

Uniform Reliability for Unbounded Homomorphism-Closed Graph Queries

Antoine Amarilli   

LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Abstract

We study the uniform query reliability problem, which asks, for a fixed Boolean query Q , given an instance I , how many subinstances of I satisfy Q . Equivalently, this is a restricted case of Boolean query evaluation on tuple-independent probabilistic databases where all facts must have probability $1/2$. We focus on graph signatures, and on queries closed under homomorphisms. We show that for any such query that is *unbounded*, i.e., not equivalent to a union of conjunctive queries, the uniform reliability problem is $\#P$ -hard. This recaptures the hardness, e.g., of s - t connectedness, which counts how many subgraphs of an input graph have a path between a source and a sink.

This new hardness result on uniform reliability strengthens our earlier hardness result on probabilistic query evaluation for unbounded homomorphism-closed queries [2]. Indeed, our earlier proof crucially used facts with probability 1, so it did not apply to the unweighted case. The new proof presented in this paper avoids this; it uses our recent hardness result on uniform reliability for non-hierarchical conjunctive queries without self-joins [3], along with new techniques.

2012 ACM Subject Classification Theory of computation \rightarrow Database query processing and optimization (theory)

Keywords and phrases Uniform reliability, $\#P$ -hardness, probabilistic databases

Digital Object Identifier 10.4230/LIPIcs.ICDT.2023.14

Related Version *Full Version*: <https://arxiv.org/abs/2209.11177> [1]

Funding Partially supported by the ANR project ANR-18-CE23-0003-02 (“CQFD”).

Acknowledgements I am grateful to Mikaël Monet, Charles Paperman, and Martin Retaux for helpful discussions about this research. Thanks to the reviewers for their helpful feedback.

1 Introduction

A long line of research [14] has investigated how to extend relational databases with probability values. The most common probabilistic model, called *tuple-independent databases* (TID), annotates each fact of the input database with an independent probability of existence. The *probabilistic query evaluation* (PQE) problem then asks for the probability that a fixed Boolean query is true in the resulting product distribution on possible worlds. The PQE problem has been historically studied for conjunctive queries (CQs) and unions of conjunctive queries (UCQs). This study led to the dichotomy result of Dalvi and Suciu [5], which identifies a class of *safe UCQs* for which the problem can be solved in PTIME:

► **Theorem 1.1** ([5]). *Let Q be a UCQ. Consider the PQE problem for Q which asks, given a TID I , to compute the probability that Q holds on I . This problem is in PTIME if Q is safe, and $\#P$ -hard otherwise.*

This result has been extended in several ways, to apply to some queries featuring negation [6], disequality (\neq) joins [10], or inequality ($<$) joins [11]. More recently, two new directions have been explored. First, our work with Ceylan [2] extended the study from UCQs to the broader class of *homomorphism-closed* queries. This class captures recursive queries such as regular path queries (RPQs) or Datalog (without inequalities or negation). In [2], we



© Antoine Amarilli;

licensed under Creative Commons License CC-BY 4.0

26th International Conference on Database Theory (ICDT 2023).

Editors: Floris Geerts and Brecht Vandevoort; Article No. 14; pp. 14:1–14:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

focused on homomorphism-closed queries that were *unbounded*, i.e., not equivalent to a UCQ. We showed that PQE is #P-hard for *any* such query, though for technical reasons the result only applies to graphs, i.e., arity-two signatures. This extended the above dichotomy to the full class of homomorphism-closed queries (on arity-two signatures).

Second, the dichotomy has been extended from PQE to restricted problems which do not allow arbitrary probabilities on the TID. Kenig and Suciu [8] have shown that the dichotomy of [5] still held for the so-called *generalized model counting* problem, where the allowed probabilities on tuples are only 0 (the tuple is missing), 1/2, or 1; this is in contrast with the original proof of the dichotomy, which uses arbitrary probabilities. Our result in [2] already held for the generalized model counting problem. What is more, for a subclass of the unsafe queries, they showed that hardness still held for the *model counting problem*, where the probabilities are either 0 or 1/2. Independently, with Kimelfeld [3], we have shown hardness of the same problem for the incomparable class of non-hierarchical CQs without self-joins. Rather than model counting, we called this the *uniform reliability* (UR) problem, following the terminology in the work of Grädel, Gurevich, and Hirsch [7].

In our opinion, this uniform reliability problem is interesting even outside of the context of probabilistic databases: we simply ask, for a fixed query Q , given a database instance I , *how many* subinstances of I satisfy Q . The UR problem also relates to computing the *causal effect* and *Shapley values* in databases [13, 9, 3]. What is more, UR for homomorphism-closed queries captures existing counting problems on graphs, such as *st-connectedness* [15] which asks how many subgraphs of an input graph contain a path between a source and a sink.

The ultimate goal of these two lines of work would be to classify the complexity of *uniform reliability*, across *all homomorphism-closed queries*. Specifically, one can conjecture:

► **Conjecture 1.2.** *Let Q be a homomorphism-closed query on an arbitrary signature. The uniform reliability problem for Q is in PTIME if Q is a safe UCQ, and #P-hard otherwise.*

To establish this, there are three obstacles to overcome. First, in the case where Q is a UCQ, one would need to establish the hardness of UR for all unsafe UCQs, extending the work of Kenig and Suciu [8]. Second, when Q is unbounded, one would need to adapt the methods of [2] to apply to UR rather than PQE. Third, the methods of [2] would need to be extended from graph signatures to arbitrary arity signatures.

Result statement. In this paper, we address the second difficulty and show the following, which extends the main result of [2] from PQE to UR, and brings us closer to Conjecture 1.2:

► **Theorem 1.3 (Main result).** *Let Q be an unbounded homomorphism-closed query on an arity-two signature. The uniform reliability problem for Q is #P-hard.*

The proof of this result has the same high-level structure as in [2], but there are significant new technical challenges to overcome. In particular, we now reduce from different problems, whose hardness rely (among other things) on the hardness of uniform reliability for the query $R(x), S(x, y), T(y)$, shown in [3]. The impossibility to assign a probability of 1 to facts also makes reductions much more challenging: intuitively, as all facts can now be missing, there is no longer a clear connection between the possible worlds of the source problem and the possible worlds of the database built in the reduction. We use multiple tools to work around this, for instance a *saturation* technique that creates a large but polynomial number of copies of some facts and argues that their absence is sufficiently unlikely to be negligible. As saturation cannot apply to unary facts, we also need to identify so-called *critical models*, a more elaborate variant of a notion in [2], minimizing carefully-chosen weight criteria.

We give a high-level structure of the proof below as it is presented in the rest of the paper, and comment in more detail on how the techniques relate to our earlier work [2].

Paper structure. We give preliminaries and the formal definition of UR in Section 2, along with the two problems from which we reduce: one problem on bipartite graphs from [3], and one variant of a connectivity problem of [15]. We show that they are #P-hard in [1].

We then review notions from [2] in Section 3: the *dissociation* operation on instances, and the notion of a *tight edge*, which makes the query false when we apply dissociation to it. We invoke a result from [2] showing that tight edges always exist for unbounded queries. This is the only place where we use the unboundedness of the query, and is unfortunately the only result from [2] that can be used as-is. Some other notions are reused and extended from [2] but they are always re-defined and re-proved in a self-contained way in the present paper.

We then present in Section 4 the notion of a *critical model*, as a model of the query which is *subinstance-minimal* and features a tight edge which is minimal by optimizing three successive quantities: *weight*, *extra weight*, and *lexicographic weight*. The notion of *weight* is from [2], the two other notions relate to *side weight* from [2] but significantly extend it. We show in this section that a query having a model with a tight edge also has a critical model.

