A Trichotomy in the Data Complexity of Certain Query Answering for Conjunctive Queries

Paraschos Koutris[1] and Jef Wijsen[2]

[1]University of Washington, Seattle, USA
[2]University of Mons, Belgium

Abstract

A relational database is said to be uncertain if primary key constraints can possibly be violated. A repair (or possible world) of an uncertain database is obtained by selecting a maximal number of tuples without ever selecting two distinct tuples with the same primary key value. For any Boolean query q, $\mathsf{CERTAINTY}(q)$ is the problem that takes an uncertain database $\mathbf{db}$ on input, and asks whether q is true in every repair of $\mathbf{db}$. The complexity of this problem has been particularly studied for q ranging over the class of self-join-free Boolean conjunctive queries. A research challenge is to determine, given q, whether $\mathsf{CERTAINTY}(q)$ belongs to complexity classes $\mathbf{FO}$, $\mathbf{P}$, or $\mathbf{coNP}$-complete. In this paper, we combine existing techniques for studying the above complexity classification task. We show that for any self-join-free Boolean conjunctive query q, it can be decided whether or not $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$. Further, for any self-join-free Boolean conjunctive query q, $\mathsf{CERTAINTY}(q)$ is either in $\mathbf{P}$ or $\mathbf{coNP}$-complete, and the complexity dichotomy is effective. This settles a research question that has been open for ten years, since [9].

1 Introduction

Primary key violations provide an elementary means for capturing uncertainty in the relational data model. A *block* is a maximal set of tuples of the same relation that agree on the primary key of the relation. Tuples in the same block are mutually exclusive: exactly one tuple is true, but we are uncertain about which one. We will refer to databases as "uncertain databases" to stress that they can violate primary key constraints.

A *repair* (or possible world) of an uncertain database is obtained by selecting exactly one tuple from each block. In general, the number of repairs of an uncertain database can be exponential in its size. For instance, if an uncertain database contains n blocks with two tuples each, then it contains $2n$ tuples and has 2^n repairs.

There are two natural semantics for answering Boolean queries q on an uncertain database. Under the *possibility semantics*, the question is whether the query evaluates to true on some repair. Under the *certainty semantics*, which is adopted in this paper, the question is whether the query evaluates to true on every repair. The certainty semantics adheres to the paradigm of *consistent query answering* [2, 5], which introduces the notion of database repairs with respect to general integrity constraints. In this work, repairing is exclusively with respect to primary key constraints, one per relation.

For any Boolean query q, the decision problem $\mathsf{CERTAINTY}(q)$ is the following.

PROBLEM:	$\mathsf{CERTAINTY}(q)$
INPUT:	uncertain database $\mathbf{db}$
QUESTION:	Does every repair of $\mathbf{db}$ satisfy q?

Three comments are in place. First, the Boolean query q is not part of the input. Every Boolean query q gives thus rise to a new problem. Since the input to $\mathsf{CERTAINTY}(q)$ is an uncertain database, we consider the *data complexity* of the problem. Second, we will assume that every relation name in q or $\mathbf{db}$ has a fixed known arity and primary key. The primary key constraints are thus implicitly present in all problems. Third, all the complexity results obtained in this paper can be carried over to non-Boolean queries; the restriction to Boolean queries eases the technical treatment, but is not fundamental.

The complexity of $\mathsf{CERTAINTY}(q)$ has gained considerable research attention in recent years, especially for q ranging over the set of self-join-free conjunctive queries. A challenging question is to distinguish queries q for which the problem $\mathsf{CERTAINTY}(q)$ is tractable from queries for which the problem is intractable. Further, if $\mathsf{CERTAINTY}(q)$ is tractable, one may ask whether it is first-order expressible. We will refer to these questions as the *complexity classification task of* $\mathsf{CERTAINTY}(q)$.

In the past decade, a variety of tools and techniques have been used in the complexity classification task of $\mathsf{CERTAINTY}(q)$ for self-join-free conjunctive queries q. In their pioneering work, Fuxman and Miller [9] introduced the notion of *join graph* (not to be confused with the classical notion of join tree). Later on, Wijsen [14] introduced the notion of *attack graph*. Kolaitis and Pema [10] applied Minty's algorithm [13] to the task. Koutris and Suciu [11] introduced the notion of *query graph* and the distinction between consistent and possibly inconsistent relations. All these techniques have limited applicability: join graphs seem too rudimentary to obtain general complexity dichotomies; attack graphs enable to characterize first-order expressibility of $\mathsf{CERTAINTY}(q)$, but only for acyclic (in the sense of [4]) queries q; Minty's algorithm has been used to establish a $\mathbf{P}$-$\mathbf{coNP}$-complete dichotomy in the complexity of $\mathsf{CERTAINTY}(q)$, but only for queries q with exactly two atoms; the framework of Koutris and Suciu has also resulted in a $\mathbf{P}$-$\mathbf{coNP}$-complete dichotomy, but only when all primary keys consist of a single attribute. On top of the limited applicability of each individual technique, there is the difficulty that complexity classifications expressed in terms of different techniques cannot be easily compared.

