|\{D' \in \mathrm{CORep}(D, \Sigma) \mid \bar{c} \in Q(D')\}|}{|\mathrm{CORep}(D, \Sigma)|}.$$

This ratio is the percentage of candidate operational repairs of D w.r.t. Σ that entail $Q(\bar{c})$, called *repair relative frequency* of $Q(\bar{c})$ w.r.t. D and Σ, denoted $\mathrm{repfreq}_{\Sigma,Q}(D, \bar{c})$. Therefore, we can conveniently restate the problem $\mathrm{OCQA}(\Sigma, M^{\mathsf{ur}}_\Sigma, Q)$ 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 repairing Markov chain generator M^{ur}_Σ:

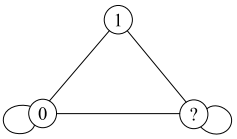
PROBLEM :	$\mathsf{RepFreq}(\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$.
OUTPUT :	$\mathrm{repfreq}_{\Sigma,Q}(D, \bar{c})$.

We discuss the proof of Theorem 5.1 by considering $\mathsf{RepFreq}(\Sigma, Q)$ instead of $\mathrm{OCQA}(\Sigma, M^{\mathsf{ur}}_\Sigma, Q)$.

[2]In 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 8.

5.1 Proof of Item (1) of Theorem 5.1

We show that $\mathsf{RepFreq}(\Sigma, Q)$ is $\sharp$P-hard for a set Σ consisting of a single key and a Boolean CQ Q. This is done via a polynomial-time Turing reduction from a counting graph-theoretic problem. 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 undirected graph



Given an undirected graph $G = (V_G, E_G)$, a homomorphism from G to H is a function $h : V_G \to V_H$ such that $\{u, v\} \in E_G$ implies $\{h(u), h(v)\} \in E_H$. We write $\mathsf{hom}(G, H)$ for the set of homomorphisms from G to H. We consider the problem

PROBLEM :	$\sharp H$-Coloring
INPUT :	An undirected graph G.
OUTPUT :	The number $\lvert\mathsf{hom}(G, H)\rvert$.

It is implicit in [Dyer and Greenhill 2000] that $\sharp H$-Coloring is $\sharp$P-hard. In fact, [Dyer and Greenhill 2000] establishes the following dichotomy result: for an undirected graph $\hat{H}$, $\sharp\hat{H}$-Coloring is $\sharp$P-hard if $\hat{H}$ 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 H above consists of a single connected component which is neither a single node, nor a complete graph with all loops present ($(1, 1)$ is missing), nor a bipartite graph, we get that $\#H$-Coloring is indeed $\sharp$P-hard.

We proceed to show via a polynomial-time Turing reduction from $\sharp H$-Coloring that $\mathsf{RepFreq}(\Sigma, Q)$ is $\sharp$P-hard, where Σ and Q are as follows. Let $\mathbf{S} = \{V/2, E/2, T/1\}$ and (A, B) be the tuple of attributes of V. The set Σ consists of

$$V : A \to B$$

and the (constant-free) Boolean CQ Q is defined as

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

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

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

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

$$\mathsf{repfreq}_{\Sigma,Q}(D_G, ()) = \frac{\lvert\mathsf{CORep}(D_G, \Sigma, Q)\rvert}{\lvert\mathsf{CORep}(D_G, \Sigma)\rvert},$$

where $\mathsf{CORep}(D_G, \Sigma, Q)$ is the set of candidate repairs D of D_G w.r.t. Σ such that $D \models Q$; we use $()$ to denote the empty tuple. Observe that there are $3^{|V_G|}$ candidate repairs of D_G w.r.t. Σ, i.e., in

each such repair D, for each node $u \in V_G$, 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,

$$\mathsf{repfreq}_{\Sigma,Q}(D_G, ()) \;=\; \frac{|\mathsf{CORep}(D_G, \Sigma, Q)|}{3^{|V_G|}}.$$

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

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

Since D_G has $3^{|V_G|}$ candidate repairs w.r.t. Σ, we get that $\mathsf{HOM}(G)$ is precisely the cardinality of the set $\mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)$, 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 5.2. $|\mathsf{hom}(G, H)| \;=\; |\mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)|$.

Proof. It suffices to show that there exists a bijection from $\mathsf{hom}(G, H)$ to $\mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)$. To this end, we define the mapping $\mu : \mathsf{hom}(G, H) \to \mathcal{P}(D_G)$: for $h \in \mathsf{hom}(G, H)$,

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

We proceed to show the following three statements:

(1) μ is correct, i.e., it is indeed a function from $\mathsf{hom}(G, H)$ to $\mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)$.
(2) μ is injective.
(3) μ is surjective.

The mapping μ is correct. Consider an arbitrary homomorphism $h \in \mathsf{hom}(G, H)$. 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 $V_G = \{u_1, \ldots, u_n\}$. Consider the sequence $s_h = op_1, \ldots, op_n$ 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 $(1, 1)$ 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 $h, h' \in \mathsf{hom}(G, H)$ 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 $\mathsf{hom}(G, H)$. Therefore, for every two distinct homomorphisms $h, h' \in \mathsf{hom}(G, H)$, $\mu(h) \neq \mu(h')$, as needed.

The mapping μ is surjective. Consider a candidate repair $D \in \mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)$. We need to show that there is $h \in \mathsf{hom}(G, H)$ such that $\mu(h) = D$. We define the mapping $h_D : V_G \to V_H$ 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 $h_D \in \mathsf{hom}(G, H)$. 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 $h_D(v) = 1$. Thus, 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 $D \in \mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)$. □

Since $\mathsf{HOM}(G) = |\mathsf{CORep}(D_G, \Sigma) \setminus \mathsf{CORep}(D_G, \Sigma, Q)|$, Lemma 5.2 implies that $\mathsf{HOM}(G) = |\mathsf{hom}(G, H)|$, which shows that HOM is a polynomial-time Turing reduction from #H-Coloring to $\mathsf{RepFreq}(\Sigma, Q)$, and the claim follows. [3]

5.2 Proof of Item (2) of Theorem 5.1

We prove that, for a set Σ of primary keys and a CQ Q, the problem $\mathsf{RepFreq}(\Sigma, Q)$ admits an FPRAS. Our proof consists of two main steps, which we briefly explain before going into the detailed proofs. For further details on why the two steps below imply the existence of the desired FPRAS, we refer the reader to Appendix A.

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

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

  More precisely, $\mathsf{A}(D, \bar{c}, \epsilon, \delta)$ samples $N = O(\frac{\log(\frac{1}{\delta})}{\epsilon^2})$ elements of $\mathsf{CORep}(D, \Sigma)$, and returns $\frac{S}{N} \cdot |\mathsf{CORep}(D, \Sigma)|$, where S is the number of sampled repairs D' such that $\bar{c} \in Q(D')$.
- The existence of an efficient sampler does not guarantee the existence of an FPRAS, which bounds the *multiplicative (a.k.a. relative)* error. To obtain an FPRAS via Monte Carlo Sampling, the number of samples should be proportional to $\frac{1}{\mathsf{repfreq}_{\Sigma,Q}(D,\bar{c})}$ [Dagum et al. 2000]. This brings us to the second step of our proof, where we show that $\mathsf{repfreq}_{\Sigma,Q}(D, \bar{c})$ is never "too small". Formally, we show that, for every database D, either $\mathsf{repfreq}_{\Sigma,Q}(D, \bar{c}) = 0$ or $\mathsf{repfreq}_{\Sigma,Q}(D, \bar{c}) \geq \frac{1}{\mathsf{pol}(\|D\|)}$ for some polynomial function pol. In this case, we can use Monte Carlo Sampling (with a different, yet polynomial, number of samples) to obtain an FPRAS.

We proceed to formalize the above two steps.

[3] A recent result establishes that the problem is ♯P-hard even for self-join-free CQs of bounded generalized hypertreewidth [Calautti et al. 2024].

	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

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

Step 1: Efficient Sampler. The existence of an efficient sampler is established by the following lemma:

Lemma 5.3. *Given a database D and a set Σ of primary keys, we can sample elements of* $\mathsf{CORep}(D, \Sigma)$ *uniformly at random in polynomial time in $\|D\|$.*

Proof. For every relation name R of the underlying schema with a primary key $R : X \to Y$ 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, a candidate operational repair will contain, for every block B with $|B| > 1$, either a single fact of B or none of the facts of B; thus, there are $|B| + 1$ possible options. A candidate 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. A candidate 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 candidate operational repair. We denote all the blocks of D (over all the relations of the schema) that have at least two facts by $B_1, \ldots, B_n$. To sample a repair of $\mathsf{CORep}(D, \Sigma)$ we select, for each block B_i, one of its $|B_i| + 1$ possible outcomes with probability $\frac{1}{|B_i|+1}$. Then, a candidate 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. The probability of obtaining each candidate operational repair is $\frac{1}{|\mathsf{CORep}(D,\Sigma)|}$, as the number of candidate operational repairs is

$$|\mathsf{CORep}(D, \Sigma)| \;=\; (|B_1| + 1) \times \cdots \times (|B_n| + 1),$$

and the claim follows. □

We give a simple example that illustrates the proof of Lemma 5.3.

Example 5.4. 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 $\Sigma = \{R : A_1 \to A_2\}$ 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 the following three blocks w.r.t. $R : A_1 \to A_2$:

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

Since the fact $f_{2,1}$ is not involved in any violations of the constraint, it will appear in every candidate operational repair. However, every candidate operational repair will contain at most one fact of the first block and at most one fact of the third block. The number of candidate operational repairs

according to the formula in the proof of Lemma 5.3 is then $(3+1)\times(2+1)=12$. (Note that the blocks of size one are not considered in the computation.) Indeed, we have:

$$\begin{array}{cccc} \{f_{2,1}\} & \{f_{1,1},f_{2,1}\} & \{f_{1,2},f_{2,1}\} & \{f_{1,3},f_{2,1}\} \\ \{f_{2,1},f_{3,1}\} & \{f_{1,1},f_{2,1},f_{3,1}\} & \{f_{1,2},f_{2,1},f_{3,1}\} & \{f_{1,3},f_{2,1},f_{3,1}\} \\ \{f_{2,1},f_{3,2}\} & \{f_{1,1},f_{2,1},f_{3,2}\} & \{f_{1,2},f_{2,1},f_{3,2}\} & \{f_{1,3},f_{2,1},f_{3,2}\}. \end{array}$$

The candidate 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 candidate repair is $\frac{1}{4}\times\frac{1}{3}=\frac{1}{12}$, and the same holds for any other candidate repair.

Step 2: Polynomial Lower Bound. Now that we have an efficient sampler for the candidate repairs, we proceed to show the following lemma stating that there is a polynomial lower bound on $\mathsf{repfreq}_{\Sigma,Q}(D,\bar{c})$, which will then imply the existence of the desired FPRAS. The general idea underlying the proof of the desired lemma is to focus on a consistent subset H of the database D, which is a homomorphic image of $Q(\bar{c})$ in D, and then show that the number of repairs of $\mathsf{CORep}(D,\Sigma)$ containing H is only polynomially smaller than $|\mathsf{CORep}(D,\Sigma)|$.

Lemma 5.5. *Consider a set Σ of primary keys and a CQ $Q(\bar{x})$. For every database D and tuple $\bar{c}\in\mathsf{dom}(D)^{|\bar{x}|}$ with $\mathsf{repfreq}_{\Sigma,Q}(D,\bar{c})>0$, it holds that $\mathsf{repfreq}_{\Sigma,Q}(D,\bar{c}) \geq \frac{1}{(2\cdot||D||)^{||Q||}}$.*

Proof. By abuse of notation, we treat 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 $\bar{c}\in\mathsf{dom}(D)^{|\bar{x}|}$. 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 $\mathsf{repfreq}_{\Sigma,Q}(D,\bar{c})=0$. Consider now the case that such a homomorphism h exists. Assuming that $Q=\{R_i(\bar{y}_i)\mid i\in[n]\}$, let $h(Q)=\{R_i(h(\bar{y}_i))\mid i\in[n]\}$. Let $m=|h(Q)|$; clearly, $m\leq|Q|$. For every relation name R of the schema with a key $R:X\to Y$ 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 $B_1,\ldots,B_n$ 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 $B_1,\ldots,B_m$. Clearly, no two facts of $h(Q)$ belong to the same block; otherwise, $h(Q)\not\models\Sigma$, which is a contradiction.

Let $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ be the set of candidate repairs $E\in\mathsf{CORep}(D,\Sigma)$ such that $E\cap B_j\neq\emptyset$ for every $j\in[m]$. Let $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ be the set of candidate repairs $E\in\mathsf{CORep}(D,\Sigma)$ such that $E\cap B_j=\emptyset$ for some $j\in[m]$. It is straightforward to see that

$$|\mathsf{CORep}(D,\Sigma)| \;=\; \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| + \left|R^{\mathsf{e}}_{D,\Sigma,h(Q)}\right|.$$

Now, consider a candidate repair E of $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ and assume that E is disjoint with precisely ℓ blocks of $\{B_1,\ldots,B_m\}$. Assume, without loss of generality, that these are the blocks $B_1,\ldots,B_\ell$. We can transform the candidate repair E into a candidate repair $E'\in R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ by bringing back an arbitrary fact of each block B_j for $j\in[\ell]$. Therefore, the candidate repair E is mapped to $|B_1|\times\cdots\times|B_\ell|$ distinct candidate repairs of $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$. Note that this mapping essentially defines a many-to-many relation from $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ to $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ that collects all the pairs of candidate repairs $(E,E')\in R^{\mathsf{e}}_{D,\Sigma,h(Q)}\times R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ such that E is transformed into E' according to the above transformation.

Observe that at most 2^m-1 candidate repairs $E\in R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ are mapped to the same candidate repair $E'\in R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$. This holds since the candidate 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 candidate repair and which fact of B_j we keep. For the blocks $B_1, \dots, B_m$, a candidate 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

$$\left|R^{\mathsf{e}}_{D,\Sigma,h(Q)}\right| \;\leq\; (2^m - 1) \times \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|$$

and

$$|\mathsf{CORep}(D,\Sigma)| \;=\; \left|R^{\mathsf{e}}_{D,\Sigma,h(Q)}\right| + \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| \;\leq\; (2^m-1)\times\left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| + \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| \;=\; 2^m \times \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|.$$

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

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

$$|\{E \in \mathsf{CORep}(D,\Sigma) \mid h(Q) \subseteq E\}| \;=\; \frac{1}{|B_1| \times \cdots \times |B_m|} \times \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|$$

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

$$\begin{aligned}
\frac{|\{E \in \mathsf{CORep}(D,\Sigma) \mid h(Q) \subseteq E\}|}{|\mathsf{CORep}(D,\Sigma)|} \;\geq\; \frac{\frac{1}{|B_1|\times\cdots\times|B_m|} \times \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|}{2^m \times \left|R^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|} \;&=\; \frac{1}{|B_1| \times \cdots \times |B_m| \times 2^m} \\
&\geq\; \frac{1}{|D|^m \times 2^m} \\
&\geq\; \frac{1}{|D|^{|Q|} \times 2^{|Q|}} \\
&=\; \frac{1}{(2 \cdot |D|)^{|Q|}} \\
&\geq\; \frac{1}{(2 \cdot ||D||)^{||Q||}}
\end{aligned}$$

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

Here is a simple example that illustrates the argument given in the proof of Lemma 5.5.

Example 5.6. Consider again the database D depicted in Figure 2 and the set $\Sigma = \{R : A_1 \to A_2\}$ consisting of a single key. Let Q be the CQ $\mathsf{Ans}(x) \text{ :- } R(a_1, x)$. 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 $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ consists of

$$\begin{array}{ccc}
\{f_{1,1}, f_{2,1}\} & \{f_{1,2}, f_{2,1}\} & \{f_{1,3}, f_{2,1}\} \\
\{f_{1,1}, f_{2,1}, f_{3,1}\} & \{f_{1,2}, f_{2,1}, f_{3,1}\} & \{f_{1,3}, f_{2,1}, f_{3,1}\} \\
\{f_{1,1}, f_{2,1}, f_{3,2}\} & \{f_{1,2}, f_{2,1}, f_{3,2}\} & \{f_{1,3}, f_{2,1}, f_{3,2}\}
\end{array}$$

and the set $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ consists of

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

According to the mapping in the proof of Lemma 5.5, the candidate repair $\{f_{2,1}\}$ is mapped to the candidate 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, $\{f_{2,1}, f_{3,1}\}$ is mapped to the candidate 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 $\{f_{2,1}, f_{3,2}\}$ is mapped to the candidate 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 candidate repair of $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ is mapped to precisely three candidate repairs of $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ 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 candidate repair of $R^{\mathsf{e}}_{D,\Sigma,h(Q)}$ is mapped to every candidate repair of $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$. Since all the facts of a block are symmetric with each other, precisely $\frac{1}{3}$ of the candidate repairs in $R^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ contain the fact $R(a_1, b_1)$, i.e., three repairs. Thus, it holds that

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

and

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

We conclude that

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

Note that

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

is indeed a lower bound for that value, and it is also a lower bound for $\mathsf{repfreq}_{\Sigma,Q}(D, (b_1)) = \frac{1}{4}$.