We then move on to the hardness proof. As in [2], there are two cases: a *non-iterable* case where we reduce from the problem on bipartite graphs, and an *iterable* case where we reduce from the connectivity problem. In Section 5, we formally define the notion of iteration (essentially identical to the notion in [2]) and show hardness when there is a non-iterable critical model. The coding used in the reduction extends that of [2] with the *saturation* technique of creating a large number of copies of some elements. There are many new technical challenges, e.g., proving that a polynomial number of copies suffices to make the absence of the facts sufficiently unlikely, and justifying that all the other facts are “necessary” for a query match, using in particular subinstance-minimality and the notion of extra weight.

Last, in Section 6, we show hardness in the case where all critical models are iterable. We first show that such models can be repeatedly iterated, and that the measure of *extra weight* must be zero in this case, allowing us to focus on the more precise criterion of lexicographic weight. Then we define the coding, which is similar to [2] up to technical modifications. The reduction does not use saturation but argues that all facts are “necessary” using the notion of lexicographic weight and a new *explosion* structure.

We then conclude in Section 7. The complete proofs are given in the full version [1].

2 Preliminaries and Problem Statement

Instances. We consider an *arity-two relational signature* σ consisting of *relations* with an associated *arity*, where the maximal arity of the signature is assumed to be 2. A σ -*instance* (or just *instance*) is a set of *facts*, i.e., expressions of the form $R(a, b)$ where a and b are constants and $R \in \sigma$. We assume without loss of generality that all relations in σ are binary, i.e., have arity two. Indeed, if there are unary relations U , we can simply code them with a binary relation U' , replacing facts $U(a)$ by $U'(a, a)$ in instances, and modifying the query to interpret $U'(a, a)$ as $U(a)$ and to ignore facts $U'(a, b)$ with $a \neq b$: this is similar to Theorem 8.4 of [2]. Accordingly, we call a fact $R(a, b)$ *unary* if $a = b$, otherwise it is *binary*.

The *domain* $\text{dom}(I)$ of an instance I is the set of constants occurring in I . A *homomorphism* from I to an instance I' is a function $h: \text{dom}(I) \rightarrow \text{dom}(I')$ such that, for each fact $R(a, b)$ of I , the fact $R(h(a), h(b))$ is in I' . We say that I' is a *subinstance* of I , written $I' \subseteq I$, if I' is a subset of the facts of I ; we then have $\text{dom}(I') \subseteq \text{dom}(I)$.

Queries. A query Q over σ is a Boolean function over σ -instances which we always assume to be *homomorphism-closed*, i.e., if Q returns true on I and I has a homomorphism to an instance I' then Q also returns true on I' . When Q returns true on I we call I a *model* of Q , or say that I *satisfies* Q (written $I \models Q$); otherwise I *violates* Q . Any homomorphism-closed query Q is *monotone*, i.e., if I satisfies Q and $I \subseteq I'$ then I' satisfies Q . A *subinstance-minimal model* of Q is a model I of Q such that no strict subinstance of I satisfies Q .

We focus on *unbounded queries*, i.e., queries having an infinite number of subinstance-minimal models. Examples of well-studied homomorphism-closed query languages include *conjunctive queries* (CQs), *unions of CQs* (UCQs), *regular path queries* (RPQs), and *Datalog* without inequalities or negations. The queries defined by Datalog or RPQs are unbounded unless they are equivalent to a UCQ (i.e., non-recursive Datalog); more generally a query is either unbounded or equivalent to a UCQ.

UR and PQE problems. In this paper, we study *uniform reliability* (UR). The problem $\text{UR}(Q)$ for a fixed query Q is the following: we are given as input an instance I , and we must return how many subinstances of I satisfy Q , i.e., the number $|\{I' \subseteq I \mid I' \models Q\}|$. Note that we have no general upper bound on the complexity of this problem, as we allow queries to be arbitrarily complex or even undecidable to evaluate, e.g., “there is a path $R(x_1), S(x_1, x_2), \dots, S(x_{n-1}, x_n), T(x_n)$ where n is the index of a Turing machine that halts”.

We will sometimes consider the generalization of UR called *probabilistic query evaluation* (PQE). The $\text{PQE}(Q)$ problem for a fixed query Q asks, given an instance I and a probability distribution $\pi: I \rightarrow [0, 1]$ mapping each fact of I to a rational in $[0, 1]$, to determine the total probability of the subinstances of I satisfying Q , when each fact $F \in I$ is drawn independently from the others with the probability $\pi(F)$. Formally, we must compute:

$$\sum_{I' \subseteq I \text{ s.t. } I' \models Q} \prod_{F \in I'} \pi(F) \times \prod_{F' \in I \setminus I'} (1 - \pi(F')).$$

The UR problem is a special case of PQE where the function π maps all facts to $1/2$, up to renormalization, i.e., multiplying by $2^{|I|}$. We will sometimes abusively talk about UR as the problem of computing that probability, because this probabilistic phrasing makes it more convenient, e.g., to reason about conditional probabilities, or about negligible probabilities.

Hard problems. The goal of this paper is to show Theorem 1.3. We will establish $\#P$ -hardness using *polynomial-time Turing reductions* [4] (see [2] for details). Specifically, we reduce from one of two $\#P$ -hard problems, depending on the query. In [2], we reduce from the problems $\#PP2DNF$ and $U\text{-}ST\text{-}CON$ (undirected source-to-target connectivity), which are shown to be $\#P$ -hard in [12]. In this paper, given our focus on UR, we reduce from variants of these problems: the λ, μ, ν -*variable-clause-variable probabilistic $\#PP2DNF$ problem* and the ϕ, η -*vertex-edge probabilistic $U\text{-}ST\text{-}CON$ problem*. We first define the first problem:

► **Definition 2.1.** Let $0 < \lambda, \nu < 1$ and $0 < \mu \leq 1$ be fixed probabilities. The λ, μ, ν -*variable-clause-variable probabilistic $\#PP2DNF$ problem* (or for brevity λ, μ, ν - $\#PP2DNF$) is the following: given a bipartite graph $(U \cup V, E)$ with $E \subseteq U \times V$, we ask for the probability that we keep an edge and its two incident vertices, where vertices of U have probability λ to be kept, edges of E have probability μ to be kept, and vertices of V have probability ν to be kept, all these choices being independent. Formally, we must compute:

$$\sum_{\substack{(U', E', V') \subseteq U \times E \times V \\ E' \cap (U' \times V') \neq \emptyset}} \lambda^{|U'|} \times (1 - \lambda)^{|U| - |U'|} \times \mu^{|E'|} \times (1 - \mu)^{|E| - |E'|} \times \nu^{|V'|} \times (1 - \nu)^{|V| - |V'|}$$

The name #PP2DNF is because of the link to positive partitioned 2-DNF formulas, which we do not need here. We can show that λ, μ, ν -#PP2DNF is #P-hard, by adapting the proof in [3] which shows the hardness of uniform reliability for the query $R(x), S(x, y), T(y)$:

► **Proposition 2.2** ([3]). *For any fixed $0 < \lambda, \nu < 1$ and $0 < \mu \leq 1$, the problem λ, μ, ν -#PP2DNF is #P-hard.*

We now define the second problem:

► **Definition 2.3.** *Let $0 < \phi \leq 1$ and $0 < \eta < 1$ be fixed probabilities. The ϕ, η -vertex-edge-probabilistic U-ST-CON problem (or for brevity ϕ, η -U-ST-CON) is the following: given an undirected graph $G = (V, E)$ and source and sink vertices $r, s \in V$ with $r \neq s$, we ask for the probability that we keep a subset of edges and vertices containing a path that connects r and s (in particular keeping r and s), where vertices have probability ϕ to be kept and edges have probability η to be kept, all these choices being independent. Formally, we must compute:*

$$\sum_{\substack{V' \subseteq V, E' \subseteq E \\ r \text{ and } s \text{ connected in } (V', E'_{|V'})}} \phi^{|V'|} \times (1 - \phi)^{|V| - |V'|} \times \eta^{|E'|} \times (1 - \eta)^{|E| - |E'|}$$