In this paper, we make significant progress in the complexity classification task of $\mathsf{CERTAINTY}(q)$ for q ranging over the set of self-join-free conjunctive queries, by establishing the following effective complexity trichotomy:

- Given a self-join-free Boolean conjunctive query q, it is decidable whether $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$. In [14], this was only shown under the assumption that queries are acyclic (in the sense of [4]).
- Given a self-join-free Boolean conjunctive query q, if $\mathsf{CERTAINTY}(q)$ is not in $\mathbf{FO}$, then it is Ł-hard. In previous works [14, 16], Hanf locality was used to show first-order inexpressibility, resulting in involved proofs. The current paper takes a complexity-theoretic approach to first-order inexpressibility, which results in an easier proof of a stronger result.
- For every self-join-free Boolean conjunctive query, $\mathsf{CERTAINTY}(q)$ is either in $\mathbf{P}$ or $\mathbf{coNP}$-complete, and the dichotomy is effective. In [11], this was only shown under the assumption that all primary keys are simple (i.e., consist of a single attribute).

The established complexity trichotomy solves a problem that has been open since 2005 [9].

Organization This paper is organized as follows. Section 2 discusses related work. Section 3 introduces our data and query model. Section 4 defines attack graphs for Boolean conjunctive queries, extending an older notion of attack graph [16] that was defined exclusively for acyclic Boolean conjunctive queries. The section also states the main result of the paper, Theorem 2. Section 5 establishes an effective procedure that takes in a self-join-free Boolean conjunctive query q, and decides whether $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$. Section 6 provides a sufficient condition for $\mathbf{coNP}$-hardness of $\mathsf{CERTAINTY}(q)$, for any self-join-free Boolean conjunctive query q. Section 7 shows that if the condition is not satisfied, then $\mathsf{CERTAINTY}(q)$ is in $\mathbf{P}$. The appendix contains the proofs of some non-trivial results.

2 Related Work

Consistent query answering (CQA) goes back to the seminal work by Arenas, Bertossi, and Chomicki [2]. Fuxman and Miller [9] were the first ones to focus on CQA under the restrictions that consistency is only with respect to primary keys and that queries are self-join-free conjunctive. The term $\mathsf{CERTAINTY}(q)$ was coined in [14]. A recent and comprehensive survey on $\mathsf{CERTAINTY}(q)$ is [18].

Little is known about $\mathsf{CERTAINTY}(q)$ beyond self-join-free conjunctive queries. An interesting recent result by Fontaine [8] goes as follows. Let UCQ be the class of Boolean first-order queries that can be expressed as disjunctions of Boolean conjunctive queries (possibly with constants and self-joins). A daring conjecture is that for every query q in UCQ, $\mathsf{CERTAINTY}(q)$ is either in $\mathbf{P}$ or $\mathbf{coNP}$-complete. Fontaine showed that this conjecture implies Bulatov's dichotomy theorem for conservative CSP [6], the proof of which is highly involved (the full paper contains 66 pages).

3 Preliminaries

We assume disjoint sets of *variables* and *constants*. If $\vec{x}$ is a sequence containing variables and constants, then $\mathsf{vars}(\vec{x})$ denotes the set of variables that occur in $\vec{x}$. A *valuation* over a set U of variables is a total mapping θ from U to the set of constants. At several places, it is implicitly understood that such a valuation θ is extended to be the identity on constants and on variables not in U. If $V \subseteq U$, then $\theta[V]$ denotes the restriction of θ to V.

If θ is a valuation over a set U of variables, x is a variable, and a is a constant, then $\theta_{[x\mapsto a]}$ is the valuation over $U \cup \{x\}$ such that $\theta_{[x\mapsto a]}(x) = a$ and for every variable y such that $y \neq x$, $\theta_{[x\mapsto a]}(y) = \theta(y)$. Notice that $x \in U$ is allowed.