5.3 Proof of Item (3) of Theorem 5.1

For showing that there exist a set Σ of FDs and a CQ Q such that, unless RP = NP, there is no FPRAS for $\mathsf{RepFreq}(\Sigma, Q)$, we provide a rather involved proof that proceeds in two main steps. We give the high level ideas of the proof; the full proof can be found in Appendix B.

We start by giving an auxiliary lemma that is needed by both steps of the proof; its proof can be found in Appendix B. An undirected graph G is called *non-trivially connected* if it contains at least two nodes and is connected. We write $\mathsf{IS}(G)$ 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 $\mathsf{CG}(D, \Sigma)$, 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 $\mathsf{CG}(D, \Sigma)$ is non-trivially connected. We then show the following:

Lemma 5.7. *Consider a non-trivially Σ-connected database D, where Σ is a set of FDs. It holds that*

$$|\mathsf{CORep}(D, \Sigma)| \;=\; |\mathsf{IS}(\mathsf{CG}(D, \Sigma))|.$$

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

Proposition 5.8. *Unless* RP = NP*, there exists a set Σ of keys over a singleton schema such that, given a non-trivially Σ-connected database D, the problem of computing $|\mathsf{CORep}(D, \Sigma)|$ 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. [4] In particular, we show that there exists a set Σ_K of keys over the schema $\mathbf{S} = \{R/\Delta + 1\}$ 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 $\mathbf{S}$ such that $\mathsf{CG}(D_G, \Sigma_K)$ is isomorphic to G. Thus, by Lemma 5.7, $|\mathsf{CORep}(D_G, \Sigma_K)| = |\mathsf{IS}(G)|$. 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 [Misra and Gries 1992]. Hence, given a database D, assuming that the problem of computing the number $|\mathsf{CORep}(D, \Sigma_K)|$ 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 to a contradiction. Therefore, Proposition 5.8 follows with $\Sigma = \Sigma_K$. Note that Proposition 5.8 establishes that for keys, unless RP = NP, the problem of counting the number of operational repairs does not admit an FPRAS. We see this as a strong indication that item (3) of Theorem 5.1 holds even for keys.

The second step of the proof establishes that, unless RP = NP, the existence of an FPRAS for $\mathsf{RepFreq}(\Sigma, Q)$, where Σ is a set of FDs and Q a CQ, would contradict Proposition 5.8 . Let Σ_K be the set of keys provided by Proposition 5.8. We show the following auxiliary result:

Lemma 5.9. *Assume that* $\mathsf{RepFreq}(\Sigma, Q)$ *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* $|\mathsf{CORep}(D, \Sigma_K)|$ *admits an FPRAS.*

To establish the above result, we show that there is 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 $\mathsf{CG}(D_F, \Sigma_F)$ consists of a graph G that is isomorphic to $\mathsf{CG}(D, \Sigma_K)$, and an additional node that is connected via an edge with every node of G. Therefore, by Lemma 5.7, we conclude that

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

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

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

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

It is now straightforward to see that from Proposition 5.8 and Lemma 5.9, we get that, unless RP = NP, there exist a set Σ of FDs and a CQ Q such that there is no FPRAS for $\mathsf{RepFreq}(\Sigma, Q)$.

6 Uniform Sequences

We now concentrate on the Markov chain generator based on uniform sequences.

Theorem 6.1. *The following hold:*

(1) There exist a set Σ *of primary keys and a CQ* Q *such that* $\mathsf{OCQA}(\Sigma, M^{\mathsf{us}}_{\Sigma}, Q)$ *is* $\sharp\mathrm{P}$*-hard.*
(2) For every set Σ *of primary keys and CQ* Q*,* $\mathsf{OCQA}(\Sigma, M^{\mathsf{us}}_{\Sigma}, Q)$ *admits an FPRAS.*

[4] Let us stress that this result is known for arbitrary, not necessarily non-trivially connected graphs [Sly 2010]. Therefore, for our purposes, we had to strengthen it to non-trivially connected graphs.

Notice that the above result does not cover the cases of arbitrary keys and FDs. 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 $\mathrm{OCQA}(\Sigma, M^{\mathsf{us}}_{\Sigma}, Q)$. We now 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$,

$$\mathsf{P}_{M^{\mathsf{us}}_{\Sigma},Q}(D,\bar{c}) \quad = \quad \frac{|\{s \in \mathsf{CRS}(D,\Sigma) \mid \bar{c} \in Q(s(D))\}|}{|\mathsf{CRS}(D,\Sigma)|}.$$

This ratio is the percentage of complete (D,Σ)-repairing sequences that lead to an operational repair that entails $Q(\bar{c})$, called *sequence relative frequency* of $Q(\bar{c})$ w.r.t. D and Σ, denoted $\mathsf{seqfreq}_{\Sigma,Q}(D,\bar{c})$. Thus, we can restate $\mathrm{OCQA}(\Sigma, M^{\mathsf{us}}_{\Sigma}, Q)$ 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^{us}_{Σ}:

PROBLEM :	$\mathrm{SeqFreq}(\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$.
OUTPUT :	$\mathsf{seqfreq}_{\Sigma,Q}(D,\bar{c})$.

We discuss the proof of Theorem 6.1 by considering $\mathrm{SeqFreq}(\Sigma, Q)$ instead of $\mathrm{OCQA}(\Sigma, M^{\mathsf{us}}_{\Sigma}, Q)$.

6.1 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 $\mathrm{RepFreq}(\Sigma, Q)$ is $\sharp$P-hard; Σ and Q are obtained from the proof of item (1) of Theorem 5.1. We show that also $\mathrm{SeqFreq}(\Sigma, Q)$ is $\sharp$P-hard via a polynomial-time Turing reduction from $\sharp H$-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. For an undirected graph G, let D_G be the database that the construction in the proof of item (1) of Theorem 5.1 builds. We show that

$$\mathsf{repfreq}_{\Sigma,Q}(D_G,()) \;=\; \mathsf{seqfreq}_{\Sigma,Q}(D_G,()),$$

which in turn implies that the polynomial-time Turing reduction from $\sharp H$-Coloring to $\mathrm{RepFreq}(\Sigma, Q)$ is also a polynomial-time Turing reduction from $\sharp H$-Coloring to $\mathrm{SeqFreq}(\Sigma, Q)$. Recall that

$$\mathsf{seqfreq}_{\Sigma,Q}(D_G,()) \;=\; \frac{|\{s \in \mathsf{CRS}(D_G,\Sigma) \mid s(D_G) \models Q\}|}{|\mathsf{CRS}(D_G,\Sigma)|}.$$

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

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

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

6.2 Proof of Item (2) of Theorem 6.1

We prove that, for a set Σ of primary keys and a CQ Q, the problem $\mathrm{SeqFreq}(\Sigma, Q)$ admits an FPRAS. As for item (2) of Theorem 5.1, the proof consists of two steps: (1) establish the existence of an efficient sampler, and (2) provide a polynomial lower bound for $\mathsf{seqfreq}_{\Sigma,Q}(D,\bar{c})$. Further details on why those two steps imply the existence of the desired FPRAS are given in Appendix A.

Step 1: Efficient Sampler. To establish that we can efficiently sample elements of $\mathrm{CRS}(D, \Sigma)$ uniformly at random, we first need a non-trivial technical lemma, which shows that, in the case of primary keys, the number of complete repairing sequences can be computed in polynomial time in the size of the database.

Lemma 6.2. *For a database D and a set Σ of primary keys, $|\mathrm{CRS}(D, \Sigma)|$ can be computed in polynomial time in $\|D\|$.*

Proof. Consider a database D. As in the proof of Lemma 5.3, let $B_1, \ldots, B_n$ 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 $0 \leq i \leq \left\lfloor \frac{m}{2} \right\rfloor$, we denote by $S_m^{\mathsf{ne},i}$ the number of sequences $s \in \mathrm{CRS}(B, \Sigma)$ such that $s(B) \neq \emptyset$, which means that $s(B)$ 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 $S_m^{\mathsf{ne},\frac{m}{2}} = 0$. In any other case, we have that

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

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 (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, $S_m^{\mathsf{e},i}$ is the number of sequences $s \in \mathrm{CRS}(B, \Sigma)$ such that $s(B) = \emptyset$ and s has precisely i pair removals. As we cannot obtain an empty repair without pair removals, $S_m^{\mathsf{e},0} = 0$. For $i > 0$,

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

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 (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, i.e., 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 $s \in \mathrm{CRS}(D, \Sigma)$ 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 $s \in \mathrm{CRS}(B_1 \cup \cdots \cup B_j, \Sigma)$ with precisely i pair removals such that $s(D) \cap B_\ell \neq \emptyset$ for k of the blocks of $B_1, \ldots, B_j$; hence, $0 \leq k \leq j$. For $k < 0$ or $k > j$, we have $P_j^{k,i} = 0$. Then,

$$\mathrm{CRS}(D, \Sigma) \;=\; \sum_{k=0}^{n} \sum_{i=0}^{\left\lfloor \frac{|B_1|}{2} \right\rfloor + \cdots + \left\lfloor \frac{|B_n|}{2} \right\rfloor} P_n^{k,i}.$$

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

$$P_1^{0,i} \;=\; S^{\mathrm{e},i}_{|B_1|} \qquad \text{and} \qquad P_1^{1,i} \;=\; S^{\mathrm{ne},i}_{|B_1|}.$$

For $j > 1$, it holds that

$$P_j^{k,i} = \sum_{\substack{0 \leq i_1 \leq \left\lfloor \frac{|B_1|}{2} \right\rfloor + \cdots + \left\lfloor \frac{|B_{j-1}|}{2} \right\rfloor, \\ 0 \leq i_2 \leq \left\lfloor \frac{|B_j|}{2} \right\rfloor, \\ i_1 + i_2 = i}} \left(P_{j-1}^{k,i_1} \times S^{\mathrm{e},i_2}_{|B_j|} \times \binom{|B_1 \cup \cdots \cup B_j| - i_1 - i_2 - k)}{|B_1 \cup \cdots \cup B_{j-1}| - i_1 - k} \right.$$
$$\left. + P_{j-1}^{k-1,i_1} \times S^{\mathrm{ne},i_2}_{|B_j|} \times \binom{|B_1 \cup \cdots \cup B_j| - i_1 - i_2 - k}{|B_1 \cup \cdots \cup B_{j-1}| - i_1 - k + 1} \right),$$

where the last expression is the number of ways to interleave a sequence of $\mathrm{CRS}(B_1 \cup \cdots \cup B_{j-1}, \Sigma)$ that has i_1 pair removals with a sequence of $\mathrm{CRS}(B_j, \Sigma)$ 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, $|B_1 \cup \cdots \cup B_j| - i_1 - i_2 - k$ is the total number of operations in the combined sequence, $|B_1 \cup \cdots \cup B_{j-1}| - i_1 - k$ (or $|B_1 \cup \cdots \cup B_{j-1}| - i_1 - k + 1$) 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 jth block. □

An example that illustrates the algorithm described in the proof of Lemma 6.2 is given in Appendix C. Having Lemma 6.2 in place, we can establish the existence of an efficient sampler.

Lemma 6.3. *Given a database D and a set Σ of primary keys, we can sample elements of $\mathrm{CRS}(D, \Sigma)$ uniformly at random in polynomial time in $\|D\|$.*

Proof. The algorithm SampleSeq, depicted in Algorithm 1, is a recursive algorithm that returns a sequence $s \in \mathrm{CRS}(D, \Sigma)$ with probability $\frac{1}{|\mathrm{CRS}(D,\Sigma)|}$. 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

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

ALGORITHM 1: The Algorithm for Sampling Elements of $\mathsf{CRS}(D, \Sigma)$ Uniformly at Random

Input: A database D and a set Σ of primary keys over a schema $\mathbf{S}$.
Output: $s \in \mathsf{CRS}(D, \Sigma)$ with probability $\frac{1}{|\mathsf{CRS}(D,\Sigma)|}$.

```
return Sample(D, Σ, ε)
Function Sample(D, Σ, s):
    if s(D) ⊨ Σ then
        return s;
    else
        Select a (s(D), Σ)-justified operation op with probability |CRS(op(s(D)),Σ)| / |CRS(s(D),Σ)|
        return Sample(D, Σ, s · op)
    end
```

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

$$\begin{aligned} & \frac{|\mathsf{CRS}(op_1(D), \Sigma)|}{|\mathsf{CRS}(\varepsilon(D), \Sigma)|} \times \frac{|\mathsf{CRS}(op_2(op_1(D)), \Sigma)|}{|\mathsf{CRS}(op_1(D), \Sigma)|} \times \cdots \times \frac{|\mathsf{CRS}(op_n(\ldots D \ldots), \Sigma)|}{|\mathsf{CRS}(op_{n-1}(\ldots D \ldots), \Sigma)|} \\ = \; & \frac{|\mathsf{CRS}(op_n(\ldots D \ldots), \Sigma)|}{|\mathsf{CRS}(\varepsilon(D), \Sigma)|} \\ = \; & \frac{1}{|\mathsf{CRS}(D, \Sigma)|}. \end{aligned}$$

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

$$|\mathsf{CRS}(op_n(\ldots D \ldots), \Sigma)| \;=\; |\mathsf{CRS}(s(D), \Sigma)| \;=\; 1$$

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. Moreover, by Lemma 6.2, for a set Σ of primary keys, we can compute $|\mathsf{CRS}(D, \Sigma)|$ in polynomial time in $||D||$ for any database D, Therefore, we get that the total running time of the algorithm is 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 $\mathsf{seqfreq}_{\Sigma,Q}(D, \bar{c})$, which will then imply the existence of the desired FPRAS. Similarly to Lemma 5.5, we prove the desired lower bound by considering a consistent subset H of the database D, which is a homomorphic image of $Q(\bar{c})$ in D, and then show that the number of repairing sequences s of $\mathsf{CRS}(D, \Sigma)$ such that $s(D)$ contains H is only polynomially smaller than $|\mathsf{CRS}(D, \Sigma)|$.

Lemma 6.4. *Consider a set Σ of primary keys and a CQ $Q(\bar{x})$. For every database D and tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$ with $\mathsf{seqfreq}_{\Sigma,Q}(D, \bar{c}) > 0$, it holds that $\mathsf{seqfreq}_{\Sigma,Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot ||D||)^{||Q||}}$.*

7 Uniform Operations

We finally consider the Markov chain generator based on uniform operations.

Theorem 7.1. *The following hold:*

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

Notice that the above result does not cover the case of FDs, which remains an open problem. However, as we show in the next section, 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. Unlike for 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 has to directly deal with the problem $\mathsf{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$ for a set Σ of FDs and a CQ Q.