This intuitively combines features of the undirected source-to-target edge-connectedness and node-connectedness problems of [15]. With standard techniques and some effort, we can show that ϕ, η -U-ST-CON is #P-hard (see the full version [1]):

► **Proposition 2.4.** *For any fixed $0 < \phi \leq 1$ and $0 < \eta < 1$, the problem ϕ, η -U-ST-CON is #P-hard.*

3 Basic Techniques: Dissociation, Tight Edges

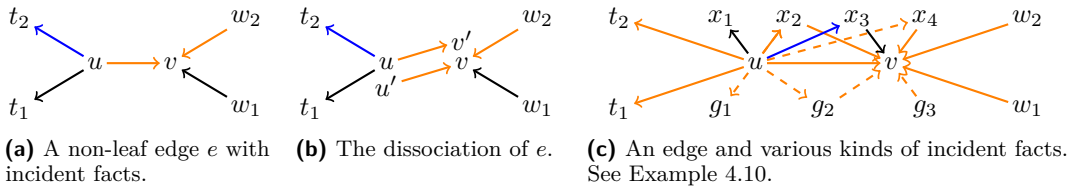
Having presented the hard problems, we now recall the notion of *edges* and how we *copy* them, and the *dissociation* operation introduced in [2]. We also present *tight edges* and re-state the result of [2] showing that unbounded queries have models with tight edges.

Edges and copies. An *edge* e in an instance I is an ordered pair (u, v) of distinct elements of $\text{dom}(I)$ such that there is at least one fact of I using both u and v , i.e., of the form $R(u, v)$ or $R(v, u)$, hence non-unary. The *covering facts* of e in I is the non-empty set of these facts. Note that (u, v) is an edge iff (v, u) is, and they have the same covering facts.

We call $e = (u, v)$ a *non-leaf edge* if I contains facts using u but not v (called *left-incident facts*) and facts using v but not u (called *right-incident facts*). An example is shown in Figure 1a (with no unary facts). The left-incident and right-incident facts are called together the *incident facts*; note that they may include unary facts.

In this paper we will often modify instances I by *copying* an edge $e = (u, v)$ of I to some other ordered pair (u', v') of elements. This means that we modify I to add, for each covering fact F of e , the fact obtained by replacing u by u' and v by v' . Note that, if u' and v' are both fresh, or if $u' = u$ and v' is fresh or $v' = v$ and u' is fresh, then the result of this process has a homomorphism back to I . Clearly, copying (u, v) on (u', v') is equivalent to copying (v, u) on (v', u') (but different from copying, say, (u, v) on (v', u')). Note that copying an edge does *not* copy its incident facts, though our constructions will often separately copy some of them.

► **Example 3.1.** In the instance $I = \{R(a), S(a, b), S'(b, a), T(b)\}$, copying (a, b) on (a, b') for a fresh element b' means adding the facts $S(a, b'), S'(b', a)$.



■ **Figure 1** Examples of Section 3 and 4.

Dissociation. One basic operation on instances is *dissociation*, which replaces one edge by two copies connected to each endpoint:

► **Definition 3.2.** Let I be an instance and $e = (u, v)$ be a non-leaf edge of I . The dissociation of e in I is obtained by modifying I to add two fresh elements u' and v' , copying e to (u', v) and to (u, v') , and then removing the covering facts of e .

The process is illustrated in Figures 1a and 1b. Note the following immediate observation:

▷ **Claim 3.3.** The dissociation of an edge in I has a homomorphism back to I .

Tight edges. We can then define a *tight edge* as one whose dissociation breaks the query:

► **Definition 3.4.** A non-leaf edge (u, v) in an instance I is tight for the query Q if I satisfies Q but the dissociation of (u, v) in I does not.

We use a result of [2] which shows that unbounded queries must have a model with a tight edge. This is the only point where we use the unboundedness of the query.

► **Theorem 3.5** (Theorem 6.6 in [2]). Any unbounded query has a model with a tight edge.

We give a proof sketch for completeness (see [2] for the proof):

Proof sketch. As the query Q is unbounded, it has infinitely many minimal models: let I be a sufficiently large one. Iteratively dissociate the non-leaf edges of I until none remain (this always terminates), and let I' be the result. If I' violates Q , then some dissociation broke Q , i.e., was applied to a tight edge in a model of Q . Otherwise, I' has no non-leaf edges and satisfies Q . We can then show thanks to the simple structure of I' that it has a constant-sized subset that satisfies Q , and deduce that Q already holds on a constant-sized subinstance of I . As I is large, this contradicts the minimality of I . ◀

Thus, in the sequel, we fix the query Q and assume that it has a model with a tight edge. Note that some bounded queries may also have a tight edge, e.g., the prototypical unsafe CQ $R(x), S(x, y), T(y)$; our results in this paper thus also apply to some bounded queries.

4 Minimality and Critical Models

In this section, we refine the notion of a tight edge to impose minimality criteria and get to the notion of *critical models*. We define three successive minimality criteria, which we present intuitively here before formalizing them in the rest of this section. The first is called *weight* and counts the covering facts; the *critical weight* Θ is the minimal weight of a tight edge. Having defined Θ , we restrict our attention to *clean* tight edges e , whose incident facts do not include so-called *garbage facts*, i.e., strict subsets of the covering facts of e . The second criterion is *extra weight* and counts the incident facts that are not isomorphic to

the covering facts; the *critical extra weight* Ξ is the minimal extra weight of a tight edge of weight Θ . The third criterion is *lexicographic weight* and counts the other left-incident and right-incident facts, ordered lexicographically: the *critical lexicographic weight* Λ is the minimal lexicographic weight of a tight edge of weight Θ and extra weight Ξ .

We then define a *critical model* as a *subinstance-minimal* model with a clean tight edge that optimizes these three weights in order, and show that such models exist.

Weight. The *weight* was defined in [2], but unlike in [2] we do not count unary facts:

► **Definition 4.1.** *The weight of an edge $e = (u, v)$ in an instance I is the number of covering facts of e (it is necessarily greater than 0).*

► **Example 4.2.** The weight of (a, b) in $I = \{R(b), T(b, c), S(b, a), S'(b, a), U(a, b)\}$ is 3.

The minimal weight of a tight edge across all models is an intrinsic characteristic of Q , called the *critical weight*:

► **Definition 4.3.** *The critical weight of the query Q , written $\Theta \geq 1$, is the minimum, across all models I of Q and tight edges e of I , of the weight of e in I .*

The point of the critical weight is that edges with weight less than Θ can never be tight:

▷ **Claim 4.4.** Let I be a model of Q and $e = (u, v)$ be a non-leaf edge of I . If the weight of e is less than Θ , then the dissociation of e in I is also a model of Q .

► **Example 4.5.** The bounded CQ $Q' : R(x), S(x, y), S'(x, y), T(y)$ has critical weight 2, as witnessed by the model $I' = \{R(a), S(a, b), S'(a, b), T(b)\}$ with a tight non-leaf edge (a, b) of weight 2 and the inexistence of a model with a tight non-leaf edge of weight 1.

As Q' has critical weight 2, in any model I of Q' , if we have an edge $e = (u, v)$ with only one covering fact using both u and v , we know that dissociating e cannot make Q' false.

Having defined Θ , to simplify further definitions, we introduce the notion of a *clean* edge as one that does not have incident facts achieving strict subsets of its covering facts:

► **Definition 4.6.** *Let I be an instance, let $e = (u, v)$ be an edge of I , and let $C \subseteq I$ be the covering facts of e . For any edge (u, t) , if its covering facts are isomorphic to a strict subset of C when renaming t to v , then we call these left-incident facts left garbage facts. Likewise, the right garbage facts are the right-incident facts that are covering facts of edges (w, v) that are isomorphic to a strict subset of C when renaming w to u .*

We call e *clean* if it has no left or right garbage facts (called collectively garbage facts).