Atoms and key-equal facts Each *relation name* R of arity n, $n \geq 1$, has a unique *primary key* which is a set $\{1, 2, \ldots, k\}$ where $1 \leq k \leq n$. We say that R has *signature* $[n, k]$ if R has arity n and primary key $\{1, 2, \ldots, k\}$. We say that R is *simple-key* if $k = 1$. Elements of the primary key are called *primary-key positions*, while $k + 1$, $k + 2$, ..., n are *non-primary-key positions*. For all positive integers n, k such that $1 \leq k \leq n$, we assume denumerably many relation names with signature $[n, k]$.

If R is a relation name with signature $[n, k]$, then $R(s_1, \ldots, s_n)$ is called an *R-atom* (or simply atom), where each s_i is either a constant or a variable ($1 \leq i \leq n$). Such an atom is commonly written as $R(\underline{\vec{x}}, \vec{y})$ where the primary key value $\vec{x} = s_1, \ldots, s_k$ is underlined and $\vec{y} = s_{k+1}, \ldots, s_n$. An *$R$-fact* (or simply fact) is an R-atom in which no variable occurs. Two facts $R_1(\underline{\vec{a}_1}, \vec{b}_1), R_2(\underline{\vec{a}_2}, \vec{b}_2)$ are *key-equal* if $R_1 = R_2$ and $\vec{a}_1 = \vec{a}_2$. An R-atom or an R-fact is *simple-key* if R is simple-key.

We will use letters F, G, H for atoms. For an atom $F = R(\underline{\vec{x}}, \vec{y})$, we denote by $\mathsf{key}(F)$ the set of variables that occur in $\vec{x}$, and by $\mathsf{vars}(F)$ the set of variables that occur in F, that is, $\mathsf{key}(F) = \mathsf{vars}(\vec{x})$ and $\mathsf{vars}(F) = \mathsf{vars}(\vec{x}) \cup \mathsf{vars}(\vec{y})$.

Uncertain database, blocks, and repairs A *database schema* is a finite set of relation names. All constructs that follow are defined relative to a fixed database schema.

An *uncertain database* is a finite set $\mathbf{db}$ of facts using only the relation names of the schema. We refer to databases as "uncertain databases" to stress that such databases can violate primary key constraints.

We write $\mathbf{adom}(\mathbf{db})$ for the active domain of $\mathbf{db}$ (i.e., the set of constants that occur in $\mathbf{db}$). A *block* of $\mathbf{db}$ is a maximal set of key-equal facts of $\mathbf{db}$. The term R-block refers to a block of R-facts, i.e., facts with relation name R. If A is a fact of $\mathbf{db}$, then $\mathsf{block}(A, \mathbf{db})$ denotes the block of $\mathbf{db}$ that contains A. An uncertain database $\mathbf{db}$ is *consistent* if no two distinct facts are key-equal (i.e., if every block of $\mathbf{db}$ is a singleton). A *repair* of $\mathbf{db}$ is a maximal (with respect to set containment) consistent subset of $\mathbf{db}$. We write $\mathsf{rset}(\mathbf{db})$ for the set of repairs of $\mathbf{db}$.

Boolean conjunctive queries A *Boolean query* is a mapping q that associates a Boolean (true or false) to each uncertain database, such that q is closed under isomorphism [12]. We write $\mathbf{db} \models q$ to denote that q associates true to $\mathbf{db}$, in which case $\mathbf{db}$ is said to *satisfy* q. A *Boolean first-order query* is a Boolean query that can be defined in first-order logic. A *Boolean conjunctive query* is a finite set $q = \{R_1(\underline{\vec{x}_1}, \vec{y}_1), \ldots, R_n(\underline{\vec{x}_n}, \vec{y}_n)\}$ of atoms. We denote by $\mathsf{vars}(q)$ the set of variables that occur in q. The set q represents the first-order sentence

$$\exists u_1 \cdots \exists u_k \left(R_1(\underline{\vec{x}_1}, \vec{y}_1) \wedge \cdots \wedge R_n(\underline{\vec{x}_n}, \vec{y}_n)\right),$$

where $\{u_1, \ldots, u_k\} = \mathsf{vars}(q)$. This query q is satisfied by uncertain database $\mathbf{db}$ if there exists a valuation θ over $\mathsf{vars}(q)$ such that for each $i \in \{1, \ldots, n\}$, $R_i(\underline{\vec{a}}, \vec{b}) \in \mathbf{db}$ with $\vec{a} = \theta(\vec{x}_i)$ and $\vec{b} = \theta(\vec{y}_i)$.