7.1 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 $\mathsf{RepFreq}(\Sigma, Q)$ is $\sharp\mathsf{P}$-hard (Σ and Q are extracted from the proof of item (1) of Theorem 5.1), we show that $\mathsf{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$ is $\sharp\mathsf{P}$-hard via a polynomial-time Turing reduction from $\sharp H$-Coloring by reusing the construction in the proof of item (1) of Theorem 5.1; H is the 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

$$\mathsf{repfreq}_{\Sigma,Q}(D_G, ()) \;=\; \mathsf{P}_{M^{\mathsf{uo}}_{\Sigma},Q}(D_G, ()),$$

which implies that the polynomial-time Turing reduction from $\sharp H$-Coloring to $\mathsf{RepFreq}(\Sigma, Q)$ is also a reduction from $\sharp H$-Coloring to $\mathsf{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$. In the proof of item (1) of Theorem 6.1, we have shown that $\mathsf{repfreq}_{\Sigma,Q}(D_G, ()) = \mathsf{seqfreq}_{\Sigma,Q}(D_G, ())$. Thus, it suffices to show that

$$\mathsf{seqfreq}_{\Sigma,Q}(D_G, ()) \;=\; \mathsf{P}_{M^{\mathsf{uo}}_{\Sigma},Q}(D_G, ()).$$

Let $M^{\mathsf{uo}}_{\Sigma}(D_G) = (V, E, \mathbf{P})$. 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 $\mathsf{CRS}(D_G, \Sigma)$ is of length precisely $|V_G|$, and for every non-leaf node $s \in V$, $|\mathsf{Ops}_s(D_G, \Sigma)| = 3 \cdot (|V_G| - |s|)$. Hence, by Definition 4.9, for each $(s, s') \in E$,

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

We conclude that, with π being the leaf distribution of M^{uo}_{Σ}, for each $s = op_1, \ldots, op_n \in \mathsf{CRS}(D_G, \Sigma)$,

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

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

7.2 Proof of Item (2) of Theorem 7.1

We prove that, for a set Σ of keys and a CQ Q, $\mathsf{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$ 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 $\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma},Q}(D, \bar{c})$. Further details on why the above two steps together imply the existence of the desired FPRAS are given in Appendix A.

Step 1: Efficient Sampler. Given a database D, the definition of M^{uo}_{Σ} immediately implies the existence of an efficient sampler that returns a sequence $s \in \mathsf{RL}(\Sigma, M^{\mathsf{uo}}_{\Sigma}(D))$ with probability $\pi(s)$, where π is the leaf distribution of $M^{\mathsf{uo}}_{\Sigma}(D)$. 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

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

Hence, we immediately obtain the following result:

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

Step 2: Polynomial Lower Bound. The rest of the section is devoted to showing that there exists a polynomial lower bound on $\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma}, Q}(D, \bar{c})$, which will then imply the existence of the desired FPRAS.

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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$ with $\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma}, Q}(D, \bar{c}) > 0$, it holds that*

$$\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma}, Q}(D, \bar{c}) \;\geq\; \frac{1}{\mathsf{pol}(||D||)}.$$

We proceed to discuss the main ideas underlying the proof of the above result. 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 :-. 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 $\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma}, Q}(D, \bar{c}) = 0$. Assume now that such a homomorphism h exists. We proceed to prove the claim for the case where $|h(Q)| = 1$. The generalization to the case where $|h(Q)| \geq 1$ can be found in Appendix D. Let f be the single fact of $h(Q)$ and let

$$\mathsf{P}_{D, M^{\mathsf{uo}}_{\Sigma}, Q}(h) \;=\; \sum_{D' \in \mathsf{ORep}(D, M^{\mathsf{uo}}_{\Sigma}) \text{ and } h(Q) \subseteq D'} \mathsf{P}_{D, M^{\mathsf{uo}}_{\Sigma}}(D').$$

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

$$\mathsf{P}_{M^{\mathsf{uo}}_{\Sigma}, Q}(D, \bar{c}) \;\geq\; \mathsf{P}_{D, M^{\mathsf{uo}}_{\Sigma}, Q}(h).$$

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

$$S_f \;=\; \{s \in \mathsf{RL}(M^{\mathsf{uo}}_{\Sigma}(D)) \mid f \in s(D)\} \qquad \text{and} \qquad S_{\neg f} \;=\; \{s \in \mathsf{RL}(M^{\mathsf{uo}}_{\Sigma}(D)) \mid f \notin s(D)\}.$$

With π being the leaf distribution of $M^{\mathsf{uo}}_{\Sigma}(D)$,

$$\mathsf{P}_{D, M^{\mathsf{uo}}_{\Sigma}, Q}(h) \;=\; \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}},$$

where

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

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

$$\mathsf{P}_{D, M^{\mathsf{uo}}_{\Sigma}, Q}(h) \;=\; \frac{\Lambda_f}{\Lambda_f + \Lambda_{\neg f}} \;\geq\; \frac{\Lambda_f}{\Lambda_f + \mathsf{pol}'(||D||) \cdot \Lambda_f} \;=\; \frac{1}{1 + \mathsf{pol}'(||D||)} \tag{1}$$

and the claim follows with $\mathrm{pol}(\|D\|) = 1 + \mathrm{pol}'(\|D\|)$. We proceed to show that a polynomial pol' such that $\Lambda_{\neg f} \leq \mathrm{pol}'(\|D\|) \cdot \Lambda_f$ exists. To this end, we establish a 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 $M^{\mathsf{uo}}_{\Sigma}(D)$:

Lemma 7.4. *There exists a function* $\mathsf{F} : S_{\neg f} \to S_f$ *such that:*

(1) There exists a polynomial pol'' *such that, for every* $s \in S_{\neg f}$, $\pi(s) \leq \mathrm{pol}''(\|D\|) \cdot \pi(\mathsf{F}(s))$.
(2) For every $s' \in S_f$, $|\{s \in S_{\neg f} \mid \mathsf{F}(s) = s'\}| \leq 2 \cdot \|D\| - 1$.

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

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

and the claim follows with $\mathrm{pol}'(\|D\|) = \mathrm{pol}''(\|D\|) \cdot (2 \cdot \|D\| - 1)$. We proceed to prove Lemma 7.4.

Proof of Lemma 7.4. The bulk of the proof is devoted to item (1), whereas item (2) relies on a simple combinatorial argument.

Item (1). Let $s \in \mathsf{RL}(M^{\mathsf{uo}}_{\Sigma}(D))$ be a repairing sequence that removes f, i.e., $s \in S_{\neg f}$. We transform s into a repairing sequence $s' \in \mathsf{RL}(M^{\mathsf{uo}}_{\Sigma}(D))$ 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

$$s = op_1, \quad op_2, \quad \ldots \quad , op_{i-1}, \quad op_i, \quad op_{i+1}, \quad \ldots \quad , op_n,$$

where $op_i = -f$. Then, we define the sequence

$$s' = op_1, \quad op_2, \quad \ldots \quad , op_{i-1}, \quad op_{i+1}, \quad \ldots \quad , op_n, \quad op'_1, \quad \ldots \quad , op'_\ell,$$

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

$$s' = op_1, \quad op_2, \quad \ldots \quad , op_{i-1}, \quad op^\star_i, \quad op_{i+1}, \quad \ldots \quad , op_n, \quad op'_1, \quad \ldots \quad , op'_\ell,$$

where $op^\star_i$ 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 can exist, 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 $\ell \in \{0, \ldots, k\}$, 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. A simple example illustrating the construction of s' follows.

Example 7.5. Consider again the database D depicted in Figure 2 and the set $\Sigma = \{R : A_1 \to A_2, R : A_2 \to A_1\}$ of keys. Consider also the query Q and the homomorphism h from Example 5.6. 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

$$s_1' \;=\; -f_{1,2}, -f_{3,1}, -f_{1,3}, -f_{2,1},$$

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

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

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

$$\mathbf{P}((op_1, \ldots, op_{j-1}), (op_1, \ldots, op_j)) \;=\; \frac{1}{N_j}.$$

Then,

$$\pi(s') \;=\; \frac{1}{N_1} \times \frac{1}{N_2} \times \cdots \times \frac{1}{N_{i-1}} \times \left[\frac{1}{N_i}\right] \times \frac{1}{N'_{i+1}} \times \cdots \times \frac{1}{N'_n} \times \frac{1}{2\ell+1} \times \frac{1}{2(\ell-1)+1} \times \frac{1}{3}.$$

The probability $\mathbf{P}((op_1, \ldots, op_{j-1}), (op_1, \ldots, op_j))$, for $2 \le j \le i-1$, is not affected by the decision to remove or keep f at the ith step. The probability $\mathbf{P}((op_1, \ldots, op_{j-1}), (op_1, \ldots, op_j))$ for $i+2 \le j \le n$, on the other hand, might decrease in the sequence s' compared with 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 $N_j \le N'_j$. Similarly, the probability $\mathbf{P}((op_1, \ldots, op_{i-1}), (op_1, \ldots, op_{i-1}, op_{i+1}))$ (in the case where $op_i = -f$) or $\mathbf{P}((op_1, \ldots, op_i^\star), (op_1, \ldots, op_i^\star, op_{i+1}))$ (in the case where $op_i = -\{f, g\}$) can only decrease compared with the probability $\mathbf{P}((op_1, \ldots, op_i), (op_1, \ldots, op_i, op_{i+1}))$ in s; hence, $N_{i+1} \le N'_{i+1}$.

The term $\frac{1}{N_i}$ denotes the probability of $op_i^\star$ and is included in brackets because it only appears in the expression if the sequence s removes the fact f jointly with some other fact g (and the operation $op_i^\star$ removes g by itself). Since all the $(D^{s'}_{i-1}, \Sigma)$-justified operations have the same probability to be selected, we get that $\mathbf{P}((op_1, \ldots, op_{i-1}), (op_1, \ldots, op_i)) = \mathbf{P}((op_1, \ldots, op_{i-1}), (op_1, \ldots, op_i^\star))$. 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 $\ell \in \{0, \ldots, k\}$ 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 7.6. We continue with Example 7.5. For the sequence s_1, we have that

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

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,

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

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 of applying this operation in s_1 (i.e., $(\mathbf{P}((-f_{1,2}, -f_{1,1}), (-f_{1,2}, -f_{1,1}, -f_{3,1})))$); in fact, it is $\frac{1}{10}$ compared with $\frac{1}{5}$. Finally, there are $\ell = 2$ facts in $s(D)$ that conflict with $f_{1,1}$ and we have that

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

and

$$\mathbf{P}((-f_{1,2}, -f_{3,1}, -f_{1,3}), (-f_{1,2}, -f_{3,1}, -f_{1,3}, -f_{2,1})) \;=\; \frac{1}{2 \times (2-1) + 1} \;=\; \frac{1}{3}.$$

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

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

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 $j \in \{i+1, \ldots, n\}$, let r_j be the difference between N_j and N'_j (i.e., $N'_j = N_j + r_j$). Hence,

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

Note that 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

$$\frac{1}{N_{i+1}} \times \cdots \times \frac{1}{N_n} \times (N_{i+1} + r_{i+1}) \times \cdots \times (N_n + r_n) \times (2\ell + 1) \times \cdots \times 3 \;\leq\; \mathrm{pol}''(\|D\|)$$

for some polynomial pol'', or, equivalently,

$$(N_{i+1} + r_{i+1}) \times \cdots \times (N_n + r_n) \times (2\ell + 1) \times \cdots \times 3 \;\leq\; \mathrm{pol}''(\|D\|) \times N_{i+1} \times \cdots \times N_n.$$

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

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

To show the above, we need to reason about the values r_j. For $j \in \{i+1, \ldots, n\}$, 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 $p \in \{1, \ldots, k\}$, let n_j^p be the number of facts in the database that violate the pth key jointly with f at that point. Note that $n_j^1 + \cdots + n_j^p \geq N_j^f$ as the same fact might violate several distinct keys jointly with f. If $n_j^p \geq 2$, then every fact that violates the pth 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 pth 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 with s, that is, 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 pth 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 pth key with f if $n_j^p = 1$, and one operation removes f itself.) Moreover,

$$N_j \;\geq\; n_j^1 + \cdots + n_j^{p_1} + \frac{n_j^1(n_j^1 - 1)}{2} + \cdots + \frac{n_j^{p_1}(n_j^{p_1} - 1)}{2} \;=\; \frac{(n_j^1)^2 + \cdots + (n_j^{p_1})^2 + n_j^1 + \cdots + n_j^{p_1}}{2}.$$

This is 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 pth key with f are also justified operations at the jth 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 since each such pair of facts jointly violates the pth key.

Example 7.7. We continue with Example 7.6. 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, namely

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

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 $R : A_1 \to A_2$, while the fact $f_{2,1}$ jointly violates with it the key $R : A_2 \to A_1$.

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 $f_{1,2}$ conflicts 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, $f_{1,1}$ introduces another justified operation, that is, the removal of this fact (i.e., $-f_{1,1}$). Hence, in the sequence

$$s_3' \;=\; -f_{3,1}, -f_{1,2}, -f_{2,1}, -f_{1,3}$$

to which s_3 is mapped to, 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. Consequently,

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

and

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

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 obtain that

$$(x_1 + \cdots + x_v)^2 \;\leq\; v \times \left(x_1^2 + \cdots + x_v^2\right).$$

Hence, we have that

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

Note that $N_j^f - (p_2 - p_1)$ is a lower bound of $n_j^1 + \cdots + n_j^{p_1}$ because, for every $p_2 \leq p$, there are no facts that violate the pth key with f, and for $p_1 < p \leq p_2$, there is a single fact that violates

the pth key with f; hence, $n_j^{p_1+1} + \cdots + n_j^{p_2} \leq p_2 - p_1$ and $n_j^{p_2+1} + \cdots + n_j^k = 0$. As said above, $n_j^1 + \cdots + n_j^k \geq N_j^f$. Therefore, we get that

$$n_j^1 + \cdots + n_j^{p_1} \;\geq\; N_j^f - (n_j^{p_1+1} + \cdots + n_j^{p_2}) - (n_j^{p_2+1} + \cdots + n_j^k) \;\geq\; N_j^f - (p_2 - p_1).$$

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

$$r_j \;\leq\; \sqrt{2p_1 N_j} + 2(p_2 - p_1) + 1 \leq \sqrt{2kN_j} + 2k + k \;\leq\; \sqrt{4k^2 N_j} + 3k\sqrt{N_j} = 5k\sqrt{N_j}.$$

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

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

We have that

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

Thus, it suffices to show that

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

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

$$(x_{i+1} + 5k) \times \cdots \times (x_n + 5k) \;\leq\; \mathrm{pol}'''(\|D\|) \times x_{i+1} \times \cdots \times x_n$$

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

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

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,

$$\frac{x_{i+1}+5k}{x_{i+1}} \times \cdots \times \frac{x_n+5k}{x_n} \leq \frac{\sqrt{n-i}+5k}{\sqrt{n-i}} \times \frac{\sqrt{n-i-1}+5k}{\sqrt{n-i-1}} \times \cdots \times \frac{1+5k}{1}$$
$$\leq \frac{\lfloor\sqrt{n-i}\rfloor+5k}{\lfloor\sqrt{n-i}\rfloor} \times \frac{\lfloor\sqrt{n-i-1}\rfloor+5k}{\lfloor\sqrt{n-i-1}\rfloor} \times \cdots \times \frac{1+5k}{1}.$$

Next, for every $m \geq 1$, it holds that $\sqrt{m-1} \geq \sqrt{m}-1$ and thus, $\lfloor\sqrt{m-1}\rfloor \geq \lfloor\sqrt{m}\rfloor - 1$. Hence,

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

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

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

Recall that $c = (2\ell+1) \times \cdots \times 3$, 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

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

and item (1) of Lemma 7.4 follows with

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

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. A sequence $s' \in S_f$ is obtained either from a sequence $s \in \mathsf{RL}(M^{\mathsf{uo}}_{\Sigma}(D))$ 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 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 7.8. We continue with Example 7.7. 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 $f_{1,1}$:

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

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:

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

This completes the proof of Lemma 7.4. □

7.3 The Case of Functional Dependencies

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 not "too small". In fact, we can show the following; the proof is in Appendix D.

Proposition 7.9. *Consider the FD set* $\{R : A_1 \to A_2\}$ *over the schema* $\{R/3\}$ *and the Boolean CQ* $Ans() :\text{-} R(0,0,0)$. *There exists a family* $\{D_n\}_{n\geq 1}$ *of databases such that*

$$0 < \mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D_n, ()) \leq \frac{1}{2^{|D_n|-1}}.$$

Proposition 7.9 tells us that for FDs there are settings where the target probability is unavoidably "too small". Note, however, that this does not preclude the existence of an FPRAS in the case of FDs, which remains an open problem. It only tells us that 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.