► **Example 4.7.** In the instance $I = \{S(a, b'), U(a), S(a, b), S'(b, a), T(c, b), S(c, b), S'(d, b), S'(b, e), S(f, b)\}$, the left garbage facts of the edge (a, b) are $\{S(a, b')\}$ on the edge (a, b') , and the right garbage facts are $\{S'(b, e)\}$ on the edge (e, b) and $\{S(f, b)\}$ on the edge (f, b) . Note that there are no garbage facts on the edge (b, c) , because the covering facts $\{T(c, b), S(c, b)\}$ of this edge are not isomorphic to a strict subset of the covering facts of (a, b) . Further note that there are no garbage facts on the edge (d, b) , because the covering facts $\{S'(d, b)\}$ are not isomorphic to a strict subset of the covering facts of (a, b) when renaming d to a .

We will always be able to ensure that tight edges with critical weight are clean, justifying that we restrict our attention to clean tight edges in the sequel:

▷ **Claim 4.8.** If Q has a model with a tight edge, then it has a model with a clean tight edge of weight Θ .

Proof sketch. We find a model with a tight edge of weight Θ by definition of Θ . Then, any edges with garbage facts have weight $< \Theta$, so they can be dissociated using Claim 4.4 and homomorphically merged to e . At the end of this process, e is clean and is still tight. \triangleleft

Extra weight. We further restrict tight edges e by limiting their number of incident facts, similarly to the notion of *side weight* in [2]. However, in this paper, we additionally partition the incident facts between so-called *extra facts* and *copy facts*. Intuitively, our reductions will use codings that introduce copies of the edge e , and the *extra facts* are those that can be “distinguished” from incident copies of e added in codings; by contrast copy facts are non-unary facts in edges that are isomorphic copies of e and therefore “indistinguishable”.

We want to minimize the number of extra facts, to intuitively ensure that they are all “necessary”, in the sense that a copy of e missing an incident extra fact can be dissociated. Let us formally define the extra facts: among the non-garbage incident facts, they are those that are part of a so-called *triangle* (i.e., involve an element occurring both in a left-incident in a right-incident fact), those which are unary, or those which are a covering fact of an edge whose covering facts are not isomorphic to the covering facts of e .

► **Definition 4.9.** Let I be an instance with an edge $e = (u, v)$, and let $C \subseteq I$ be the covering facts of e . An element $w \in \text{dom}(I)$ forms a triangle with e if both (u, w) and (v, w) are edges.

Let (u', v') be some edge of I . We call (u', v') a copy of (u, v) if the covering facts of (u', v') are isomorphic to C by the isomorphism mapping u' to u and v' to v .

We partition the non-garbage left-incident facts of (u, v) between:

- The left copy facts, i.e., the binary facts involving u and an element v' such that (u, v') is a copy of (u, v) and v' does not form a triangle with e : we call v' a left copy element of e .
- The left extra facts, which comprise all other non-garbage left-incident facts, namely:
 - The unary facts on u .
 - The non-garbage binary facts involving u and some element x such that:
 - * the element x forms a triangle with e ; or
 - * the covering facts of the edge (u, x) are not isomorphic to C .

We partition the non-garbage right-incident facts into right extra facts and right copy facts with right copy elements in a similar way. Note that, as we prohibit triangles, the left copy elements and right copy elements are disjoint. We talk of the copy elements, copy facts, extra facts of e to denote both the left and right kinds.

► **Example 4.10.** Consider the instance of Figure 1c and the edge $e = (u, v)$. The covering facts C of e are represented as an orange edge, and the other orange edges represent edges which are copies of e . The left and right copy elements are respectively t_1 and t_2 and w_1 and w_2 . The dashed orange edges represent edges whose covering facts are a strict subset of C , i.e., they are garbage facts. The extra facts include unary facts (not pictured), facts with x_1 (the black edge (u, x_1) represents non-garbage facts not isomorphic to C), and facts with x_2 , x_3 , and x_4 (which form triangles).

Note that garbage facts are neither extra facts nor copy facts, and are ignored in the definition above except in that they may help form triangles. This does not matter: thanks to Claim 4.8, garbage facts will only appear in intermediate steps of some proofs. We can now define the *critical extra weight* as the minimal extra weight of a tight edge with weight Θ :

► **Definition 4.11.** The critical extra weight of Q , written $\Xi \geq 0$, is the minimum across all models I of Q and tight edges e of I of weight Θ , of the number of extra facts of e in I .

► **Example 4.12.** Continuing Example 4.5, the query Q' had critical extra weight 2, as witnessed by I' . The query $Q'' : R(x), S(x, y), S(x', y), S(x', y'), T(y')$, has critical weight 1 and critical extra weight 0, as witnessed by the model $I'' = \{R(a), S(a, b), S(a', b), S(a', b'), T(b')\}$ where the edge (a', b) is tight and has weight 1 and extra weight 0.

Again, the definition of critical extra weight clearly ensures:

▷ **Claim 4.13.** Let I be a model of Q and $e = (u, v)$ be a non-leaf edge. If e has weight Θ and extra weight $< \Xi$, then the dissociation of e in I is also a model of Q .

Lexicographic weight. We then impose a third minimality requirement on tight edges e , which is needed in Section 6 (but unused in Section 5). The intuition is that we want to limit the number of copy elements. Specifically, we minimize first the number τ of left copy elements, then the number ω of right copy elements, hence the name *lexicographic weight*. This is why, when choosing a tight edge, we also choose an orientation (i.e., choosing (u, v) as a tight edge is different from choosing (v, u)):

► **Definition 4.14.** Let I be an instance with an edge $e = (u, v)$. Let τ be the number of left copy elements and ω be the number of right copy elements of e . The lexicographic weight of e is the ordered pair (τ, ω) . We order these ordered pairs lexicographically, i.e., $(\tau, \omega) < (\tau', \omega')$ with $\tau, \tau', \omega, \omega' \in \mathbb{N}$ iff $\tau < \tau'$ or $\tau = \tau'$ and $\omega < \omega'$.

The critical lexicographic weight Λ of Q is the minimum, over all models I of Q and all tight edges of e with weight Θ and extra weight Ξ , of the lexicographic weight of e .

Note that minimizing the lexicographic weight does not always minimize the total number of copy facts¹, e.g., $(1, 3) < (2, 1)$ but $1 + 3 > 2 + 1$. However, it is always the case that removing a copy fact of an edge e causes the lexicographic weight of e to decrease (and does not cause the extra weight to increase, as the remaining covering facts of the edge are garbage facts).

Again, we have:

▷ **Claim 4.15.** Let I be a model of Q and $e = (u, v)$ be a non-leaf edge with weight Θ , extra weight Ξ , and lexicographic weight $< \Lambda$. Then, the dissociation of e in I is also a model of Q .

Critical models. We now define *critical models* (significantly refining the so-called *minimal tight patterns* of [2]). A critical model I is intuitively a model of Q with a clean tight edge e that achieves the minimum of our three weight criteria, and where we additionally impose that I is subinstance-minimal. For convenience we also specify a choice of incident facts in the critical model, but this choice is arbitrary, i.e., we can pick any pair of a left-incident fact and right-incident fact.