We say that a Boolean conjunctive query q has a *self-join* if some relation name occurs more than once in q. If q has no self-join, then it is called *self-join-free*. By a little abuse of notation, we may confuse atoms with their relation names in a self-join-free Boolean conjunctive query q. That is, if we use a relation name R at places where an atom is expected, then we mean the (unique) R-atom of q.

If q is a Boolean conjunctive query, $\vec{x} = \langle x_1, \ldots, x_\ell \rangle$ is a sequence of distinct variables that occur in q, and $\vec{a} = \langle a_1, \ldots, a_\ell \rangle$ is a sequence of constants, then $q_{[\vec{x} \mapsto \vec{a}]}$ denotes the query obtained from q by replacing all occurrences of x_i with a_i, for all $1 \leq i \leq \ell$.

Typed uncertain databases For every variable x, we assume an infinite set of constants, denoted $\mathsf{type}(x)$, such that $x \neq y$ implies $\mathsf{type}(x) \cap \mathsf{type}(y) = \emptyset$. Let q be a self-join-free Boolean conjunctive query, and let $\mathbf{db}$ be an uncertain database. We say that $\mathbf{db}$ is *typed relative to* q if for every atom $R(x_1, \ldots, x_n)$ in q, for every $i \in \{1, \ldots, n\}$, if x_i is a variable, then for every fact $R(a_1, \ldots, a_n)$ in $\mathbf{db}$, $a_i \in \mathsf{type}(x_i)$ and the constant a_i does not occur in q. Significantly, since q is self-join-free, the assumption that uncertain databases are typed is without loss of generality.

Purified uncertain databases Let q be a Boolean conjunctive query, and let $\mathbf{db}$ be an uncertain database. We say that a fact $A \in \mathbf{db}$ is *relevant for* q *in* $\mathbf{db}$ if for some valuation θ over $\mathsf{vars}(q)$, $A \in \theta(q) \subseteq \mathbf{db}$. We say that $\mathbf{db}$ is *purified relative to* q if every fact $A \in \mathbf{db}$ is relevant for q in $\mathbf{db}$.

Frugal repairs For every uncertain database $\mathbf{db}$, Boolean conjunctive query q, and $X \subseteq \mathsf{vars}(q)$, we define a preorder $\preceq_q^X$ on $\mathsf{rset}(\mathbf{db})$, as follows. For every two repairs $\mathbf{r}_1, \mathbf{r}_2$, we define $\mathbf{r}_1 \preceq_q^X \mathbf{r}_2$ if for every valuation θ over X, $\mathbf{r}_1 \models \theta(q)$ implies $\mathbf{r}_2 \models \theta(q)$. Here, $\theta(q)$ is the query obtained from q by replacing all occurrences of each $x \in X$ with $\theta(x)$; variables not in X remain unaffected (i.e., θ is understood to be the identity on variables not in X). Clearly, $\preceq_q^X$ is a preorder (i.e., it is reflexive and transitive), and its minimal elements are called $\preceq_q^X$*-frugal repairs*.[1]

Functional dependencies Let q be a Boolean conjunctive query. A *functional dependency for* q is an expression $X \rightarrow Y$ where $X, Y \subseteq \mathsf{vars}(q)$. We say that an uncertain database $\mathbf{db}$ *satisfies* $X \rightarrow Y$ *for* q, denoted $\mathbf{db} \Vvdash_q X \rightarrow Y$, if for all valuations θ, μ over $\mathsf{vars}(q)$ such that $\theta(q), \mu(q) \subseteq \mathbf{db}$, if $\theta[X] = \mu[X]$, then $\theta[Y] = \mu[Y]$.

Example 1 The relation R shown next does not satisfy the standard functional dependency $2 \rightarrow 3$, because its tuples agree on the second position, but disagree on the third position. Nevertheless, for $q = \exists y \exists z R(a, y, z)$, we have $R \Vvdash_q y \rightarrow z$. The second tuple of R is not relevant for the query, because a and d are distinct constants; the relation R' is purified relative to q.

R	$\underline{1}$	2	3
	a	b	c
	d	b	f

R'	$\underline{1}$	2	3
	a	b	c

$\triangleleft$

Certain query answering For every Boolean conjunctive query q, the decision problem $\mathsf{CERTAINTY}(q)$ takes on input an uncertain database $\mathbf{db}$, and asks whether q is satisfied by every repair of $\mathbf{db}$.

It is easy to show the following upper bound on the complexity of $\mathsf{CERTAINTY}(q)$.