8 Singleton Operations

So far we considered the operational CQA framework for FDs in its full generality, where an operation can remove a singe fact or a pair of facts. An interesting question that comes up is whether we can gain something in terms of complexity or approximability when we consider the simpler version of the framework where only operations that remove a single fact are considered; this is precisely what we call here *singleton operations*. Formally, given a database D and a set Σ of FDs, we consider only the set of sequences of $\mathsf{CRS}(D,\Sigma)$, denoted $\mathsf{CRS}^1(D,\Sigma)$, mentioning only operations of the form $-f$, i.e., removing a single fact. Analogously, we consider the set of candidate operational repairs $\mathsf{CORep}^1(D,\Sigma) = \{D' \in \mathsf{CORep}(D,\Sigma) \mid s(D) = D' \text{ for some } s \in \mathsf{CRS}^1(D,\Sigma)\}$. We would like to understand whether the main results established so far, i.e., Theorems 5.1, 6.1, and 7.1, are affected when we focus on singleton operations. The goal of this final section is to provide an answer to this question. As we shall see, adopting singleton operations does not affect Theorem 5.1, Theorem 6.1, and item (1) of Theorem 7.1 that deals with exact query answering. On other hand, item (2) of Theorem 7.1 can be stated for arbitrary FDs (not only for keys). This is a significant finding as it allows us for the first time to push the limits of approximability beyond keys. The main results of this section are summarized in Table 3.

Classical CQA Under Primary Keys. As we shall see, in the case of primary keys, we will be able to inherit results from the classical CQA framework. In particular, we can show that in the case of primary keys and singleton deletions, uniform OCQA coincides with the classical CQA framework, which is an interesting fact in its own right that allows us to inherit known complexity and approximability results. The classical CQA framework is based on the notion of repair, which is a maximal consistent subset of the database. Formally, given a database D and a set Σ of primary

Table 3. The Complexity of (Exact ; Approximate) OCQA for the Markov Chain Generators $M_\Sigma^{\mathsf{ur},1}$, $M_\Sigma^{\mathsf{us},1}$ and $M_\Sigma^{\mathsf{uo},1}$

	Primary Keys	**Keys**	**Functional Dependencies**
$\mathrm{OCQA}(\Sigma, M_\Sigma^{\mathsf{ur},1}, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; ?	♯P-hard ; no FPRAS
$\mathrm{OCQA}(\Sigma, M_\Sigma^{\mathsf{us},1}, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; ?	♯P-hard ; ?
$\mathrm{OCQA}(\Sigma, M_\Sigma^{\mathsf{uo},1}, Q(\bar{x}))$	♯P-hard ; FPRAS	♯P-hard ; FPRAS	♯P-hard ; FPRAS

By FPRAS we mean that the problem in question admits and FPRAS, whereas no FPRAS means that it does not admit an FPRAS, unless RP = NP. The symbol ? refers to the fact that it remains open whether the problem in question admits an FPRAS.

keys, a repair D' of D w.r.t. Σ is a database $D' \subseteq D$ such that $D' \models \Sigma$ but, for every $D' \subsetneq D''$, $D'' \not\models \Sigma$. Let $\mathsf{rep}_\Sigma(D)$ be the set of all repairs of D w.r.t. Σ. It is known that there is a simple way to construct the set $\mathsf{rep}_\Sigma(D)$ via the notion of block. For every relation name R of the underlying schema with a primary key $R : X \to Y$ 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. We write $\mathsf{blocks}_\Sigma(D)$ for the set of all blocks of D (over all the relations of the schema) w.r.t. Σ. It is then easy to verify that the following holds:

Proposition 8.1. *Consider a database D and a set Σ of primary keys. Assuming that* $\mathsf{blocks}_\Sigma(D) = \{B_1, \ldots, B_n\}$, $\mathsf{rep}_\Sigma(D) = \{\{f_1, \ldots, f_n\} \mid (f_1, \ldots, f_n) \in B_1 \times \cdots \times B_n\}$.

The above proposition essentially states that a repair of D w.r.t. Σ can be formed by simply keeping exactly one fact from each block of $\mathsf{blocks}_\Sigma(D)$. Now, the relative frequency of an answer is the percentage of repairs entailing that answer [Maslowski and Wijsen 2013]. To formally define this notion, for a CQ $Q(\bar{x})$ and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, let $\mathsf{rep}_{\Sigma,Q,\bar{c}}(D)$ be the set of repairs D' of D w.r.t. Σ such that $\bar{c} \in Q(D')$. The relative frequency of $\bar{c}$ w.r.t. Q, D and Σ is defined as the ratio

$$\mathsf{rfreq}_{\Sigma,Q}(D,\bar{c}) \;=\; \frac{|\mathsf{rep}_{\Sigma,Q,\bar{c}}(D)|}{|\mathsf{rep}_\Sigma(D)|}.$$

A key problem in the context of CQA is $\mathsf{RelFreq}(\Sigma, Q(\bar{x}))$, where Σ is a set of primary keys and $Q(\bar{x})$ a CQ, which accepts as input a database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, and asks for $\mathsf{rfreq}_{D,\Sigma}(Q,\bar{t})$:

PROBLEM :	$\mathsf{RelFreq}(\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$
OUTPUT :	$\mathsf{rfreq}_{\Sigma,Q}(D,\bar{c})$

The next result is well-known; item (1) has been established in [Maslowski and Wijsen 2013], whereas item (2) is implicit in [Dalvi and Suciu 2007] and made explicit in [Calautti et al. 2019]:

Theorem 8.2. *The following hold:*

(1) There exist a set Σ of primary keys and a CQ Q such that $\mathsf{RelFreq}(\Sigma, Q(\bar{x}))$ *is* ♯P*-hard.*
(2) For every set Σ of primary keys and CQ Q, $\mathsf{RelFreq}(\Sigma, Q(\bar{x}))$ *admits an FPRAS.*

Classical vs. Operational CQA Under Primary Keys. As said above, in the case of primary keys we will be able to inherit results from the classical CQA framework by showing that uniform OCQA coincides with the classical CQA framework. The latter will rely on the following result that relates candidate operational repairs and classical repairs; the proof is in Appendix E:

Proposition 8.3. *For a database D and a set Σ of primary keys,* $\mathsf{CORep}^1(D,\Sigma) = \mathsf{rep}_\Sigma(D)$.

We now proceed to discuss our main results on singleton operations.

8.1 Uniform Repairs

We focus on the repairing Markov chain generator $M_\Sigma^{\mathsf{ur},1}$ such that:

- $\mathsf{ORep}(D, M_\Sigma^{\mathsf{ur},1}) = \mathsf{CORep}^1(D, \Sigma)$, and
- for every repair $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{ur},1})$, it holds that $\mathsf{P}_{D,M_\Sigma^{\mathsf{ur},1}}(D') = \frac{1}{\left|\mathsf{ORep}(D,M_\Sigma^{\mathsf{ur},1})\right|}$.

It is not difficult to adapt Definition 4.3 in order to obtain the Markov chain generator $M_\Sigma^{\mathsf{ur},1}$. We proceed to prove the version of Theorem 5.1 that considers singleton operations:

Theorem 8.4. *The following hold:*

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

As we did for Theorem 5.1, we can 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, $\mathsf{P}_{M_\Sigma^{\mathsf{ur},1},Q}(D, \bar{c}) = \mathsf{repfreq}^1_{\Sigma,Q}(D, \bar{c})$, where

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

Hence, $\mathsf{OCQA}(\Sigma, M_\Sigma^{\mathsf{ur},1}, Q(\bar{x}))$ coincides with the following problem:

PROBLEM :	$\mathsf{RepFreq}^1(\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$.
OUTPUT :	$\mathsf{repfreq}^1_{\Sigma,Q}(D, \bar{c})$.

We proceed to establish Theorem 8.4 by directly considering problem $\mathsf{RepFreq}^1(\Sigma, Q)$, which is independent from $M_\Sigma^{\mathsf{ur},1}$, instead of $\mathsf{OCQA}(\Sigma, M_\Sigma^{\mathsf{ur},1}, Q)$.

Proof of Items (1) and (2) of Theorem 8.4

Having Theorem 8.2, to establish items (1) and (2) of Theorem 8.4, it suffices to show the following:

Proposition 8.5. *For every database D, set Σ of primary keys, CQ $Q(\bar{x})$, and tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$,*

$$\mathsf{repfreq}^1_{\Sigma,Q}(D, \bar{c}) \;=\; \mathsf{rfreq}_{\Sigma,Q}(D, \bar{c}).$$

Proof. By Proposition 8.3, we get that $\mathsf{CORep}^1(D, \Sigma) = \mathsf{rep}_\Sigma(D)$. Since, by definition, $M_\Sigma^{\mathsf{ur},1}$ induces the uniform distribution over its operational repairs, it is easy to see that

$$\mathsf{repfreq}^1_{\Sigma,Q}(D, \bar{c}) \;=\; \frac{|\{D' \in \mathsf{CORep}^1(D, \Sigma) \mid \bar{c} \in Q(D')\}|}{|\mathsf{CORep}^1(D, \Sigma)|} \;=\; \frac{|\mathsf{rep}_{\Sigma,Q,\bar{c}}(D)|}{|\mathsf{rep}_\Sigma(D)|} \;=\; \mathsf{rfreq}_{\Sigma,Q}(D, \bar{c}),$$

and the claim follows. □

Proposition 8.5 essentially shows that the problems $\mathsf{OCQA}(\Sigma, M_\Sigma^{\mathsf{ur},1}, Q(\bar{x}))$ and $\mathsf{RelFreq}(\Sigma, Q(\bar{x}))$ coincide. Therefore, items (1) and (2) of Theorem 8.4 readily follow from Theorem 8.2.

Proof of Item (3) of Theorem 8.4

The proof of this item is along the lines of the proof of item (3) of Theorem 5.1. Here we highlight the key differences. We first need to prove a result analogous to Lemma 5.7, but for the setting of singleton operations. For an undirected graph G, we denote by $\mathsf{IS}_{\neq\emptyset}(G)$ the set of all *non-empty* independent sets of G.

Lemma 8.6. *Consider a non-trivially Σ-connected database D, where Σ is a set of FDs. It holds that,*

$$|\mathsf{CORep}^1(D,\Sigma)| \;=\; |\mathsf{IS}_{\neq\emptyset}(\mathsf{CG}(D,\Sigma))|.$$

Proof. ($\subseteq$) Consider a candidate repair $D' \in \mathsf{CORep}^1(D,\Sigma)$. Since $D' \models \Sigma$, 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 $\mathsf{CG}(D,\Sigma)$. Hence, D' is an independent set of $\mathsf{CG}(D,\Sigma)$. It remains to show that $D' \neq \emptyset$. Since $D' \in \mathsf{CORep}^1(D,\Sigma)$, there is a sequence $s = op_1, \ldots, op_n \in \mathsf{CRS}^1(D,\Sigma)$ such that $s(D) = D'$. Since op_n must be $(s_{n-1}(D),\Sigma)$-justified, there is a violation $(\phi,\{f,g\}) \in \mathsf{V}(s_{n-1}(D),\Sigma)$, for some FD $\phi \in \Sigma$. 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 $D' \in \mathsf{IS}_{\neq\emptyset}(\mathsf{CG}(D,\Sigma))$. As shown in the proof of Lemma 5.7 in Appendix B, one can construct a sequence $s \in \mathsf{CRS}(D,\Sigma)$ 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, $s \in \mathsf{CRS}^1(D,\Sigma)$. Hence, $D' \in \mathsf{CORep}^1(D,\Sigma)$. □

The rest of the proof proceeds in two steps. We first prove the following result, which is analogous to Proposition 5.8. We write $\sharp\mathsf{CORep}^{\mathsf{con},1}(\Sigma)$ for the problem of computing $|\mathsf{CORep}^1(D,\Sigma)|$, given a non-trivially Σ-connected database D.

Proposition 8.7. *Unless* RP = NP*, there exists a set Σ of keys over a singleton schema such that, given a non-trivially Σ-connected database D, the problem of computing $|\mathsf{CORep}^1(D,\Sigma)|$ 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 $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$, for some integer $\Delta \geq 0$, being the problem of computing $|\mathsf{IS}_{\neq\emptyset}(G)|$, given a non-trivially connected graph G with degree Δ, we can prove the following; the proof can be found in Appendix E:

Lemma 8.8. *Unless* RP = NP*, for each $\Delta \geq 6$, it holds that $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$ has no FPRAS.*

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 $|\mathsf{IS}_{\neq\emptyset}(G)| = |\mathsf{CORep}^1(D_G,\Sigma_K)|$. Hence, unless RP = NP, the existence of an FPRAS for the problem of computing $|\mathsf{CORep}^1(D_G,\Sigma_K)|$ would imply an FPRAS for $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$, contradicting Lemma 8.8. The set Σ_K and the database D_G are defined as in the proof of Proposition 5.8. We recall that D_G and Σ_K are such that $|\mathsf{IS}(G)| = |\mathsf{IS}(\mathsf{CG}(D_G,\Sigma_K))|$. Hence, $|\mathsf{IS}_{\neq\emptyset}(G)| = |\mathsf{IS}_{\neq\emptyset}(\mathsf{CG}(D_G,\Sigma_K))|$. Since D_G is non-trivially Σ_K-connected, by Lemma 8.6, $|\mathsf{IS}_{\neq\emptyset}(\mathsf{CG}(D_G,\Sigma_K))| = |\mathsf{CORep}^1(D_G,\Sigma_K)|$. Hence, $|\mathsf{IS}_{\neq\emptyset}(G)| = |\mathsf{CORep}^1(D_G,\Sigma_K)|$, as needed. □

It remains to prove a result analogous to Lemma 5.9. Let Σ_K be the set of keys provided by Proposition 8.7.

Lemma 8.9. *Assume that $\mathsf{RepFreq}^1(\Sigma,Q)$ admits an FPRAS, for every set Σ of FDs and CQ Q. Given a Σ_K-connected database D, the problem of computing $|\mathsf{CORep}^1(D,\Sigma_K)|$ admits an FPRAS.*

Proof. The proof proceeds in the same way as the one of Lemma 5.9. 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

$$\mathsf{repfreq}^1_{\Sigma_F,Q_F}(D_F,()) \;=\; \frac{1}{\left|\mathsf{CORep}^1(D,\Sigma_K)\right|+1}.$$

This is done using the same argument as in the proof of Lemma 5.9, with the difference that we use Lemma 8.6, instead of Lemma 5.7, to prove that $|\mathsf{CORep}^1(D_F,\Sigma_F)| = |\mathsf{CORep}^1(D,\Sigma_K)|+1$. □

It is now easy to see that Proposition 8.7 and Lemma 8.9 imply item (3) of Theorem 8.4.

8.2 Uniform Sequences

Recall that, for a database D and a set Σ of FDs, $\mathsf{CRS}^1(D,\Sigma)$ is the set of sequences of $\mathsf{CRS}(D,\Sigma)$ using only singleton operations. We focus on the repairing Markov chain generator $M^{\mathsf{us},1}_\Sigma$ such that, for every $s \in \mathsf{CRS}^1(D,\Sigma)$, assuming that π is the leaf distribution of $M^{\mathsf{us},1}_\Sigma(D)$, $\pi(s) = \frac{1}{|\mathsf{CRS}^1(D,\Sigma)|}$. It is not difficult to adapt Definition 4.6 in order to obtain the Markov chain generators $M^{\mathsf{us},1}_\Sigma$. We proceed to establish the version of Theorem 6.1 that considers singleton operations:

Theorem 8.10. *The following hold:*

(1) There exist a set Σ of primary keys and a CQ Q such that $\mathsf{OCQA}(\Sigma, M^{\mathsf{us},1}_\Sigma, Q)$ is $\sharp$P-hard.

(2) For every set Σ of primary keys and CQ Q, $\mathsf{OCQA}(\Sigma, M^{\mathsf{us},1}_\Sigma, Q)$ admits an FPRAS.

As for Theorem 6.1, we can 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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, $\mathsf{P}_{M^{\mathsf{us},1}_\Sigma,Q}(D,\bar{c}) = \mathsf{seqfreq}^1_{\Sigma,Q}(D,\bar{c})$, where

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

Hence, $\mathsf{OCQA}(\Sigma, M^{\mathsf{us},1}_\Sigma, Q(\bar{x}))$ is in fact the following problem, which is independent from $M^{\mathsf{us},1}_\Sigma$:

PROBLEM :	$\mathsf{SeqFreq}^1(\Sigma, Q(\bar{x}))$
INPUT :	A database D and a tuple $\bar{c} \in \mathsf{dom}(D)^{\lvert\bar{x}\rvert}$.
OUTPUT :	$\mathsf{seqfreq}^1_{\Sigma,Q}(D,\bar{c})$.