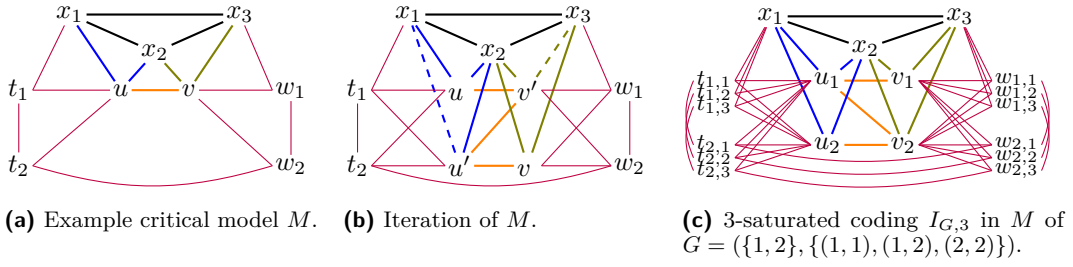
► **Definition 4.16.** A critical model (I, e, F_L, F_R) is a model I of Q which is subinstance-minimal, a clean tight edge e of I having weight Θ , extra weight Ξ , and lexicographic weight Λ , and a left-incident fact $F_L \in I$ and a right-incident fact $F_R \in I$ of e .

We can now claim that critical models exist:

► **Proposition 4.17.** If a query Q has a model with a tight edge, then it has a critical model.

¹ Minimizing the total number of copy facts, or minimizing along the componentwise partial order on $\mathbb{N} \times \mathbb{N}$, would suffice almost everywhere in the proof except in part of Section 6.

14:10 Uniform Reliability for Unbounded Homomorphism-Closed Graph Queries



■ **Figure 2** Examples of Section 5 and illustration of the notation.

Proof sketch. The existence of models with tight edges achieving the critical weights is by definition, cleanliness can be imposed by the process used to prove Claim 4.8, and subinstance-minimality can easily be imposed by picking some minimal subset of facts of the model that satisfy the query. ◀

5 Hardness with a Non-Iterable Critical Model

Having defined critical models, we now start our hardness proof. As in [2], we will distinguish two cases, based on whether we can break Q with an *iteration* process on a critical model.

► **Definition 5.1.** Let $M = (I, e, F_L, F_R)$ be a critical model, let $e = (u, v)$, and let C be the covering facts of e . Let A and B be the set of the left-incident and right-incident facts of e in I , respectively. The iteration of M is obtained by modifying I in the following way:

- Add fresh elements u' and v' , copy e on (u, v') , (u', v') , (u', v) , and remove the facts of C .
- Create a copy of the facts of $A \setminus \{F_L\}$ where we replace u by u' .
- Create a copy of the facts of $B \setminus \{F_R\}$ where we replace v by v' .

► **Example 5.2.** Consider the critical model in Figure 2a, with edge (u, v) and where F_L and F_R are binary facts respectively using u and x_1 and v and x_3 . Its iteration is shown in Figure 2b, with dashed edges representing edges where F_L and F_R are missing.

A *non-iterable* critical model M is one whose iteration no longer satisfies the query; otherwise M is *iterable*. In this section, we show hardness when there is a non-iterable critical model:

► **Proposition 5.3.** Assume that Q has a non-iterable critical model. Then the uniform reliability problem for Q is $\#P$ -hard.

We prove this result in the rest of this section.

Fixing notation. Fix the critical model $M = (I, e, F_L, F_R)$ and let $e = (u, v)$ be the tight clean edge. We must introduce some notation to talk about the incident facts of e in I , which is summarized in Figure 2a. As e is clean, we know that its incident facts are either extra facts or copy facts – there are no garbage facts.

Let $C \subseteq I$ be the covering facts of e in I (in orange on the picture), with $|C| = \Theta$. Let $X = \{x_1, \dots, x_k\}$ be the elements different from u and v with which one of u or v has a (non-unary) extra fact or has one of the two facts F_L and F_R . Note that some of the elements in X may have facts with both u and v (i.e., triangles), like x_2 in the picture. We may have $k = 0$, specifically when F_L and F_R are unary facts and any other extra facts are unary.

Further let $T = \{t_1, \dots, t_\tau\}$ be the left copy elements of e not in X , and let $W = \{w_1, \dots, w_\omega\}$ be the right copy elements of e not in X , with T and W disjoint (because copy elements cannot form triangles). We exclude elements of X because, if F_L (resp., F_R) is a copy fact, then X contains exactly one left copy element (resp., exactly one right copy element)². Also note that we may have $\tau = \omega = 0$, i.e., if there are no copy facts except possibly those of the edges of F_L and of F_R .

To recapitulate, the incident facts of e in I only involve elements from $X \sqcup T \sqcup W$. Specifically, they are the unary facts on u , the unary facts on v , the non-unary extra facts (which involve one of $\{u, v\}$ and one element of X), the facts F_L and F_R which respectively involve u and v and (if they are non unary) one element of X , and the other left and right copy facts forming isomorphic copies of e as edges (u, t_j) with $1 \leq j \leq \tau$ and (w_i, v) with $1 \leq i \leq \omega$. Notice again how, if F_L or F_R are copy facts, then these notations handle them as extra facts along with any other covering facts of their edge. Note that our description of the incident facts of e does not describe the facts that may exist between elements of $X \sqcup T \sqcup W$, and indeed these may be arbitrary (some are pictured in Figure 2a).

Coding bipartite graphs. We will reduce from our variant of #PP2DNF (Definition 2.1) by using M to code a bipartite graph $G = (U \sqcup V, E)$. Intuitively, we will create one copy u_i of u for each vertex i of U , one copy v_j of v for each vertex j of V , and copy the edge e on (u_i, v_j) for each edge (i, j) of E . The reason why we distinguish X and T and W is because we will handle them differently. For the incident facts of e that are unary or involve elements of X , we will create one single copy of them for each u_i and each v_j . Indeed, we will show that edges (u_i, v_j) that are missing one such incident fact can be dissociated (if an extra fact is missing, using Claim 4.13) or mapped in a specific way in the iteration (if one of F_L or F_R is a copy fact and we are missing one of the covering facts of their edge). For the (copy) facts involving $T \sqcup W$, we will copy them (using the fact that they are binary) by creating a large number q of copies of $T \sqcup W$. This *saturation* process will in fact create a large number of copies of all facts involving some element of $T \sqcup W$, which we call the *saturated facts*.

Let us accordingly define the *saturated coding* of a bipartite graph in M :

► **Definition 5.4.** *Let $G = (U \sqcup V, E)$ be a non-empty bipartite graph, and assume without loss of generality that $U = \{1, \dots, n\}$ and $V = \{1, \dots, m\}$.*

Let $q > 0$ be some integer. The q -saturated coding of G in M , written $I_{G,q}$, is the instance defined by modifying I in the following way:

- *For all $1 \leq p \leq q$, create fresh elements $T_p = \{t_{1,p}, \dots, t_{\tau,p}\}$ and $W_p = \{w_{1,p}, \dots, w_{\omega,p}\}$. Identify $t_j = t_{j,1}$ for $1 \leq j \leq \tau$ and $w_i = w_{i,1}$ for $1 \leq i \leq \omega$.*
- *Letting Φ be the set of the saturated facts, for each $1 \leq p \leq q$, create a copy of Φ where each element t_j is replaced by $t_{j,p}$ and each element w_i is replaced by $w_{i,p}$.*
- *Create elements u_1, \dots, u_n and v_1, \dots, v_m , where we identify $u = u_1$ and $v = v_1$.*
- *Create a copy of all incident facts of e for all u_i and v_j . Formally, let A and B be the set of the left-incident and right-incident facts of e in the current model (i.e., involving the $t_{j,p}$ and $w_{i,p}$): note that A (resp., B) contains in particular F_L (resp., F_R) and any unary facts on u (resp., on v). For each $1 \leq i \leq n$, create a copy of the facts of A replacing u by u_i , and for each $1 \leq j \leq m$ create a copy of the facts of B replacing v by v_j .*
- *Copy e (i.e., C) on (u_i, v_j) for each $(i, j) \in E$, and remove the facts of C if $(u_1, v_1) \notin E$.*

² Because of this, in general (τ, ω) may be less than the critical lexicographic weight Λ .