Theorem 1 *For every Boolean first-order query q,* $\mathsf{CERTAINTY}(q)$ *is in* $\mathbf{coNP}$.

The following two lemmas are useful in the study of the complexity of $\mathsf{CERTAINTY}(q)$.

Lemma 1 ([17]) *Let q be a Boolean conjunctive query. Let $\mathbf{db}$ be an uncertain database. It is possible to compute in polynomial time an uncertain database $\mathbf{db}'$ that is purified relative to q such that every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}'$ satisfies q.*

[1] $\mathbf{r}_1$ is minimal if for all $\mathbf{r}_2$, if $\mathbf{r}_2 \preceq_q^X \mathbf{r}_1$ then $\mathbf{r}_1 \preceq_q^X \mathbf{r}_2$.

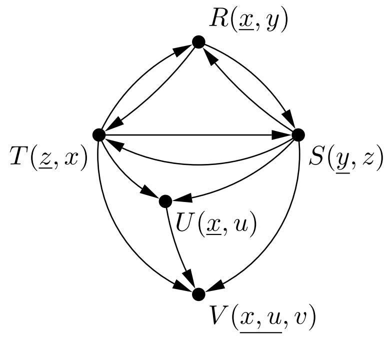

Figure 1: Attack graph of the query in Example 2.

Lemma 2 *Let q be a self-join-free Boolean conjunctive query, and $X \subseteq \mathsf{vars}(q)$. Let $\mathbf{db}$ be an uncertain database. Then, every repair of $\mathbf{db}$ satisfies q if and only if every $\preceq_q^X$-frugal repair of $\mathbf{db}$ satisfies q.*

4 Attack Graphs

Attack graphs were introduced in [14] for studying first-order expressibility of $\mathsf{CERTAINTY}(q)$ for acyclic (in the sense of [4]) self-join-free conjunctive queries q. Here, we extend the notion of attack graph to all (cyclic or acyclic) self-join-free conjunctive queries.

Let q be a self-join-free Boolean conjunctive query. We define $\mathcal{K}(q)$ as the following set of functional dependencies:

$$\mathcal{K}(q) := \{\mathsf{key}(F) \rightarrow \mathsf{vars}(F) \mid F \in q\}$$

For every atom $F \in q$, we define $F^{+,q}$ and $F^{\boxplus,q}$ as the following sets of variables.

$$\begin{array}{rcl} F^{+,q} & := & \{x \in \mathsf{vars}(q) \mid \mathcal{K}(q \setminus \{F\}) \models \mathsf{key}(F) \rightarrow x\} \\ F^{\boxplus,q} & := & \{x \in \mathsf{vars}(q) \mid \mathcal{K}(q) \models \mathsf{key}(F) \rightarrow x\} \end{array}$$

The *attack graph of* q is a directed graph whose vertices are the atoms of q. There is a directed edge from F to G ($F \neq G$) if there exists a sequence

$$F_0, F_1, \ldots, F_n \tag{1}$$

of (not necessarily distinct) atoms of q such that

- $F_0 = F$ and $F_n = G$; and
- for all $i \in \{0, \ldots, n-1\}$, $\mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1}) \nsubseteq F^{+,q}$.

A directed edge from F to G in the attack graph of q is also called an *attack from F to G*, denoted by $F \overset{q}{\rightsquigarrow} G$. The sequence (1) is called a *witness* for the attack $F \overset{q}{\rightsquigarrow} G$. We will often add variables to a witness: if we write $F_0 \overset{z_1}{\frown} F_1 \overset{z_2}{\frown} F_2 \ldots \overset{z_n}{\frown} F_n$, then it is understood that for $i \in \{1, \ldots, n\}$, $z_i \in \mathsf{vars}(F_{i-1}) \cap \mathsf{vars}(F_i)$ and $z_i \notin {F_0}^{+,q}$. If $F \overset{q}{\rightsquigarrow} G$, then we also say that F *attacks* G (or that G is attacked by F).

An attack from F to G is called *weak* if $\mathcal{K}(q) \models \mathsf{key}(F) \rightarrow \mathsf{key}(G)$; otherwise it is *strong*. A directed cycle in the attack graph of of q is called *weak* if all attacks in the cycle are weak; otherwise the cycle is called *strong*.