As for uniform repairs, we establish Theorem 8.10 by showing that for every database D, set Σ of primary keys, CQ $Q(\bar{x})$, and tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, it holds that $\mathsf{seqfreq}^1_{\Sigma,Q}(D,\bar{c}) = \mathsf{rfreq}_{\Sigma,Q}(D,\bar{c})$; the proof can be found in Appendix E.

8.3 Uniform Operations

For a database D and a set Σ of FDs, we write $\mathsf{RS}^1(D,\Sigma)$ for the set of sequences in $\mathsf{RS}(D,\Sigma)$ using only operations of the form $-f$, i.e., removing a single fact. Similarly, we write $\mathsf{Ops}^1_s(D,\Sigma)$ for $\{s' \in \mathsf{RS}^1(D,\Sigma) \mid s' = s \cdot op \text{ for some } D\text{-operation } op\}$. Then, we define the Markov chain generator $M^{\mathsf{uo},1}_\Sigma$ such that for $s, s' \in \mathsf{RS}^1(D,\Sigma)$, assuming that $M^{\mathsf{uo},1}_\Sigma(D) = (V,E,\mathbf{P})$, if $s' \in \mathsf{Ops}^1_s(D,\Sigma)$ then

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

Observe, however, that the Markov chain generator $M_\Sigma^{\mathsf{uo},1}$ is defined over all the sequences of $\mathsf{RS}(D,\Sigma)$. If $s \in \mathsf{RS}^1(D,\Sigma)$ but $s' \in \mathsf{RS}(D,\Sigma)\setminus\mathsf{RS}^1(D,\Sigma)$ (and $s' \in \mathsf{Ops}_s(D,\Sigma)$), then we define $\mathbf{P}(s,s') = 0$. If $s \in \mathsf{RS}(D,\Sigma)\setminus\mathsf{RS}^1(D,\Sigma)$, none of the leaves of the subtree T_s is reachable with non-zero probability, and thus, $\mathbf{P}(s,s')$, for any $s' \in \mathsf{Ops}_s(D,\Sigma)$, can get an arbitrary probability (as long as the sum of probabilities equals one), e.g., $\frac{1}{|\mathsf{Ops}_s(D,\Sigma)|}$. We proceed to establish the version of Theorem 7.1 that considers singleton operations:

Theorem 8.11. *The following hold:*

(1) There exist a set Σ of primary keys and a CQ Q such that $\mathsf{OCQA}(\Sigma, M_\Sigma^{\mathsf{uo},1}, Q)$ is $\sharp\mathsf{P}$-hard.

(2) For every set Σ of FDs and CQ Q, $\mathsf{OCQA}(\Sigma, M_\Sigma^{\mathsf{uo},1}, Q)$ admits an FPRAS.

Proof of Item (1) of Theorem 8.11

As in the case of uniform repairs and uniform sequences, we show that OCQA and classical CQA coincide, and then use item (1) of Theorem 8.2.

Proposition 8.12. *For every database D, set Σ of primary keys, CQ $Q(\bar{x})$, and tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$,*

$$\mathsf{P}_{M_\Sigma^{\mathsf{uo},1},Q}(D,\bar{t}) \;=\; \mathsf{rfreq}_{D,\Sigma}(Q,\bar{c}).$$

Proof. Let D be a database and Σ a set of primary keys. It suffices to show the following:

(1) The set of operational repairs and the set of classical repairs coincide, i.e., $\mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1}) = \mathsf{rep}_\Sigma(D)$.

(2) The Markov chain generator $M_\Sigma^{\mathsf{uo},1}$ induces the uniform distribution over the operational repairs, i.e., for every operational repair $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1})$, it holds that

$$\mathsf{P}_{D,M_\Sigma^{\mathsf{uo},1}}(D') \;=\; \frac{1}{|\mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1})|}.$$

We first show item (1). By Proposition 8.3, $\mathsf{CORep}^1(D,\Sigma) = \mathsf{rep}_\Sigma(D)$. Note that the latter statement is independent from the underlying Markov chain generator. Hence, it remains to show that $\mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1}) = \mathsf{CORep}^1(D,\Sigma)$. It is straightforward, from the definition of $M_\Sigma^{\mathsf{uo},1}$, that $\mathsf{RL}(M_\Sigma^{\mathsf{uo},1}(D)) = \mathsf{CRS}^1(D,\Sigma)$, and since the repairs of $\mathsf{CORep}^1(D,\Sigma)$ are precisely those obtained by sequences from $\mathsf{CRS}^1(D,\Sigma)$, we get that $\mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1}) = \mathsf{CORep}^1(D,\Sigma)$, as needed.

We proceed with item (2). Let $D', D'' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{uo},1})$ be operational repairs of D w.r.t. $M_\Sigma^{\mathsf{uo},1}$. Assume that $\mathsf{blocks}_\Sigma(D) = \{B_1, \ldots, B_n\}$. For some $f \in D$, let B_f be the unique block that contains f. Recall that, for $D^* \in \{D', D''\}$,

$$\mathsf{P}_{D,M_\Sigma^{\mathsf{uo},1}}(D^*) \;=\; \sum_{s \in \mathsf{RL}(M_\Sigma^{\mathsf{uo},1}(D)) \text{ and } D^* = s(D)} \pi(s),$$

where π is the leaf distribution of $M_\Sigma^{\mathsf{uo},1}(D)$. Let $\mathsf{RL}_{D^*}(M_\Sigma^{\mathsf{uo},1}(D))$ be the subset of reachable leaves $s \in \mathsf{RL}(M_\Sigma^{\mathsf{uo},1}(D))$ such that $s(D) = D^*$. We provide a bijection $\mu : \mathsf{RL}_{D'}(M_\Sigma^{\mathsf{uo},1}(D)) \to \mathsf{RL}_{D''}(M_\Sigma^{\mathsf{uo},1}(D))$ such that, for every $s \in \mathsf{RL}_{D'}(M_\Sigma^{\mathsf{uo},1}(D))$, $\pi(s) = \pi(\mu(s))$. Since every repair is of the same size (as every repair contains one fact from each block), we also have that every repairing sequence in $\mathsf{CRS}^1(D,\Sigma)$ is of the same length, i.e., it contains $|D\setminus D'|$ many singleton operations. Let $m = |D\setminus D'|$. Fix some repairing sequence $s' \in \mathsf{RL}_{D'}(M_\Sigma^{\mathsf{uo},1}(D))$ and let $s' = op'_1, \ldots, op'_m$. We are now going to construct a repairing sequence $s'' = op''_1, \ldots, op''_m$ such that $s''(D) = D''$. For each $i \in [m]$, if $op'_i = -f$ and $f \notin D''$, then let $op''_i = op'_i$; otherwise, If $op'_i = -f$ and $f \in D''$, then let $g \in D' \cap B_f$ and $op''_i = -g$. Note that g, and thus the operation op''_i, is uniquely defined as

there is only one fact from every block in D'. Now, let $\mu(s') = s''$, for each $s' \in \mathsf{RL}_{D'}(M_\Sigma^{\mathrm{uo},1}(D))$. Intuitively, we define s'' to be the same as s', except when some operation op'_i removes a fact which appears in D'', then op''_i removes the fact from the same block which appears in D' instead. It is straightforward to see that μ is indeed bijective. In fact, it follows directly from the definition that μ is injective. Furthermore, note that $|\mathsf{RL}_{D'}(M_\Sigma^{\mathrm{uo},1}(D))| = |\mathsf{RL}_{D''}(M_\Sigma^{\mathrm{uo},1}(D))| = m!$ since the singleton operations removing the m facts in $D \setminus D'$ or $D \setminus D''$ can be applied in an arbitrary order (see the discussion in the proof of Proposition 8.5), and thus, μ is also surjective.

It remains to show that $\pi(s') = \pi(\mu(s'))$, for every $s' \in \mathsf{RL}_{D'}(M_\Sigma^{\mathrm{uo},1}(D))$. Recall that the probability of a sequence is the product of the probabilities of every operation in that sequence. In fact, we will show by induction that the ith operation of s' has the same probability as the i-th operation of s'', for each $i \in [m]$

For some $i \in [m]$, let $s'_i = op'_1, \ldots, op'_{i-1}$ ($s''_i = op''_1, \ldots, op''_{i-1}$) and $op'_j = -f'_j$ ($op''_j = -f''_j$), for $j \in [i]$. We need to show that for $M_\Sigma^{\mathrm{uo},1}(D) = (\epsilon, P)$, it holds that $P(s'_i, s'_i \cdot op'_i) = P(s''_i, s''_i \cdot op''_i)$. Note that, for every $op''_j = -f''_j$, for $j \in [i]$, we know that $f''_j \in B_{f'_j}$ by construction of s''. Since for every operation op'_i, for $i \in [m]$, we have that $P(s'_i, s'_i \cdot op'_i) = \frac{1}{|\mathrm{Ops}^1_{s'_i}(D,\Sigma)|}$, and similarly, $P(s''_i, s''_i \cdot op''_i) = \frac{1}{|\mathrm{Ops}^1_{s''_i}(D,\Sigma)|}$, it suffices to show that $|\mathrm{Ops}^1_{s'_i}(D,\Sigma)| = |\mathrm{Ops}^1_{s''_i}(D,\Sigma)|$ for each $i \in [m]$. Indeed, we will show that $|B_j \cap s'_i(D)| = |B_j \cap s''_i(D)|$ for all $j \in [n]$ and $i \in [m]$ by induction on i. Note that this statement is stronger since for some repairing sequence $s \in \mathsf{RS}(D,\Sigma)$ we have that

$$|\mathrm{Ops}^1_s(D,\Sigma)| = \sum_{j \in [n] \text{ with } |B_j \cap s(D)| > 2} |B_j \cap s(D)|.$$

For $i = 1$, since $s'_1 = s''_1 = \varepsilon$, i.e., both s'_1 and s''_1 are the empty sequence, it straightforward to see that indeed $|B_j \cap s'_i(D)| = |B_j \cap s''_i(D)|$ for all $j \in [n]$, and thus, $|\mathrm{Ops}^1_{s'_i}(D,\Sigma)| = |\mathrm{Ops}^1_{s''_i}(D,\Sigma)|$. Assume now that $|B_j \cap s'_k(D)| = |B_j \cap s''_k(D)|$ for all $j \in [n]$ and $k \in [i-1]$. We want to show that $|B_j \cap s'_i(D)| = |B_j \cap s''_i(D)|$, for each $j \in [n]$. Fix some $j \in [n]$. Clearly, if $f'_i \notin B_j$, i.e. op'_i does not remove a fact from block B_j, then also op''_i does not remove a fact from B_j, as f'_i and f''_i are of the same block for all operations in s' and s''; thus, $|B_j \cap s'_i(D)| = |B_j \cap s'_{i-1}(D)| = |B_j \cap s''_{i-1}(D)| = |B_j \cap s''_i(D)|$ by induction hypothesis. Consider now the case where $f'_i \in B_j$, and thus, also $f''_i \in B_j$. Since both operations op'_i and op''_i remove a single fact from block B_j, by induction hypothesis, $|B_j \cap s'_i(D)| = |B_j \cap s'_{i-1}(D)| - 1 = |B_j \cap s''_{i-1}(D)| - 1 = |B_j \cap s''_i(D)|$. This concludes our proof. □

Proof of Item (2) of Theorem 8.11

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. Again, further details on why the above two steps together imply the existence of the desired FPRAS are given in Appendix A. Concerning the efficient sampler, we can sample elements of $\mathsf{RL}(M_\Sigma^{\mathrm{uo},1}(D))$ according to the leaf distribution of $M_\Sigma^{\mathrm{uo},1}(D)$ in polynomial time in $\|D\|$. This is done by employing the same iterative algorithm as the one used to sample elements of $\mathsf{RL}(M_\Sigma^{\mathrm{uo}}(D))$, but with the difference that only justified operations that consist of singleton operations are considered: 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

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

Hence, we immediately obtain the following result:

Lemma 8.13. *Given a database D and a set Σ of FDs, we can sample elements of $\mathsf{RL}(M_\Sigma^{\mathsf{uo},1}(D))$ according to the leaf distribution of $M_\Sigma^{\mathsf{uo},1}(D)$ in polynomial time in $\|D\|$.*

We now show that there exists a polynomial lower bound on the target probability.

Lemma 8.14. *Consider a set Σ of FDs and a CQ $Q(\bar{x})$. For every database D and tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$ with $\mathsf{P}_{M_\Sigma^{\mathsf{uo},1},Q}(D,\bar{c}) > 0$, it holds that*

$$\mathsf{P}_{M_\Sigma^{\mathsf{uo},1},Q}(D,\bar{c}) \;\geq\; \frac{1}{(e \cdot \|D\|)^{\|Q\|}}.$$

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 $\mathsf{P}_{M_\Sigma^{\mathsf{uo},1},Q}(D,\bar{c}) = 0$. We now focus on the case where such a homomorphism h exists. Assume that $|h(Q)| = m$ for some $m \in [|Q|]$. 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:

$$\mathsf{P}_{D,M_\Sigma^{\mathsf{uo},1},Q}(h) \;\geq\; \frac{1}{\binom{n+m}{m}},$$

where

$$\mathsf{P}_{D,M_\Sigma^{\mathsf{uo},1},Q}(h) \;=\; \sum_{D' \in \mathsf{ORep}(D,M_\Sigma^{\mathsf{uo},1}) \text{ and } h(Q) \subseteq D'} \mathsf{P}_{D,M_\Sigma^{\mathsf{uo},1}}(D').$$

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, that is, D itself. In this case, the probability of obtaining an operational repair that contains all the facts of $h(Q)$ is $1 = \frac{1}{\binom{0+m}{m}}$.

Inductive Step. We now assume that the claim holds for databases where $n = 0, \ldots, k-1$, and we prove that it holds for databases D where $n = k$. Every repairing sequence $s \in \mathsf{RL}(M_\Sigma^{\mathsf{uo},1}(D))$ 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 $f_1, \ldots, f_k$ be these facts of $D \setminus h(Q)$, and for each $i \in \{1, \ldots, k\}$, let op_i be the operation that removes the fact f_i. We then have that $\mathbf{P}(\varepsilon, (op_i)) \geq \frac{1}{k+m}$. 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 removed. For every $i \in \{1, \ldots, p\}$, 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

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

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

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

Therefore, we can conclude that

$$\mathrm{P}_{D,M_\Sigma^{\mathrm{uo},1},Q}(h) \;\geq\; \sum_{i=1}^{k}\left(\frac{1}{k+m}\times\frac{1}{\binom{k-1+m}{m}}\right) \;=\; \frac{k}{k+m}\times\frac{1}{\binom{k-1+m}{m}} \;=\; \frac{k}{k+m}\times\frac{1}{\frac{(k-1+m)!}{m!\times(k-1)!}}$$
$$= \frac{m!\times k!}{(k+m)!}$$
$$= \frac{1}{\binom{k+m}{m}}.$$

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 $D\backslash h(Q)$ contains n facts that are involved in violations of the FDs, we have that

$$\mathrm{P}_{D,M_\Sigma^{\mathrm{uo},1},Q}(h) \;\geq\; \frac{1}{\binom{n+m}{n}} \;\geq\; \frac{1}{\left(\frac{e(n+m)}{m}\right)^m} \;=\; \frac{m^m}{e^m}\times\frac{1}{(n+m)^m} \;\geq\; \left(\frac{m}{e}\right)^m\times\frac{1}{||D||^m} \;\geq\; \left(\frac{1}{e}\right)^{|Q|}\times\frac{1}{||D||^{||Q||}}.$$

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

$$\mathrm{P}_{M_\Sigma^{\mathrm{uo},1},Q}(D,\bar{c}) \;\geq\; \mathrm{P}_{D,M_\Sigma^{\mathrm{uo},1},Q}(h) \;\geq\; \frac{1}{(e||D||)^{||Q||}} \;\geq\; \frac{1}{(e||D||)^{||Q||}},$$

which concludes our proof. □

9 Conclusions

The take-home message of our work is that uniform operational CQA is flexible enough to lead to approximability results that go beyond the simple case of primary keys. Although we understand relatively well uniform operational CQA, there are still interesting and highly non-trivial open problems concerning approximability:

(1) the case of keys and uniform repairs (we only have a negative result for the problem of counting repairs),
(2) the case of keys/FDs and uniform sequences, and
(3) the case of FDs and uniform operations (we only have a positive result assuming singleton operations).