14:12 Uniform Reliability for Unbounded Homomorphism-Closed Graph Queries

The saturated coding process is illustrated in Figure 2c. Note that the process is in polynomial time if the value q is polynomial in the size $|G|$ of the input bipartite graph.

Understanding the coding. Letting $G = (U \sqcup V, E)$ be a non-empty bipartite graph and writing $U = \{1, \dots, n\}$ and $V = \{1, \dots, m\}$, we study the coding $I_{G,q}$ to relate subsets of $I_{G,q}$ to subsets of $U \times E \times V$. For this, we partition the facts of $I_{G,q}$ in five kinds (see Figure 2c):

- The *base facts* (pictured in black), which are the facts that do not involve any of the elements $u_1, \dots, u_n, v_1, \dots, v_m$ or any element of $\bigsqcup_{1 \leq p \leq q} T_p \sqcup W_p$ (but they may involve elements of X). These facts are precisely the facts of I that do not involve the elements u or v or any element of $T \sqcup W$, and they are unchanged in the coding.
- The *saturated facts* (in purple), i.e., the facts involving some element of $T_p \sqcup W_p$ for some $1 \leq p \leq q$. These facts exist in q copies, and some (corresponding to facts of I between u or v and an element of $T \sqcup W$) have been further copied n times (if they involve u) or m times (if they involve v).
- The *non-saturated left-incident facts* (in blue) of each vertex $i \in U$, which are the facts which involve u_i and do not involve the $T_p \sqcup W_p$, i.e., are unary or involve an element of X . These facts include in particular one copy of F_L .
- The *non-saturated right-incident facts* (in green) of each vertex $j \in V$, that involve v_j and not the $T_p \sqcup W_p$, i.e., are unary or involve an element of X ; they include one copy of F_R .
- The *copy of e* (in orange) for each edge $(i, j) \in E$, which is on the edge (u_i, v_j) of $I_{G,q}$.

The last three kinds are what we are interested in for the reduction, but the first two kinds need to be dealt with. We will show that the base facts must all be present to satisfy the query, and that each edge has some copy of the saturated facts with high probability.

Base facts. We say that a subinstance of $I_{G,q}$ is *well-formed* if all base facts are present, and *ill-formed* if at least one is missing. The following is easy to see by subinstance-minimality of I :

► **Proposition 5.5.** *The ill-formed subinstances do not satisfy the query.*

Hence, the number of subinstances of $G_{I,q}$ satisfying the query is the number of well-formed subinstances that do. Thus, in the sequel, we only consider well-formed subinstances.

Saturated facts. For the saturated facts, we will intuitively define *valid* subinstances where, for each ordered pair of vertices $(i, j) \in U \times V$, considering the copies u_i and v_j of u and v , there is a complete copy of the saturated facts that are “relevant” to them. More precisely, looking back at the original instance I , and considering the facts of I involving an element of $T \sqcup W$, there are of two types. The first type are the facts that do not involve u or v , i.e., they only involve elements of $T \sqcup W$ and possibly of $\text{dom}(I) \setminus \{u, v\}$. Each such fact has been copied q times in $I_{G,q}$, and the copy numbered $1 \leq p \leq q$ uses one or two elements of $T_p \sqcup W_p$. The second type are the facts involving u or v in I (they cannot involve both). These facts have been copied $n \times q$ or $m \times q$ times in $I_{G,q}$, each copy using one element of $T_p \sqcup W_p$ for some $1 \leq p \leq q$ and one u_i for some $1 \leq i \leq n$ or one v_j for some $1 \leq j \leq m$. What we require of a valid subinstance $J \subseteq I_{G,q}$ is that, for each pair of vertices $(i, j) \in U \times V$, we have in J some copy $1 \leq p \leq q$ containing all facts of the first type and all facts of the second type involving u_i and v_j :

► **Definition 5.6.** We partition the saturated facts of $I_{G,q}$ in q copies: formally, the p -th saturated copy for $1 \leq p \leq q$ is the subset of the saturated facts of $I_{G,q}$ that involve some element of $T_p \sqcup W_p$. A saturation index for $I_{G,q}$ is a function $\iota: U \times V \rightarrow \{1, \dots, q\}$.

For $J \subseteq I_{G,q}$, we say that J is valid for ι if, for each $(i, j) \in U \times V$, letting $p := \iota(i, j)$, considering the facts of the p -th saturated copy, J contains all such facts that are:

- of the first type, i.e., J contains all facts of $I_{G,q}$ that involve some element of $T_p \sqcup W_p$ and do not involve any elements of $\{u_{i'} \mid 1 \leq i' \leq n\} \sqcup \{v_{j'} \mid 1 \leq j' \leq m\}$;
- of the second type and involve u_i or v_j , i.e., J contains all facts of $I_{G,q}$ that involve some element of $T_p \sqcup W_p$ and involve either u_i or v_j .

We call J valid if there is a saturation index for which it is valid; otherwise J is invalid.

Note that, for each choice of ordered pair $(i, j) \in U \times V$, the required facts can be found in a different saturated copy $\iota(i, j)$, i.e., we do not require that there is a p such that J contains all facts of the p -th saturated copy. Indeed this stronger requirement would be too hard to ensure: intuitively, the number of facts required for each (i, j) is constant (it only depends on I), but the number of facts in the p -th saturated copy depends on G (it is linear in $|U| \times |V|$).

We now show that we can pick a number q of copies which is polynomial in the input G , but makes it very unlikely that a random subinstance is invalid. Thanks to this, we do not need to know which ones of the invalid subinstances satisfy Q . Indeed, the proportion of subinstances of $I_{G,q}$ that satisfy Q will be the proportion of valid subinstances that do, up to an error which is much less than the probability of any valid subinstance and can be eliminated by rounding:

► **Lemma 5.7.** There is a polynomial P_M depending on the critical model M such that, for any non-empty bipartite graph $G = (U \sqcup V, E)$, letting $\chi := |U| + |V| + |E|$ be the size of G and defining $q := P_M(\chi)$, the proportion of subinstances of $I_{G,q}$ that are invalid is strictly less than $2^{-(\chi|I|+1)}$.

Thanks to this, we focus on the well-formed subinstances J where we keep some subset of the saturated facts making J valid. We now fix q to the value of Lemma 5.7, and build $I_{G,q}$ in polynomial time in the input bipartite graph G (with the critical model M being fixed).

Good and bad subinstances. Let us now study the status of the last three kinds of facts:

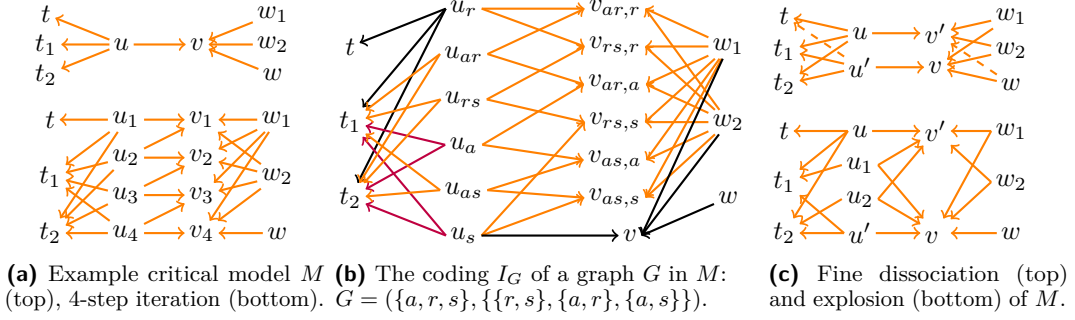
► **Definition 5.8.** Let $J \subseteq I_{G,q}$. For $1 \leq i \leq n$ (resp., $1 \leq j \leq m$), the vertex $i \in U$ (resp., $j \in V$) is complete in J if all its non-saturated left-incident facts (resp., non-saturated right-incident facts) are present in J , and incomplete otherwise. The edge $(i, j) \in E$ is complete in J if all covering facts of (u_i, v_j) in $I_{G,q}$ are present in J , and incomplete otherwise. We call J good if there is an edge $(i, j) \in E$ with (i, j) , i , and j complete, and bad otherwise.