Example 2 Let $q = \{R(\underline{x}, y), S(\underline{y}, z), T(\underline{z}, x), U(\underline{x}, u), V(\underline{x, u}, v)\}$. By a little abuse of notation, we denote each atom by its relation name (e.g., R is used to denote the atom $R(\underline{x}, y)$). We have $R^{+,q} = \{x, u, v\}$. A witness for $R \overset{q}{\rightsquigarrow} T$ is $R \overset{y}{\frown} S \overset{z}{\frown} T$. The complete attack graph is shown in Fig. 1. All attacks are weak. $\lhd$

The above notion of attack graph is purely syntactic. Semantically, an attack from an R-atom to an S-atom in the attack graph of q means that there exists an uncertain database $\mathbf{db}$ such that every repair of $\mathbf{db}$ satisfies q, and such that two R-facts of a same R-block join exclusively with two S-facts belonging to distinct S-blocks. For the query of Example 2, such a database could be $\mathbf{db} = \{R(\underline{1}, a), R(\underline{1}, b), S(\underline{a}, \alpha), S(\underline{b}, \beta), \dots\}$, in which the two R-facts belong to the same R-block, and $R(\underline{1}, a)$ joins exclusively with $S(\underline{a}, \alpha)$, and $R(\underline{1}, b)$ joins exclusively with $S(\underline{b}, \beta)$, and the two S-facts belong to distinct S-blocks. Therefore, the attack graph of Fig. 1 contains a directed edge from the R-atom to the S-atom.

Equipped with the notion of attack graph, we can now present the effective complexity trichotomy in the set $\{\mathsf{CERTAINTY}(q) \mid q \text{ is a self-join-free Boolean conjunctive query}\}$.

Theorem 2 (Trichotomy Theorem) *Let q be a self-join-free Boolean conjunctive query.*

1. *If the attack graph of q is acyclic, then $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$.*
2. *If the attack graph of q is cyclic but contains no strong cycle, then $\mathsf{CERTAINTY}(q)$ is in $\mathbf{P}$ and is Ł-hard.*
3. *If the attack graph of q contains a strong cycle, then $\mathsf{CERTAINTY}(q)$ is $\mathbf{coNP}$-complete.*

The rest of the paper presents the proof of Theorem 2. We first present some properties of attack graphs that will be useful in subsequent sections.

Lemma 3 *Let q be a self-join-free Boolean conjunctive query. If $F \overset{q}{\rightsquigarrow} G$ and $G \overset{q}{\rightsquigarrow} H$, then either $F \overset{q}{\rightsquigarrow} H$ or $G \overset{q}{\rightsquigarrow} F$ (or both).*

Lemma 4 *Let q be a self-join-free Boolean conjunctive query.*

1. *If the attack graph of q contains a cycle, then it contains a cycle of size two.*
2. *If the attack graph of q contains a strong cycle, then it contains a strong cycle of size two.*

Lemma 5 *Let q be a self-join-free Boolean conjunctive query. Let $x \in \mathsf{vars}(q)$ and let a be an arbitrary constant.*

1. *If the attack graph of q is acyclic, then the attack graph of $q_{[x \mapsto a]}$ is acyclic.*
2. *If the attack graph of q contains no strong cycle, then the attack graph of $q_{[x \mapsto a]}$ contains no strong cycle.*

We conclude this section with three definitions. The following definition is taken from [3] and applies to directed graphs in general.

Definition 1 A directed graph is *strongly connected* if there is a directed path from any vertex to any other. The maximal strongly connected subgraphs of a graph are vertex-disjoint and are called its *strong components*. If S_1 and S_2 are strong components such that an edge leads from a vertex in S_1 to a vertex in S_2, then S_1 is a *predecessor* of S_2 and S_2 is a *successor* of S_1. A strong component is called *initial* if it has no predecessor. $\triangleleft$

Strong components in the attack graph should not be confused with strong attacks.

Example 3 In the attack graph of Fig. 1, the atoms $R(\underline{x}, y)$, $S(\underline{y}, z)$, and $T(\underline{z}, x)$ together constitute an initial strong component. $\triangleleft$

So far we have defined an attack from an atom to another atom. The following definition introduces attacks from an atom to a variable.