Another interesting direction for future research is to consider conceptually relevant distributions that deviate from the uniform ones considered in this work, and analyze the complexity of exact and approximate operational CQA. It will be also interesting to consider application-centric Markov chain generators that are designed having a specific application in mind such as data integration.

References

Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. 1999. Consistent query answers in inconsistent databases. In *PODS*. 68–79.

Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press.

Marco Calautti, Marco Console, and Andreas Pieris. 2019. Counting database repairs under primary keys revisited. In *PODS*. 104–118.

Marco Calautti, Marco Console, and Andreas Pieris. 2021. Benchmarking approximate consistent query answering. In *PODS*. 233–246.

Marco Calautti, Leonid Libkin, and Andreas Pieris. 2018. An operational approach to consistent query answering. In *PODS*. 239–251.

Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. 2022. Counting database repairs entailing a query: The case of functional dependencies. In *PODS*. 403–412.

Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. 2024. Combined approximations for uniform operational consistent query answering. *Proc. ACM Manag. Data* 2, 2 (2024), 99.

Jan Chomicki and Jerzy Marcinkowski. 2005. Minimal-change integrity maintenance using tuple deletions. *Inf. Comput.* 197, 1-2 (2005), 90–121.
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.
Nilesh N. Dalvi and Dan Suciu. 2007. Management of probabilistic data: Foundations and challenges. In *PODS*. 1–12.
Martin E. Dyer and Catherine S. Greenhill. 2000. The complexity of counting graph homomorphisms. *Random Struct. Algorithms* 17, 3-4 (2000), 260–289.
Ariel Fuxman, Elham Fazli, and Renée J. Miller. 2005. ConQuer: Efficient management of inconsistent databases. In *SIGMOD*. 155–166.
Ariel Fuxman and Renée J. Miller. 2007. First-order query rewriting for inconsistent databases. *J. Comput. Syst. Sci.* 73, 4 (2007), 610–635.
Floris Geerts, Fabian Pijcke, and Jef Wijsen. 2015. First-order under-approximations of consistent query answers. In *SUM*. 354–367.
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.
Richard M. Karp and Richard J. Lipton. 1980. Some connections between nonuniform and uniform complexity classes. In *STOC*. 302–309.
Christoph Koch. 2008. On Query Algebras for Probabilistic Databases. *SIGMOD Rec.* 37, 4 (2008), 78–85.
Paraschos Koutris and Dan Suciu. 2014. A dichotomy on the complexity of consistent query answering for atoms with simple keys. In *ICDT*. 165–176.
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.
Paraschos Koutris and Jef Wijsen. 2021. Consistent query answering for primary keys in datalog. *Theory Comput. Syst.* 65, 1 (2021), 122–178.
Dany Maslowski and Jef Wijsen. 2013. A dichotomy in the complexity of counting database repairs. *J. Comput. Syst. Sci.* 79, 6 (2013), 958–983.
Jayadev Misra and David Gries. 1992. A constructive proof of Vizing's theorem. *Inform. Process. Lett.* 41, 3 (1992), 131–133.
Allan Sly. 2010. Computational transition at the uniqueness threshold. In *FOCS*. 287–296.

Appendices

A Efficient Approximations

We discuss the general technique we employ in the main body of the article for obtaining the desired FPRASs: establish the existence of an efficient sampler and provide a polynomial lower bound for the ratio of interest. Consider an alphabet Γ and let $x \in \Gamma^*$ be a string over this alphabet. Moreover, let $\mathcal{S} : \Gamma^* \to 2^{\Gamma^*}$ be a function that assigns a finite set of strings to a string, $\mathcal{P} : \Gamma^* \times \Gamma^* \to \{0, 1\}$, and $\mathcal{D}$ be a randomized procedure such that, given a string $x \in \Gamma^*$, produces an element $y \in \mathcal{S}(x)$ with some probability $p_{x,y}$ such that $\sum_{y \in \mathcal{S}(x)} p_{x,y} = 1$, namely $\mathcal{D}$ samples elements from $\mathcal{S}(x)$ according to some probability distribution. Assuming that $\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)$ denotes the probability of sampling some $y \in \mathcal{S}(x)$ such that $\mathcal{P}(x, y) = 1$, i.e., the number

$$\sum_{y \in \mathcal{S}(x) \text{ and } \mathcal{P}(x,y)=1} p_{x,y},$$

we are interested in the following problem:

PROBLEM :	$\mathsf{ExpVal}(\mathcal{S}, \mathcal{P}, \mathcal{D})$
INPUT :	A string $x \in \Gamma^*$.
OUTPUT :	$\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)$.

Now, an FPRAS for $\mathsf{ExpVal}(\mathcal{S}, \mathcal{P}, \mathcal{D})$ is a randomized algorithm A that takes as input a string $x \in \Gamma^*$, $\epsilon > 0$, and $0 < \delta < 1$, runs in polynomial time in $|x|$, $1/\epsilon$ and $\log(1/\delta)$, and produces a random variable $\mathsf{A}(x, \epsilon, \delta)$ such that

$$\Pr\left(\left|\mathsf{A}(x, \epsilon, \delta) - \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)\right| \leq \epsilon \cdot \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)\right) \geq 1 - \delta.$$

Consider, for example, the problem $\mathsf{RepFreq}(\Sigma, Q(\bar{x}))$ for a set Σ of FDs and a CQ Q. It is easy to verify that $\mathsf{RepFreq}(\Sigma, Q(\bar{x}))$ is essentially the problem $\mathsf{ExpVal}\mathcal{S}, \mathcal{P}, \mathcal{D})$, where a string $x \in \Gamma^*$ encodes a database D together with a tuple $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$, $\mathcal{S}(x)$ is the set of all encodings of repairs in $\mathsf{CORep}(D, \Sigma)$, $\mathcal{P}(x, y) = 1$ iff y encodes a repair $D' \in \mathsf{CORep}(D, \Sigma)$ such that $\bar{c} \in Q(D')$, and $\mathcal{D}$ samples elements from $\mathcal{S}(x)$ uniformly at random, in which case

$$\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) \;=\; \frac{|\{y \in \mathcal{S}(x) \mid \mathcal{P}(x, y) \;=\; 1\}|}{|\mathcal{S}(x)|} \;=\; \mathsf{repfreq}_{\Sigma,Q}(D, \bar{c}).$$

Hence, the existence of an FPRAS for $\mathsf{ExpVal}(\mathcal{S}, \mathcal{P}, \mathcal{D})$, with $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{D}$ defined as above immediately implies the existence of an FPRAS for $\mathsf{RepFreq}(\Sigma, Q(\bar{x}))$; we can have an analogous discussion for the case of uniform sequences and uniform operations. We now recall the following result, which is somehow well-known in probability theory, on which we heavily rely in the main body for providing FPRASs for the problems of interest.

Theorem A.1. *Consider a function $\mathcal{S} : \Gamma^* \to 2^{\Gamma^*}$ that assigns a finite set of strings to a string, a function $\mathcal{P} : \Gamma^* \times \Gamma^* \to \{0, 1\}$, and a randomized procedure $\mathcal{D}$ such that, given a string $x \in \Gamma^*$, samples elements of $\mathcal{S}(x)$ according to some probability distribution. The problem* $\mathsf{ExpVal}(\mathcal{S}, \mathcal{P}, \mathcal{D})$ *admits an FPRAS if the following conditions hold:*

- *$\mathcal{D}$ runs in polynomial time w.r.t. the size of its input.*
- *For all $x \in \Gamma^*$ and $y \in \mathcal{S}(x)$, $\mathcal{P}(x, y)$ can be computed in polynomial time w.r.t. $|x|$.*
- *There exists a polynomial function* $\mathsf{pol} : \mathbb{N} \to \mathbb{N}$ *such that*

$$\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) > 0 \quad \textit{implies} \quad \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) \geq \frac{1}{\mathsf{pol}(|x|)}.$$

Proof. Consider the randomized procedure A that takes as input a string $x \in \Gamma^*$, $\epsilon > 0$, and $0 < \delta < 1$, and performs the following steps:

(1) $N := \dfrac{\log(2/\delta)}{2 \cdot \epsilon^2} \cdot \mathsf{pol}(|x|)^2$.
(2) $ctr := 0$.
(3) For $i = 1$ to N do the following:
   (a) Sample $y \in \mathcal{S}(x)$ by executing $\mathcal{D}$ with input x.
   (b) If $\mathcal{P}(x, y) = 1$, then $ctr := ctr + 1$.
(4) Return $\dfrac{ctr}{N}$.

We proceed to show that A is an FPRAS for $\mathsf{ExpVal}(\mathcal{S}, \mathcal{P}, \mathcal{D})$. The fact that A runs in polynomial time w.r.t. $|x|$, $1/\epsilon$, and $\log(1/\delta)$ follows from the assumption that $\mathcal{D}$ runs in polynomial time, $\mathcal{P}(x, y)$ can be computed in polynomial time w.r.t. $|x|$, and by definition of N. It remains to show that A enjoys the desired probabilistic guarantees. We first observe that, if $\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) = 0$, then the claim follows immediately. Hence, we assume that $\mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) > 0$. In this case, we note that A computes the random average $\overline{X}_N$ of N independent Bernoulli random variables $X_1, \ldots, X_N$, i.e., random variables whose value is either 1 or 0. In other words,

$$\overline{X}_N \;=\; \frac{1}{N} \cdot \sum_{i=1}^{N} X_i.$$

Moreover, since the elements of $\mathcal{S}(x)$ are all sampled according to the same probability distribution given by $\mathcal{D}$, we have that all such variables have the same probability of being equal to 1, namely

$$p \;=\; \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x).$$

By the well-known Hoeffding's inequality from probability theory, for every $N \geq \frac{\log(2/\delta)}{2 \cdot \epsilon^2 \cdot p^2}$,

$$\Pr\left(\left|\overline{X}_N - p\right| \leq \epsilon \cdot p\right) \;\geq\; 1 - \delta.$$

Since $p > 0$, by hypothesis, we have that

$$p \;=\; \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x) \;\geq\; \frac{1}{\mathrm{pol}(|x|)},$$

and since $N = \frac{\log(2/\delta)}{2 \cdot \epsilon^2} \cdot \mathrm{pol}(|x|)^2$, we conclude that

$$N \;=\; \frac{\log(2/\delta)}{2 \cdot \epsilon^2} \cdot \mathrm{pol}(|x|)^2 \;\geq\; \frac{\log(2/\delta)}{2 \cdot \epsilon^2 \cdot p^2},$$

which, together with Hoeffding's inequality, implies that

$$\Pr\left(\left|\mathsf{A}(x,\epsilon,\delta) - \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)\right| \;\leq\; \epsilon \cdot \mathsf{P}_{\mathcal{S},\mathcal{P},\mathcal{D}}(x)\right) \;\geq\; 1 - \delta,$$

and the claim follows. □

B Proofs of Section 5

Recall that the main result of Section 5 is the following:

Theorem 5.1. *The following hold:*

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

As discussed in Section 5, we actually need to prove the above result for $\mathrm{RepFreq}(\Sigma, Q)$. Items (1) and (2) have been already proved in the main body of the article. On the other hand, for item (3) we only discussed the high level ideas underlying its proof. We proceed to give the full proof.

B.1 Proof of Item (3) of Theorem 5.1

As discussed in the main body of the article, the proof of item (3) of Theorem 5.1 proceeds in two main steps, which correspond to Proposition 5.8 and Lemma 5.9. 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, $\mathsf{IS}(G)$ 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 $\mathrm{CG}(D, \Sigma) = (V, E)$, where $V = D$, and $\{f, g\} \in E$ if $\{f, g\} \not\models \Sigma$. We call D non-trivially (resp., trivially) Σ-connected if $\mathrm{CG}(D, \Sigma)$ is non-trivially (resp., trivially) connected.

To prove the desired claims, we need 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 5.7, which we recall and prove here:

Lemma 5.7. *Consider a non-trivially Σ-connected database D, where Σ is a set of FDs. It holds that*

$$|\mathrm{CORep}(D, \Sigma)| \;=\; |\mathsf{IS}(\mathrm{CG}(D, \Sigma))|.$$

Proof. We establish the lemma by showing that actually $\mathsf{CORep}(D,\Sigma) = \mathsf{IS}(\mathsf{CG}(D,\Sigma))$.

($\subseteq$) Let $D' \in \mathsf{CORep}(D,\Sigma)$. 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 $\mathsf{CG}(D,\Sigma)$. Hence, D' is an independent set of $\mathsf{CG}(D,\Sigma)$.

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

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

- $L_0 = D'$.
- For each $i \geq 1$, $L_i = \{f \in D \mid f \notin L_0 \cup \ldots \cup L_{i-1}$ and there is $f' \in L_{i-1}$ such that $\{f, f'\} \not\models \Sigma\}$.

Observe that, since $\mathsf{CG}(D,\Sigma)$ 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 $L_i = \{f_1^i, \ldots, f_{|L_i|}^i\}$, for each $i \in [n]$. We now construct the desired sequence s as follows. We let the first $|L_n|$ operations be $-f_1^n, \ldots, -f_{|L_n|}^n$. 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 $op_{|L_n|+1}, \ldots, op_{|L_n|+|L_{n-1}|}$ will be $-f_1^{n-1}, \ldots, -f_{|L_{n-1}|}^{n-1}$, 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 $-f_1^1, \ldots, -f_{|L_1|}^1$. Again, these operations are justified since, by definition, each of the facts $f_1^1, \ldots, f_{|L_1|}^1$ is in conflict with some fact from $L_0 = D'$. Summing up, s is

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

We have that $s \in \mathsf{RS}(D,\Sigma)$ and that $s(D) = D' \models \Sigma$. Hence, $s \in \mathsf{CRS}(D,\Sigma)$, which implies $D' \in \mathsf{CORep}(D,\Sigma)$, 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 $\mathsf{CG}(D,\Sigma)$ 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 $-f_1^n, \ldots, -f_{|L_n|}^n$. We continue with the remaining strata L_{n-1} to L_2 as before. The last $|L_1|$ operations are defined as $-f_1^1, \ldots, -f_{|L_1|-1}^1, -\{f_{|L_1|}^1, f^*\}$. Note that, by definition of L_1, every fact $f_1^1, \ldots, f_{|L_1|-1}^1$ 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 $s \in \mathsf{RS}(D,\Sigma)$ and $s(D) = D' = \emptyset \models \Sigma$. Hence, $s \in \mathsf{CRS}(D,\Sigma)$, which in turn implies that $D' \in \mathsf{CORep}(D,\Sigma)$, as needed. □

We can now proceed with the two steps of the proof of item (3), which correspond to Proposition 5.8 and Lemma 5.9, respectively. Note that both results are essentially dealing with the following counting problem for a set Σ of FDs:

PROBLEM :	$\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma)$
INPUT :	A non-trivially Σ-connected database D.
OUTPUT :	The number $\lvert\mathsf{CORep}(D,\Sigma)\rvert$.

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 5.8 given below is more compact than the one given in the main body of the article since we explicitly use the name $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma)$.

Proposition 5.8. *Unless* RP = NP*, there exists a set* Σ *of keys over a singleton schema such that* $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma)$ *does not admit an FPRAS.*

Before giving the proof of Proposition 5.8, 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 :	$\sharp\mathsf{IS}_\Delta$
INPUT :	An undirected graph G of degree Δ.
OUTPUT :	The number $\lvert\mathsf{IS}(G)\rvert$.