We now claim that, among the well-formed valid subinstances, the good ones satisfy the query, and the bad ones do not. This is easy to see for good subinstances:

► **Proposition 5.9.** For any good valid well-formed subinstance $J \subseteq I_{G,q}$, there is a homomorphism from I to J .

Proof sketch. As J is well-formed all base facts are present, and J is valid for some saturation index ι . Let $(i, j) \in E$ be an edge witnessing that J is good. The homomorphism maps $T \sqcup W$ to $T_{\iota(i,j)} \sqcup W_{\iota(i,j)}$, maps u to u_i and v to v_j , and is the identity otherwise. ◀

For bad subinstances, we show with much more effort that they do not satisfy the query:



■ **Figure 3** Examples of Section 6 and illustration of the notation.

► **Proposition 5.10.** *Any bad subinstance $J \subseteq I_{G,q}$ does not satisfy the query.*

Proof sketch. It suffices to study the case with no saturation, i.e., $q = 1$. We dissociate incomplete edges with Claim 4.4, and dissociate complete edges missing at least one incident extra fact with Claim 4.13, which does not break Q . Then we show how to map this homomorphically to the iteration I' of M , by mapping complete vertices to u and v in the dissociation, and mapping the vertices which are missing facts of the edges of F_L or F_R to u' and v' respectively (after dissociating these edges if F_L or F_R are copy facts). This contradicts the assumption that M was non-iterable, i.e., that I' violates Q . ◀

This establishes that the status of Q on a valid well-formed subinstance J depends on whether J is good or bad, i.e., depends on which of the last three kinds of facts were kept in J . Now, the subsets of these facts are clearly in correspondence with the subsets of $U \times E \times V$ for the λ, μ, ν -#PP2DNF problem (see Definition 2.1), for some choice of constant probabilities λ, μ, ν . Further, a subset of $U \times E \times V$ is counted in λ, μ, ν -#PP2DNF if and only if the corresponding subset of the last three kinds of facts yields a good subinstance. As the ill-formed subinstances are easy to count, and the invalid ones are negligible, we can conclude the reduction and establish Proposition 5.3. The complete proof is given in [1].

6 Hardness when all Critical Models are Iterable

In this last section, we show hardness in the case where all critical models are iterable:

► **Proposition 6.1.** *Assume that Q has a critical model and that all critical models of Q are iterable. Then the uniform reliability problem for Q is #P-hard.*

A first observation is that, in this case, we have $\Xi = 0$, by contraposition of the following:

▷ **Claim 6.2.** *If the critical extra weight is > 0 , then Q has a non-iterable critical model.*

Proof sketch. Take a critical model $M = (I, e, F_L, F_R)$ with $e = (u, v)$ and one of F_L, F_R an extra fact. The edge (u', v') in the iteration of M has weight Θ and extra weight $< \Xi$, so we can dissociate it without breaking Q and merge the two resulting copies. This yields the so-called *fine dissociation* (illustrated in Figure 3c, and formally defined in the full version of this paper [1]), which violates Q . ◀

Hence, in the rest of the section, we assume $\Xi = 0$, and fix an iterable critical model $M = (I, e, F_L, F_R)$. All incident facts of $e = (u, v)$ in I are copy facts, so we let $t, t_1, \dots, t_{\tau-1}$ be the left copy elements and $w, w_1, \dots, w_{\omega-1}$ be the right copy elements, where t and w are

the elements that occur in F_L and F_R respectively (the choice of F_L and F_R from now on only matters in that it distinguishes two copy elements t and w). The lexicographic weight of e in I is thus $\Lambda = (\tau, \omega)$ with $\tau, \omega \geq 1$. We let C be the covering facts of e in I . See Figure 3a.

n -step iteration. Let us now define the n -step iteration of M . It is related to iteration in [2], but specialized to the case where $\Xi = 0$, i.e., all incident facts are copy facts.

► **Definition 6.3.** For $n > 0$, the n -step iteration of M is obtained by modifying I :

- Create elements u_1, \dots, u_n and v_1, \dots, v_n , where we identify u and u_1 and v_n and v .
- For all $1 \leq i, j \leq n$, copy e on $(u_i, t_{j'})$ and $(w_{i'}, v_j)$ for all $1 \leq j' < \tau$ and $1 \leq i' < \omega$.
- For all $1 \leq i \leq n$, copy e on (u_i, v_i) for all $1 \leq i \leq n$ and on (u_{i+1}, v_i) for all $1 \leq i < n$.
- Remove the facts of C , except in the trivial case where $n = 1$.

The iteration is illustrated in Figure 3a. Note that the 1-step iteration is exactly I . Further, the 2-step iteration resembles the iteration in Section 5, but omits some incomplete copies of (u, t) and (w, v) (i.e., the dashed edges in Figure 2b): as t and w are copy elements these facts would be garbage facts so the difference is inessential.

We now show that, if the iteration process of Section 5 cannot break Q on any critical model, then Q must also be satisfied in the n -step iteration of any critical model M for any $n > 0$. This proposition summarizes how we use the hypothesis that all critical models are iterable:

► **Proposition 6.4.** Let Q be a query that has a critical model. Assume that all critical models for Q are iterable. Then $\Xi = 0$ and, for any critical model M of Q , for any $n > 0$, the n -iteration of M satisfies Q ; further it is a subinstance-minimal model of Q .

Proof sketch. Intuitively, the n -step iteration can be achieved by repeatedly performing the iteration from Section 5. A tedious point in the proof is to show that subinstance-minimality is preserved throughout this process. ◀

Coding. We explain how to code an undirected graph to reduce from ϕ, η -U-ST-CON for some $0 < \phi \leq 1$ and $0 < \eta < 1$ (see Definition 2.3): this time no saturation is needed. Proposition 6.4 will then intuitively show that some paths in the coding make Q true.

► **Definition 6.5.** Let $G = (V, E)$ be an undirected graph with source r and sink s , with $r \neq s$. The coding I_G of G in M is the instance defined by modifying I in the following way:

- For all $a \in V$, create a fresh element u_a , and copy $(u, t_{j'})$ on $(u_a, t_{j'})$ for all $1 \leq j' < \tau$.
- We identify u to u_r , so u_r also occurs in another copy of e , namely the edge (u_r, t) .
- For each edge $\pi = \{a, b\} \in E$, create fresh elements $u_\pi, v_{\pi,a}, v_{\pi,b}$, copy $(u, t_{j'})$ on $(u_\pi, t_{j'})$ for all $1 \leq j' < \tau$, copy $(w_{i'}, v)$ on $(w_{i'}, v_{\pi,\beta})$ for all $1 \leq i' < \omega$ and $\beta \in \{a, b\}$, and copy (u, v) on $(u_a, v_{\pi,a})$, $(u_\pi, v_{\pi,a})$, $(u_\pi, v_{\pi,b})$, and $(u_b, v_{\pi,b})$.
- Copy (u, v) on (u_s, v) , and then remove the facts of C .

An example is given in Figure 3b, shortening the vertex names for readability. The coding I_G can clearly be built in polynomial time in G . We partition the facts of I_G in four kinds:

- The *base facts* (not pictured), i.e., the facts involving no element of $\{u_a \mid a \in V\} \cup \{v_{\pi,\beta} \mid \pi \in E, \beta \in \pi\} \cup \{v\}$.
- The *supplementary base facts* (in black), i.e., the covering facts of (u_r, t) and $(u_r, t_{j'})$ for $1 \leq j' < \tau$, and the covering facts of (u_s, v) and (w, v) and $(w_{i'}, v)$ for $1 \leq i' < \omega$.