Definition 2 Let q be a self-join-free Boolean conjunctive query. Let R be a relation name with signature $[1, 1]$ such that R does not occur in q. For $F \in q$ and $z \in \mathsf{vars}(q)$, we say that F *attacks* z, denoted $F \overset{q}{\rightsquigarrow} z$, if $F \overset{q'}{\rightsquigarrow} R(\underline{z})$ where $q' = q \cup \{R(\underline{z})\}$. $\triangleleft$

Example 4 Clearly, if $F_0 \overset{z_1}{\frown} F_1 \dots \overset{z_n}{\frown} F_n$ is a witness for $F_0 \overset{q}{\rightsquigarrow} F_n$, then $F_0 \overset{q}{\rightsquigarrow} z_i$ for every $i \in \{1, \dots, n\}$. Notice also that if $q = \{R(\underline{x}, y)\}$, then the attack graph of q contains no edge, yet $R \overset{q}{\rightsquigarrow} y$. $\triangleleft$

Finally, we introduce the notion of *sequential proof*, which mimics an algorithm for testing logical implication for functional dependencies [1, Algorithm 8.2.7].

Definition 3 Let q be a self-join free Boolean conjunctive query. Let $X \subseteq \mathsf{vars}(q)$ and $y \in \mathsf{vars}(q)$. A *sequential proof* of $\mathcal{K}(q) \models X \rightarrow y$ is a sequence $H_0, H_1, \dots, H_\ell$ of atoms of q such that

- $y \in X \cup \bigcup_{i=1}^{\ell} \mathsf{vars}(H_i)$; and
- for $i \in \{0, \dots, \ell\}$, $\mathsf{key}(H_i) \subseteq X \cup \bigcup_{j=0}^{i-1} \mathsf{vars}(H_j)$.

Notice that if $y \in X$, then the empty sequence is a sequential proof of $\mathcal{K}(q) \models X \rightarrow y$. $\triangleleft$

5 First-Order Expressibility

In this section, we prove the first item in the statement of Theorem 2, as well as the Ł-hard lower complexity bound stated in the second item.

Theorem 3 *Let q be a self-join-free Boolean conjunctive query. Then the following are equivalent:*

1. $\mathsf{CERTAINTY}(q)$ *is in* **FO***;*
2. *the attack graph of q is acyclic.*

That is, acyclicity of the attack graph of q is both a necessary and sufficient condition for first-order expressibility of $\mathsf{CERTAINTY}(q)$. In Section 5.1, we show the contrapositive of the implication $1 \implies 2$. In Section 5.2, we show the implication $2 \implies 1$.

5.1 Necessary Condition

Let $q_0 = \{R_0(\underline{x}, y), S_0(\underline{y}, x)\}$. In [15], it was shown that $\mathsf{CERTAINTY}(q_0)$ is not in **FO**. The following lemma shows a stronger result.

Lemma 6 *Let $q_0 = \{R_0(\underline{x}, y), S_0(\underline{y}, x)\}$. Then $\mathsf{CERTAINTY}(q_0)$ is Ł-hard.*

Lemma 7 *Let q be a self-join-free Boolean conjunctive query. If the attack graph of q is cyclic, then $\mathsf{CERTAINTY}(q)$ is Ł-hard (and hence not in* **FO***).*

Proof Assume that the attack graph of q is cyclic. We show hereinafter that there exists a first-order many-one reduction from $\mathsf{CERTAINTY}(q_0)$ to $\mathsf{CERTAINTY}(q)$. The desired result then follows from Lemma 6.

By Lemma 4, we can assume two distinct atoms $F, G \in q$ such that $F \overset{q}{\rightsquigarrow} G \overset{q}{\rightsquigarrow} F$ is an attack cycle of size two. We will assume hereinafter that the relation names in F and G are R and S respectively.

For all constants a, b we define the valuation Θ_b^a over $\mathsf{vars}(q)$ as follows. Let $\perp$ be a fixed constant not occurring elsewhere. For every variable $u \in \mathsf{vars}(q)$,

1. if $u \in F^{+,q} \setminus G^{+,q}$, then $\Theta_b^a(u) = a$;
2. if $u \in G^{+,q} \setminus F^{+,q}$, then $\Theta_b^a(u) = b$;
3. if $u \in F^{+,q} \cap G^{+,q}$, then $\Theta_b^a(u) = \perp$;
4. if $u \in \mathsf{vars}(q) \setminus (F^{+,q} \cup G^{+,q})$, then $\Theta_b^a(u) = \langle a, b \rangle$.

Sublemma 1 *For all constants a, b, a', b', if $H \in q \setminus \{F, G\}$, then $\{\Theta_b^a(H), \Theta_{b'}^{a'}(H)\}$ is consistent.*

Proof of Sublemma 1 Assume that for all $u \in \mathsf{key}(H)$, $\Theta_b^a(u) = \Theta_{b'}^{a'}(u)$. We distinguish four cases.