We know from [Sly 2010] that the following holds:

Proposition B.3. *For every* $\Delta \geq 6$*, unless* RP = NP*,* $\sharp\mathsf{IS}_\Delta$ *does not admit an FPRAS.*

Note that the above result states the inapproximability of $\sharp\mathsf{IS}_\Delta$ for arbitrary, not necessarily non-trivially connected graphs. However, for showing Proposition 5.8, we need the stronger version of Proposition B.3 that establishes the inapproximability of $\sharp\mathsf{IS}_\Delta$ even for non-trivially connected graphs. Let $\sharp\mathsf{IS}_\Delta^{\mathsf{con}}$ be the problem defined as $\sharp\mathsf{IS}_\Delta$ with the difference that the input is a non-trivially connected undirected graph. We proceed to show the following:

Lemma B.3. *For every* $\Delta \geq 6$*, unless* RP = NP*,* $\sharp\mathsf{IS}_\Delta^{\mathsf{con}}$ *does not admit an FPRAS.*

Proof. By contradiction, assume that $\sharp\mathsf{IS}_\Delta^{\mathsf{con}}$ 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 $\mathsf{A}(G, \epsilon, \delta)$ such that

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

From this, we can construct an FPRAS A′ for $\sharp\mathsf{IS}_\Delta$ as follows. Given a graph $G = (V, E)$ of degree Δ, let the connected components, i.e., the maximal connected subgraphs, of G be $\mathsf{CC}_1, \ldots, \mathsf{CC}_n$ with $\mathsf{CC}_i = (V_i, E_i)$. Furthermore, we assume, without loss of generality, that $\mathsf{CC}_1, \ldots, \mathsf{CC}_\ell$ are all the trivially connected components of G, for $\ell \leq n$. Given G, $\epsilon > 0$, and $0 < \delta < 1$, A′ is defined as

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

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

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

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

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

Finally, we know (see, e.g., [Jerrum et al. 1986]) that the following hold: for $0 \leq x \leq 1$ and $m \geq 1$,

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

Consequently,

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

Hence, A' fulfills 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 $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta}$, for each $i \in [n]$, the random variable $\mathsf{A}(\mathsf{CC}_i, \frac{\epsilon}{2(n-\ell)}, \frac{\delta}{2(n-\ell)})$ can be computed in polynomial time w.r.t. CC_i, $\frac{|V|}{\epsilon}$, and $\log(\frac{|V|}{\delta})$. Since A' multiplies at most $|V|$ such random variables, A' is an FPRAS for $\sharp\mathsf{IS}_{\Delta}$, which contradicts Proposition B.3. □

With Lemma B.3 in place, we can now prove Proposition 5.8.

Proof of Proposition 5.8. Consider a non-trivially connected undirected graph $G = (V, E)$ with degree $\Delta = 6$. Let $\mathbf{S}$ be the schema consisting of the single relation name $\{R/\Delta + 1\}$ with $(A_1, \ldots, A_{\Delta+1})$ being the tuple of attributes of R, and $\Sigma_K = \{\phi_1, \ldots, \phi_{\Delta+1}\}$ a set of keys over $\mathbf{S}$, where $\phi_i = R : A_i \to \mathsf{att}(R)$ for each $i \in [\Delta + 1]$. We show that we can construct a non-trivially Σ_K-connected database D_G in polynomial time in $\|G\|$ such that $|\mathsf{IS}(G)| = |\mathsf{CORep}(D_G, \Sigma_K)|$. Hence, the existence of an FPRAS for $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma_K)$ would imply the existence of an FPRAS for $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta}$, which in turn contradicts Lemma B.3.

The database D_G should enjoy the following key property: there exists a bijection $\mu : V \to D_G$ 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 $|\mathsf{IS}(G)| = |\mathsf{IS}(\mathsf{CG}(D_G, \Sigma_K))|$, which, together with the fact that G is non-trivially connected, and hence, D_G is non-trivially Σ_K-connected, implies that $|\mathsf{IS}(G)| = |\mathsf{CORep}(D_G, \Sigma_K)|$ by Lemma 5.7. 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 $\|G\|$ [Misra and Gries 1992]. Therefore, we are able to efficiently assign the colours $C = \{c_1, \ldots, c_{\Delta+1}\}$ to the edges of G in such a way that none of the nodes belongs to two distinct edges of the same colour. Let $M : E \to C$ be such a coloring. The database D_G is such that $\mathsf{dom}(D_G) = E \cup F$, where F is a finite a set of constants with $E \cap F = \emptyset$, and consists of the following facts:

(1) for each node $v \in V$, we add to D_G a fact of the form $R\left(a^v_1, \ldots, a^v_{\Delta+1}\right)$, 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 $a^u_i = a^v_i = e$, and
   (b) for every a^v_i not defined in the above step, we let $a^v_i = f$ for some constant $f \in F$ only to be used once.

For brevity, we write $R(\bar{a}^v)$ for $R(a^v_1, \ldots, a^v_{\Delta+1})$. Note that every a^v_i is well-defined since v has at most one edge of colour c_i, and thus, a^v_i 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 the database D_G:

Lemma B.3. *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 $R(a^u_1, \ldots, a^u_{\Delta+1})$ and $R(a^v_1, \ldots, a^v_{\Delta+1})$, having

$a_i^u = a_i^v = e$. Since there are no multiple edges between two nodes, 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 $R(a_1^u, \ldots, a_{\Delta+1}^u)$ and $R(a_1^v, \ldots, a_{\Delta+1}^v)$ that together violate $\phi_i = R : A_i \to \mathsf{att}(R)$ 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, $a_i^u = a_i^v$ because $\{u, v\}$ is an edge of G, and the claim follows. □

By Lemma B.3, we get that $|\mathsf{IS}(G)| = |\mathsf{IS}(\mathsf{CG}(D_G, \Sigma_K))|$, and that D_G is non-trivially Σ-connected. Hence, by Lemma 5.7, $|\mathsf{IS}(G)| = |\mathsf{IS}(\mathsf{CG}(D_G, \Sigma_K))| = |\mathsf{CORep}(D_G, \Sigma_K)|$, and the claim follows.

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 5.9. The formal statement, already given in the main body, and its proof follow. Note that the statement of Lemma 5.9 given below is more compact than the one given in the main body since we explicitly use the name of the problem $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma_K)$, where Σ_K is the set of keys provided by Proposition 5.8.

Lemma 5.9. *Assume that* $\mathsf{RepFreq}(\Sigma, Q)$ *admits an FPRAS, for every set* Σ *of FDs and CQ* Q*. Then,* $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma_K)$ *admits an FPRAS.*

Proof. By Proposition 5.8, unless RP = NP, Σ_K is a set of keys over a schema $\{R/n\}$ such that $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma_K)$ does not admit an FPRAS. Let $\mathbf{S} = \{R'/m\}$, where $m = n + 2$, and, assuming that $(A_1, \ldots, A_n)$ is the tuple of attributes of R, let $(A, B, A_1, \ldots, A_n)$ be the tuple of attributes of R'. We first show that there exist a set Σ_F of FDs over $\mathbf{S}$ and a Boolean CQ Q_F over $\mathbf{S}$ 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 $\mathbf{S}$ such that

$$\mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ()) \;=\; \frac{1}{|\mathsf{CORep}(D, \Sigma_K)| + 1}. \tag{$*$}$$

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

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

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

We define the set Σ_F of FDs over $\mathbf{S}$ as follows:

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

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 $\mathsf{dom}(D)$:

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

For brevity, let $f^* = R'(a, a, \ldots, a)$. It is not difficult to verify that the number $|\mathsf{CORep}(D_F, \Sigma_F)|$ is

$$|\{D' \in \mathsf{CORep}(D_F, \Sigma_F) \mid f^* \in D'\}| \;+\; |\{D' \in \mathsf{CORep}(D_F, \Sigma_F) \mid f^* \notin D'\}|,$$

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 $R' : A \to B$. Moreover, one can easily devise a sequence $s \in \mathsf{CRS}(D_F, \Sigma_F)$ 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 5.7, $\mathsf{CORep}(D_F, \Sigma_F) = \mathsf{IS}(\mathsf{CG}(D_F, \Sigma_F))$. Since $\{f^*\}$ is the only candidate repair of D_F containing f^*, and thus, the only independent set of $\mathsf{CG}(D_F, \Sigma_F)$ containing f^*, the set of candidate repairs without f^*, i.e., $\{D' \in \mathsf{CORep}(D_F, \Sigma_F) \mid f^* \notin D'\} = \mathsf{IS}(\mathsf{CG}(D_F \setminus \{f^*\}, \Sigma_F))$. 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 5.7, $\mathsf{IS}(\mathsf{CG}(D_F \setminus \{f^*\}, \Sigma_F)) = \mathsf{CORep}(D_F \setminus \{f^*\}, \Sigma_F)$.

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

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

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

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

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 $\mathsf{CORep}(D_F, \Sigma_F)$ that satisfies the query Q_F is $\{f^*\}$, i.e.,

$$\mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ()) \;=\; \frac{1}{|\mathsf{CORep}(D_F, \Sigma_F)|}.$$

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

Building the FPRAS. We proceed to devise an FPRAS for $\sharp\mathsf{CORep}^{\mathsf{con}}(\Sigma_K)$ by exploiting the equation (∗), the fact that D_F can be constructed in polynomial time, and the FPRAS A' for $\mathsf{RepFreq}(\Sigma_F, Q_F)$, 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 $r = \max\left\{\frac{1-\epsilon'}{2\cdot(1+2^{|D|})}, \mathsf{A}'(D_F, (), \epsilon', \delta)\right\}$;
(4) Output $\frac{1}{r} - 1$.

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

$$\Pr\left((1-\epsilon') \cdot \mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ()) \leq \mathsf{A}'(D_F, (), \epsilon', \delta) \leq (1+\epsilon') \cdot \mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ())\right) \geq 1 - \delta.$$

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

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

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

$$\Pr\left((1-\epsilon') \cdot \mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ()) \leq \mathsf{A}'(D_F, (), \epsilon', \delta) \leq (1+\epsilon') \cdot \mathsf{repfreq}_{\Sigma_F, Q_F}(D_F, ())\right) = \Pr(E),$$

where E is the event

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

Note that $|\mathsf{CORep}(D, \Sigma_K)|$ is at most the the number of all possible subsets of D. Hence,

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

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

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

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

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

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

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

For short, let X be the random variable $\frac{1}{\max\{p, \mathsf{A}'(D_F, (), \epsilon', \delta)\}}$. 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

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

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

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

By expanding the products in the above expression, we obtain

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

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

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

Thus, the above probability is less or equal than

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

which coincides with

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

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

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

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

It is now straightforward to see that from Proposition 5.8 and Lemma 5.9, we can conclude item (3) of Theorem 5.1.

C Proofs of Section 6

C.1 Proof of Lemma 6.2

The proof of Lemma 6.2 has already been given in the main body of the article. Here is an example that illustrates the algorithm described in the proof of Lemma 6.2.

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

$$\begin{array}{lll} -f_{1,1}, -f_{1,2} & -f_{1,1}, -f_{1,3} & -f_{1,1}, -\{f_{1,2}, f_{1,3}\} \\ -f_{1,2}, -f_{1,1} & -f_{1,2}, -f_{1,3} & -f_{1,2}, -\{f_{1,1}, f_{1,3}\} \\ -f_{1,3}, -f_{1,1} & -f_{1,3}, -f_{1,2} & -f_{1,3}, -\{f_{1,1}, f_{1,2}\} \\ -\{f_{1,1}, f_{1,2}\} & -\{f_{1,1}, f_{1,3}\} & -\{f_{1,2}, f_{1,3}\}. \end{array}$$

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, which has one pair removal:

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

For the first block, we have that

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

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, which has two facts, it holds that

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

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

$$P_1^{0,0} = S_3^{e,0} = 0 \qquad P_1^{1,0} = S_3^{ne,0} = 6 \qquad P_1^{0,1} = S_3^{e,1} = 3 \qquad P_1^{1,1} = S_3^{ne,1} = 3.$$

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 no pair removals, $s(D) \cap B_i \neq \emptyset$ for $i \in \{1,2\}$; hence, $P_2^{0,k} > 0$ only for $k = 2$. Thus,

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

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

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

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

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

Finally,

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

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

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

We conclude that

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

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

C.2 Proof of Lemma 6.4

Lemma 6.4. *Consider a set Σ of primary keys and a CQ $Q(\bar{x})$. For every database D and tuple $\bar{c} \in \mathrm{dom}(D)^{|\bar{x}|}$ with $\mathrm{seqfreq}_{\Sigma,Q}(D, \bar{c}) > 0$, it holds that $\mathrm{seqfreq}_{\Sigma,Q}(D, \bar{c}) \geq \frac{1}{(2 \cdot \|D\|)^{\|Q\|}}$.*

Proof. The proof is very similar to the proof of Lemma 5.5, 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 with $h(\bar{x}) = \bar{c}$, we denote by $h(Q)$ the set $\{R_i(h(\bar{y}_i)) \mid i \in [n]\}$. Here, we denote by $S^{\mathrm{e}}_{D,\Sigma,h(Q)}$ the set of sequences $s \in \mathrm{CRS}(D, \Sigma)$ such that $s(D) \cap B_j = \emptyset$ for at least one block of $\{B_1, \ldots, B_m\}$ (recall that these are

the blocks that contain the facts of $h(Q)$), and by $S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ the set of sequences $s \in \mathsf{CRS}(D,\Sigma)$ such that $s(D) \cap B_j \neq \emptyset$ for every block of $\{B_1, \ldots, B_m\}$.

Now, for every sequence $s \in S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ 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 $s \in S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ to a sequence $s' \in S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ by replacing the last operation of s over each such $B_j \in \{B_1, \ldots, B_m\}$ with an operation that removes only one of the facts of the pair, i.e., either f or g. Hence, if $s(D) \cap B_j = \emptyset$ for precisely ℓ of the blocks of $\{B_1, \ldots, B_m\}$, the sequence s is mapped to 2^ℓ distinct sequences of $S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$. Note that this mapping essentially defines a many-to-many relation from $S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ to $S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ that collects all the pairs of sequences $(s, s') \in S^{\mathsf{e}}_{D,\Sigma,h(Q)} \times S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ such that s is transformed into s' according to the above transformation.

Similarly to the proof of Lemma 5.5, for every sequence $s' \in S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$, there are $2^m - 1$ sequences $s \in S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ that are mapped to it. This is because the sequence s' determines all the operations over the blocks outside $\{B_1, \ldots, B_m\}$, and for each block $B_j \in \{B_1, \ldots, B_m\}$, 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 $\{B_1, \ldots, B_m\}$ and 2^m possibilities in total. And, again, we have to disregard the possibility that is equivalent to s' itself.

Therefore, we have that

$$\left|S^{\mathsf{e}}_{D,\Sigma,h(Q)}\right| \;\leq\; (2^m - 1) \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|$$

and

$$|\mathsf{CRS}(D,\Sigma)| \;=\; \left|S^{\mathsf{e}}_{D,\Sigma,h(Q)}\right| + \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| \;\leq\; (2^m - 1) \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| + \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right| \;=\; 2^m \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|.$$

As said above, each sequence s of $S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ can be mapped to 2^ℓ distinct sequences of $S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$, where ℓ is the number of blocks in $\{B_1, \ldots, B_m\}$ for which $E \cap B_j = \emptyset$. Moreover, there are sequences $s' \in S^{\mathsf{ne}}_{D,\Sigma,h(Q)}$ such that no sequence $s \in S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ is mapped to s'. These are the sequences s' where the last operation of s' over every block of $\{B_1, \ldots, B_m\}$ is a pair removal (but s' keeps some fact of each B_j). Hence, $(2^m - 1) \times |S^{\mathsf{ne}}_{D,\Sigma,h(Q)}|$ is only an upper bound on $|S^{\mathsf{e}}_{D,\Sigma,h(Q)}|$. Since all the facts of a single block are symmetric,

$$|\{s \in \mathsf{CRS}(D,\Sigma) \mid h(Q) \subseteq s(D)\}| \;=\; \frac{1}{|B_1| \times \cdots \times |B_m|} \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|$$

and we conclude that

$$\begin{aligned}
\frac{|\{s \in \mathsf{CRS}(D,\Sigma) \mid h(Q) \subseteq s(D)\}|}{|\mathsf{CRS}(D,\Sigma)|} &\geq \frac{\frac{1}{|B_1| \times \cdots \times |B_m|} \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|}{2^m \times \left|S^{\mathsf{ne}}_{D,\Sigma,h(Q)}\right|} \\
&= \frac{1}{|B_1| \times \cdots \times |B_m| \times 2^m} \\
&\geq \frac{1}{|D|^m \times 2^m} \\
&\geq \frac{1}{(2 \cdot |D|)^{|Q|}} \\
&\geq \frac{1}{(2 \cdot ||D||)^{||Q||}}.
\end{aligned}$$

Since all the sequences of $\{s \in \mathsf{CRS}(D,\Sigma) \mid h(Q) \subseteq s(D)\}$ are such that $h(Q) \subseteq s(D)$ and $h(\bar{x}) = \bar{c}$, the claim follows. □

We proceed to give an example that illustrates the argument given in the proof of Lemma 6.4.