14:16 Uniform Reliability for Unbounded Homomorphism-Closed Graph Queries

- The *vertex facts* (in purple) of each vertex $a \in V \setminus \{r\}$, i.e., the covering facts of $(u_a, t_{j'})$ for $1 \leq j' < \tau$.
- The *edge facts* (in orange) of each edge $\pi = \{a, b\}$ of E , i.e., all covering facts and incident facts of $(u_\pi, v_{\pi,a})$ and $(u_\pi, v_{\pi,b})$, including the covering facts of $(u_a, v_{\pi,a})$ and $(u_b, v_{\pi,b})$.

Similarly to Section 5, the base facts of I_G are precisely the facts of I that do not involve u or v . A subinstance $J \subseteq I_G$ is *well-formed* if it contains all base facts and supplementary base facts, and *ill-formed* otherwise. We can then use subinstance-minimality to show:

▷ **Claim 6.6.** The ill-formed subinstances do not satisfy the query.

Now, consider a well-formed subinstance $J \subseteq I_G$. A vertex $a \in V$ is *complete* in J if all vertex facts of a are present, and *incomplete* otherwise; and an edge $\pi \in E$ is *complete* in J if all its edge facts of π are present, and *incomplete* otherwise. A *complete path* in J is a path connecting r and s in G such that all traversed edges and vertices are complete in J (except r , for which completeness was not defined). We say that J is *good* if it has a complete path, and *bad* otherwise. We can easily see that good subinstances satisfy the query, because they contain an iterate of M and we can use Proposition 6.4:

▷ **Claim 6.7.** For any good well-formed subinstance $J \subseteq I_G$, there is a homomorphism from the $(2n + 1)$ -step iteration of M to J , where n is the length of a complete path in J .

It is again far more challenging to show the other claim:

▷ **Claim 6.8.** Any bad subinstance $J \subseteq I_G$ does not satisfy the query.

Proof sketch. We dissociate all copies of e that are missing a fact or are of the form $(u_\beta, v_{\pi,\beta})$ and are missing an incident fact with some element $w_{i'}$. Then, we map the result by a homomorphism h to a structure called the *explosion* (illustrated in Figure 3c and formally defined in the full version of this paper [1]), which intuitively reflects all maximal strict subsets of the $\{t_1, \dots, t_{\tau-1}\}$, and violates Q (by considering the lexicographic weight of its edges). We define h along the cut of G defined by considering the vertices reachable from r via a complete path. ◁

We then show hardness by reducing from ϕ, η -U-ST-CON for well-chosen constant probabilities ϕ and η (up to assuming that the source vertex r is always kept) and thus conclude the reduction, establishing Proposition 6.1. Together with Proposition 5.3, as Q has a critical model by Proposition 4.17 and Theorem 3.5, we have shown our main result (Theorem 1.3).

7 Conclusion

We have proved the intractability of uniform reliability for unbounded homomorphism-closed queries on arity-two signatures. We have not investigated the related problem of *weighted uniform reliability* [3], which is the restricted case of probabilistic query evaluation where we impose that all facts of the input TID must have some fixed probability different from $1/2$. We expect that our hardness result should extend to this problem when the fixed probability is the same across all relations (and is different from 0 and 1). It seems more challenging to understand the setting where the fixed probability can depend on the relation, in particular if we can require some relations to be deterministic, i.e., only have tuples with probability 1. In this setting, some unbounded homomorphism-closed queries would become tractable (e.g., Datalog queries that involve only the deterministic relations), and it is not clear what one can hope to show.

Coming back to the problem of (non-weighted) uniform reliability, an ambitious direction for future work would be to extend our intractability result towards Conjecture 1.2. The two remaining obstacles are the case of unbounded queries on arbitrary signatures, which we intend to study in future work; and the case of bounded queries, i.e., UCQs, where the general case is left open by Kenig and Suciu [8].

Other natural extensions include the study of queries satisfying weaker requirements than closure under homomorphisms; or other notions of possible worlds, e.g., induced subinstances; or other notions of intractability, e.g., the inexistence of lineages in tractable circuit classes from knowledge compilation. Another broad question is whether the techniques developed here have any connection to other areas of research, e.g., constraint satisfaction problems (CSPs).

References

- 1 Antoine Amarilli. Uniform reliability for unbounded homomorphism-closed graph queries. Full version with proofs, 2023. [arXiv:2209.11177](https://arxiv.org/abs/2209.11177).
- 2 Antoine Amarilli and İsmail İlkan Ceylan. The dichotomy of evaluating homomorphism-closed queries on probabilistic graphs. *LMCS*, 2021. [arXiv:1910.02048](https://arxiv.org/abs/1910.02048), doi:10.46298/lmcs-18(1:2)2022.
- 3 Antoine Amarilli and Benny Kimelfeld. Uniform reliability of self-join-free conjunctive queries. *LMCS*, 18(4), 2022. [arXiv:1908.07093](https://arxiv.org/abs/1908.07093), doi:10.46298/lmcs-18(4:3)2022.
- 4 Stephen A. Cook. The complexity of theorem-proving procedures. In *Proc. STOC*, 1971. URL: <https://www.cs.toronto.edu/~sacook/homepage/1971.pdf>.
- 5 Nilesh N. Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM*, 59(6):30, 2012. URL: <https://homes.cs.washington.edu/~suciu/jacm-dichotomy.pdf>.
- 6 Robert Fink and Dan Olteanu. Dichotomies for queries with negation in probabilistic databases. *TODS*, 41(1), 2016. URL: <http://www.cs.ox.ac.uk/people/Dan.Olteanu/papers/fo-tods16.pdf>.
- 7 Erich Grädel, Yuri Gurevich, and Colin Hirsch. The complexity of query reliability. In *Proc. PODS*, 1998. URL: https://www.researchgate.net/profile/Yuri_Gurevich2/publication/2900852_The_Complexity_of_Query_Reliability/links/0c96053321102376cd000000/The-Complexity-of-Query-Reliability.pdf.
- 8 Batya Kenig and Dan Suciu. A dichotomy for the generalized model counting problem for unions of conjunctive queries. In *Proc. PODS*, 2021. [arXiv:2008.00896](https://arxiv.org/abs/2008.00896).
- 9 Ester Livshits, Leopoldo E. Bertossi, Benny Kimelfeld, and Moshe Sebag. The Shapley value of tuples in query answering. In *Proc. ICDT*, volume 155, 2020. doi:10.4230/LIPIcs.ICDT.2020.20.
- 10 Dan Olteanu and Jiewen Huang. Using OBDDs for efficient query evaluation on probabilistic databases. In *Proc. SUM*, volume 5291, 2008. URL: <https://www.cs.ox.ac.uk/people/dan.olteanu/papers/oh-sum08.pdf>.
- 11 Dan Olteanu and Jiewen Huang. Secondary-storage confidence computation for conjunctive queries with inequalities. In *Proc. SIGMOD*, 2009. URL: <https://www.cs.ox.ac.uk/people/dan.olteanu/papers/oh-sigmod09.pdf>.
- 12 J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4), 1983.
- 13 Babak Salimi, Leopoldo E. Bertossi, Dan Suciu, and Guy Van den Broeck. Quantifying causal effects on query answering in databases. In *TAPP*, 2016. [arXiv:1603.02705](https://arxiv.org/abs/1603.02705).
- 14 Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
- 15 Leslie Gabriel Valiant. The complexity of computing the permanent. *TCS*, 8(2):189–201, 1979. doi:10.1016/0304-3975(79)90044-6.