Example C.2. Consider the database D, the set Σ of keys, the query Q, and the homomorphism h from Example 5.6. Recall that $h(Q) = \{R(a_1, b_1)\}$. The set $S^{\mathsf{e}}_{D,\Sigma,h(Q)}$ 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 6.4, this sequence is mapped to the sequences

$$-f_{1,2}, -f_{3,1}, -f_{1,1} \qquad \text{and} \qquad -f_{1,2}, -f_{3,1}, -f_{1,3}$$

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

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

These are 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:

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

Moreover, as we have seen in Example C.1, we have that

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

Indeed, it holds that

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

as claimed.

D Proofs of Section 7

Recall that the main result of Section 7 is the following:

Theorem 7.1. *The following hold:*

(1) There exist a set Σ of primary keys and a CQ Q such that $\mathrm{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$ is $\sharp\mathrm{P}$-hard.
(2) For every set Σ of keys and CQ Q, $\mathrm{OCQA}(\Sigma, M^{\mathsf{uo}}_{\Sigma}, Q)$ admits an FPRAS.

Item (1) has been shown in the main body of the article. Concerning item (2), we have shown in the main body of the article that there exists an efficient sampler (see Lemma 7.2) and that the probability that we want to approximate is not "too small" (see Proposition 7.3), which together imply the existence of the desired FPRAS. However, the proof of Proposition 7.3 given in the main body of the article considers only the case where the image of the query is a singleton. It remains to prove the arbitrary case where where the image of the query consists of more than one atom. Before giving the proof, let us first recall the result in question.

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 $\bar{c} \in \mathsf{dom}(D)^{|\bar{x}|}$ with $\mathrm{P}_{M^{\mathsf{uo}}_{\Sigma},Q}(D,\bar{c}) > 0$, it holds that*

$$\mathrm{P}_{M^{\mathsf{uo}}_{\Sigma},Q}(D,\bar{c}) \;\geq\; \frac{1}{\mathrm{pol}(||D||)}.$$

Proof. 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 :-. 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 $\mathsf{P}_{M^{\mathsf{uo}}_\Sigma,Q}(D, \bar{c}) = 0$. Assume now that such a homomorphism h exists. The claim for the case where $|h(Q)| = 1$ has been shown in the main body of the article. We now proceed to generalize it to the case where $|h(Q)| = m$, for some $m \geq 1$.

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 $s \in \mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D))$ be a repairing sequence that removes r of the facts of $h(Q)$, for some $r \in [m]$:

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

where the operations $op_{i_1}, \ldots, op_{i_r}$ 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 $s' \in \mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D))$ 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 $o^\star_{i_j}$ that removes only the fact g. At the end of the sequence s', we add operations $op'_1, \ldots, op'_\ell$ that remove the facts that are in conflict with one of the facts of $h(Q)$ that appears in $s(D)$. As we have already explained in the proof for the case where $|h(Q)| = 1$, 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 $\mathbf{P}((op_1, \ldots, op_{j-1}), (op_1, \ldots, op_j))$, for $2 \leq j \leq i_1 - 1$, is not affected by the decision to remove or keep a certain fact at the i_1th step. However, for $j \geq i_1$, the probability of applying the operation op_j might decrease in the sequence s' compared with the sequence s because the additional facts of $h(Q)$, which 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 in the main body of the article, if the number of (D^s_{j-1}, 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, that is, the facts of $h(Q)$, can increase this number by at most $5km\sqrt{N_j}$. We 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 can conclude that

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

with

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

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, since 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, then an upper bound on the number of (D', Σ)-justified operations, which 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

$$\frac{1}{c} \;\geq\; \frac{1}{(rk+r+1)^2} \times \frac{1}{(rk+r)^2} \times \frac{1}{(rk+r-1)^2} \times \cdots \times \frac{1}{3} \;\geq\; \frac{1}{((rk+r+1)^2)!} \geq \frac{1}{((mk+m+1)^2)!}$$

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

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

Hence, with

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

we have that

$$\pi(s) \;\leq\; \mathrm{pol}''(||D||) \times \pi(s').$$

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 $r \in [m]$ of facts of $h(Q)$. Furthermore, 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 alone, while the remaining $r-\ell$ facts will be removed jointly with another fact. Given s', there are at most $\binom{|D|-1}{\ell}$ possible choices for the positions of the additional singleton deletions, since the length of a sequence is bounded by $|D|-1$, and $\ell!$ ways to order these operations in the sequence. Moreover, there are at most $\binom{|D|-1}{r-\ell}$ possible choices for the individual fact removals from s' that are pair removals in s. Note that here we do not need to multiply by $(r-\ell)!$ since the ordering of these operations is determined by the sequence s'. Hence, the number of sequences that remove a fact of $h(Q)$ that are mapped to the sequence s' is bounded by

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

$$
\begin{aligned}
&\leq \sum_{r=1}^{|Q|} \sum_{\ell=0}^{|Q|} (e|Q|)^{\ell} \times (e(|D|-1))^{\ell} \times \ell! \times (e(|D|-1))^{|Q|-\ell} \\
&\leq |Q| \times (|Q|+1) \times (e|Q|)^{|Q|} \times (e(|D|-1))^{|Q|} \times |Q|! \times (e(|D|-1))^{|Q|} \\
&\leq (e|Q|)^2 \times (e|Q|)^{|Q|} \times (e(|D|-1))^{|Q|} \times |Q|! \times (e(|D|-1))^{|Q|} \\
&= (e|Q|)^{|Q|+2} \times (e(|D|-1))^{2|Q|} \times |Q|!
\end{aligned}
$$

This is clearly a polynomial in $||D||$. We define

$$
\mathrm{pol}'(||D||) \;=\; \mathrm{pol}''(||D||) \cdot \left((e|Q|)^{|Q|+2} \times (e(|D|-1))^{2|Q|} \times |Q|! \right).
$$

Then, similarly to the case where $|h(Q)| = 1$ (Equation 1), we have that

$$
\mathsf{P}_{D,M^{\mathrm{uo}}_{\Sigma},Q}(h) \;\geq\; \frac{1}{1+\mathrm{pol}'(||D||)}.
$$

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

$$
\mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D,\bar{c}) \geq \mathsf{P}_{D,M^{\mathrm{uo}}_{\Sigma},Q}(h) \;\geq\; \frac{1}{\mathrm{pol}(||D||)},
$$

which concludes our proof. □

D.1 Proof of Proposition 7.9

Unlike the case of keys, in the case of FDs, there is no polynomial lower bound on the target probability. We proceed to give the proof of this statement, which has been already stated in the main body of the article and we recall again here.

Proposition 7.9. *Consider the FD set* $\{R : A_1 \to A_2\}$ *over the schema* $\{R/3\}$ *and the Boolean CQ* $Ans() \coloneq R(0,0,0)$. *There exists a family* $\{D_n\}_{n\geq 1}$ *of databases such that*

$$
0 < \mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D_n,()) \leq \frac{1}{2^{|D_n|-1}}.
$$

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 $i \in \{1,\ldots,n-1\}$. 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 $0 < \mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D,())$ as the operational repair that keeps the fact $R(0,0,0)$ entails Q. We prove by induction on n, i.e., 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

$$
\mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D,()) \;\leq\; \frac{1}{2^{n-1}}.
$$

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

$$
\mathsf{P}_{M^{\mathrm{uo}}_{\Sigma},Q}(D,()) \;=\; \frac{1}{2^{1-1}} \;=\; 1.
$$

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

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

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 removed. For every $i \in \{1, \dots, p\}$, we denote by D_i the database $op_i(D)$. By the inductive hypothesis,

$$\mathsf{P}_{M^{\mathsf{uo}}_\Sigma, Q}(D_i, ()) \;\leq\; \frac{1}{2^{p-1}}.$$

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

$$\mathsf{P}_{M^{\mathsf{uo}}_\Sigma, Q}(D, ()) \;=\; \sum_{\substack{s \in \mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D)) \\ () \in Q(s(D))}} \pi(s) \;=\; \sum_{i=1}^{p} \left(\mathbf{P}(\varepsilon, (op_i)) \times \sum_{\substack{s_i \in \mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D_i)) \\ () \in Q(s_i(D_i))}} \pi(s_i) \right).$$

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

$$\mathsf{P}_{M^{\mathsf{uo}}_\Sigma, Q}(D_i, ()) \;=\; \sum_{\substack{s_i \in \mathsf{RL}(M^{\mathsf{uo}}_\Sigma(D_i)) \\ () \in Q(s_i(D_i))}} \pi(s_i) \;\leq\; \frac{1}{2^{p-1}}.$$

Therefore, we can conclude that

$$\mathsf{P}_{M^{\mathsf{uo}}_\Sigma, Q}(D, ()) \;\leq\; \sum_{i=1}^{p} \left(\frac{1}{1+2p} \times \frac{1}{2^{p-1}} \right) \;=\; \frac{p}{(1+2p) \times 2^{p-1}} \;=\; \frac{p}{2^{p-1} + p \times 2^p} \;\leq\; \frac{p}{p \times 2^p} \;=\; \frac{1}{2^p},$$

and the claim follows. □

E Proofs of Section 8

E.1 Proof of Proposition 8.3

Proposition 8.3. *For a database D and a set Σ of primary keys,* $\mathsf{CORep}^1(D, \Sigma) = \mathsf{rep}_\Sigma(D)$.

Proof. (⊆) Let $D' \in \mathsf{CORep}^1(D, \Sigma)$. We proceed to show that $D' \in \mathsf{rep}_\Sigma(D)$. Assuming that $\mathsf{blocks}_\Sigma(D) = \{B_1, \dots, B_n\}$, we observe that $D' \in \{\{f_1, \dots, f_n\} \mid (f_1, \dots, f_n) \in B_1 \times \cdots \times B_n\}$, i.e., D' keeps exactly one fact from each block $B_1, \dots, B_n$. Clearly, D' does not keep more than one fact from a block since, in this case, it would be inconsistent, which cannot be the case since it is a candidate operational repair. Moreover, during the construction of D', we cannot remove all the facts of a block with singleton deletions since the removal of the last fact f of a block would not be a justified operation as it does not violate Σ jointly with another fact (all facts that violate Σ together with f have already been removed). Consequently, by Proposition 8.1, we get that $D' \in \mathsf{rep}_\Sigma(D)$.

(⊇) Let $D' \in \mathsf{rep}_\Sigma(D)$. We proceed to show that $D' \in \mathsf{CORep}^1(D, \Sigma)$. Consider the sequence s' which removes all facts in $D \setminus D'$, one by one, in an arbitrary order. Clearly, $s'(D) = D'$. We need to show that, for every fact $f \in D \setminus D'$, the operation $-f$ in s' is indeed justified. Fix some $f \in D \setminus D'$.

Since $D' \cup \{f\}$ is inconsistent while D' is consistent (by the maximality of D'), there must exist a violation of the form $\{f, g\}$ in $D' \cup \{f\}$, for some primary key in Σ and fact $g \in D'$; thus, $-f$ is a justified operation. We get that $s' \in \mathrm{CRS}^1(D, \Sigma)$, and hence, $D' \in \mathrm{CORep}^1(D, \Sigma)$. □

E.2 Proof of Lemma 8.8

Lemma 8.8. *Unless* RP = NP*, for each* $\Delta \geq 6$*, it holds that* $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$ *has no FPRAS.*

Proof. By contradiction, assume that $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$ admits an FPRAS, for some $\Delta \geq 6$. We then show that $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta}$ admits an FPRAS, contradicting Lemma B.3. Assume that A is an FPRAS for $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$. Let A′ be the randomized algorithm that, given a non-trivially connected undirected graph G, $\epsilon > 0$ and $0 < \delta < 1$, is such that $\mathsf{A}'(G, \epsilon, \delta) = \mathsf{A}(G, \epsilon, \delta) + 1$. We proceed to show that A′ is an FPRAS for $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta}$. Since A is an FPRAS for $\sharp\mathsf{IS}^{\mathsf{con}}_{\Delta,\neq\emptyset}$,

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

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

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

Since

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

and

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

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

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

and

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

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

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

and the claim follows. □

E.3 Proof of Theorem 8.10

Theorem 8.10. *The following hold:*

(1) There exist a set Σ *of primary keys and a CQ* Q *such that* $\mathrm{OCQA}(\Sigma, M^{\mathsf{us},1}_{\Sigma}, Q)$ *is* $\sharp$P*-hard.*

(2) For every set Σ *of primary keys and CQ* Q*,* $\mathrm{OCQA}(\Sigma, M^{\mathsf{us},1}_{\Sigma}, Q)$ *admits an FPRAS.*

As in the case of uniform repairs, we show that OCQA and classical CQA coincide, which in turn allows us to inherit intractability and approximability results from classical CQA. We show that:

Proposition E.1. *For every database* D*, set* Σ *of primary keys, CQ* $Q(\bar{x})$*, and tuple* $\bar{c} \in \mathrm{dom}(D)^{|\bar{x}|}$*,*

$$\mathsf{seqfreq}^1_{\Sigma,Q}(D,\bar{c}) = \mathsf{rfreq}_{\Sigma,Q}(D,\bar{c}).$$

Proof. It suffices to show that the set of operational repairs and the set of classical repairs coincide, and that the Markov chain generator $M^{\mathsf{us},1}_{\Sigma}$ induces the uniform distribution over the operational repairs. We start with the first property.

Let D be a database and Σ a set of primary keys. By Proposition 8.3, $\mathrm{CORep}^1(D, \Sigma) = \mathsf{rep}_{\Sigma}(D)$. It thus suffices to show that indeed $\mathrm{ORep}(D, M^{\mathsf{us},1}_{\Sigma}) = \mathrm{CORep}^1(D, \Sigma)$. For every database D we have that every complete repairing sequence of singleton deletions is a reachable leaf of the Markov chain

$M_\Sigma^{\mathsf{us},1}(D)$ by definition; thus, we know that every candidate operational repair $D' \in \mathsf{CORep}^1(D,\Sigma)$ is also an operational repair w.r.t. the Markov chain generator $M_\Sigma^{\mathsf{us},1}$, i.e. $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{us},1})$. The opposite direction follows directly from the definition of operational repair, that is, every operational repair is necessarily also a candidate operational repair.

We continue to show that $M_\Sigma^{\mathsf{us},1}$ induces the uniform distribution over the repairs. Let D be a database and Σ a set of primary keys. From the definition of $M_\Sigma^{\mathsf{us},1}$, we know that for every $s \in \mathsf{CRS}^1(D,\Sigma)$, assuming that π is the leaf distribution of $M_\Sigma^{\mathsf{us},1}(D)$, it holds that $\pi(s) = \frac{1}{|\mathsf{CRS}^1(D,\Sigma)|}$. As the probability of an operational repair $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{us},1})$ is defined as $\sum_{s \in \mathsf{RL}(M_\Sigma^{\mathsf{us},1}) \text{ and } D'=s(D)} \pi(s)$, it suffices to show that every repair is obtained via the same number of complete repairing sequences of singleton operations. We have already argued in the proof of Proposition 8.1 that D' can be obtained via a repairing sequence s which removes the facts in $D \setminus D'$ one by one in an arbitrary order. Hence, every operational repair D' gives rise to $(|D \setminus D'|)!$ many different repairing sequences leading to that repair. Thus, for every repair $D' \in \mathsf{ORep}(D, M_\Sigma^{\mathsf{us},1})$, we have that

$$\mathsf{P}_{D,M_\Sigma^{\mathsf{us},1}}(D') \;=\; \frac{(|D \setminus D'|)!}{|\mathsf{CRS}^1(D,\Sigma)|} \;=\; \frac{(|D \setminus D'|)!}{(|D \setminus D'|)! \cdot |\mathsf{ORep}(D, M_\Sigma^{\mathsf{us},1})|} \;=\; \frac{1}{|\mathsf{ORep}(D, M_\Sigma^{\mathsf{us},1})|},$$

which concludes our proof. □

Received 22 August 2023; revised 4 July 2025; accepted 6 October 2025