Case $a = a'$ and $b = b'$. Then $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a = a'$ and $b \neq b'$. Then $\mathsf{key}(H) \subseteq F^{+,q}$, hence $\mathsf{vars}(H) \subseteq F^{+,q}$. Then $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a \neq a'$ and $b = b'$. Then $\mathsf{key}(H) \subseteq G^{+,q}$, hence $\mathsf{vars}(H) \subseteq G^{+,q}$. Then $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a \neq a'$ and $b \neq b'$. Then $\mathsf{key}(H) \subseteq F^{+,q} \cap G^{+,q}$, hence $\mathsf{vars}(H) \subseteq F^{+,q} \cap G^{+,q}$. Then $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

$\dashv$

Sublemma 2 *For all constants a, b, a', b',*

1. *$\Theta_b^a(F)$ and $\Theta_{b'}^{a'}(F)$ are key-equal if and only if $a = a'$.*
2. *$\Theta_b^a(F) = \Theta_{b'}^{a'}(F)$ if and only if $a = a'$ and $b = b'$.*

3. $\Theta^a_b(G)$ *and* $\Theta^{a'}_{b'}(G)$ *are key-equal if and only if* $b = b'$.

4. $\Theta^a_b(G) = \Theta^{a'}_{b'}(G)$ *if and only if* $a = a'$ *and* $b = b'$.

Proof of Sublemma 2

$\boxed{1. \implies}$ Consequence of $\mathsf{key}(F) \nsubseteq G^{+,q}$ (because $G \overset{q}{\rightsquigarrow} F$). $\boxed{1. \impliedby}$ Consequence of $\mathsf{key}(F) \subseteq F^{+,q}$.

$\boxed{2. \implies}$ Consequence of $\mathsf{vars}(F) \nsubseteq F^{+,q}$ (because $F \overset{q}{\rightsquigarrow} G$). $\boxed{2. \impliedby}$ Trivial.

The proof of the remaining items is analogous. ⊣

For every uncertain database $\mathbf{db}$ with R_0-facts and S_0-facts, we define $f(\mathbf{db})$ as the following uncertain database:

1. for every $R_0(\underline{a}, b)$ in $\mathbf{db}$, $f(\mathbf{db})$ contains $\Theta^a_b(q \setminus \{G\})$; and
2. for every $S_0(\underline{b}, a)$ in $\mathbf{db}$, $f(\mathbf{db})$ contains $\Theta^a_b(q \setminus \{F\})$.

It is easy to see that f is computable in $\mathbf{FO}$.

In what follows, we assume that $\mathbf{db}$ is typed, as explained in Section 3. It will be understood that $a, a_1, a_2, \dots$ belong to $\mathsf{type}(x)$, and that $b, b_1, b_2, \dots$ belong to $\mathsf{type}(y)$.

Let us define $g(\mathbf{db})$ as follows:

$$g(\mathbf{db}) := f(\mathbf{db}) \setminus \left(\{\Theta^a_b(F) \mid R_0(\underline{a}, b) \in \mathbf{db}\} \cup \{\Theta^a_b(G) \mid S_0(\underline{b}, a) \in \mathbf{db}\}\right).$$

That is, $g(\mathbf{db})$ contains all facts of $f(\mathbf{db})$ that are neither R-facts nor S-facts.

By Sublemmas 1 and 2,

$$\mathsf{rset}(f(\mathbf{db})) = \{f(\mathbf{r}) \cup g(\mathbf{db}) \mid \mathbf{r} \in \mathsf{rset}(\mathbf{db})\}. \tag{2}$$

Let $\mathbf{db}$ be an arbitrary database with R_0-facts and S_0-facts. It suffices to show that the following are equivalent for every repair $\mathbf{r}$ of $\mathbf{db}$:

1. $\mathbf{r}$ satisfies q_0;
2. $f(\mathbf{r}) \cup g(\mathbf{db})$ satisfies q.

$\boxed{1 \implies 2}$ This is the easier part.

$\boxed{2 \implies 1}$ Let θ be a substitution over $\mathsf{vars}(q)$ such that $\theta(q) \subseteq f(\mathbf{r}) \cup g(\mathbf{db})$.

By our construction, we can assume $R_0(\underline{a}, b) \in \mathbf{r}$ such that $\theta(F) \in \Theta^a_b(q \setminus \{G\})$. Likewise, we can assume $S_0(\underline{b'}, a') \in \mathbf{r}$ such that $\theta(G) \in \Theta^{a'}_{b'}(q \setminus \{F\})$.

It suffices to show that $a = a'$ and $b = b'$.

Before giving the proof, we provide some intuition. For every fact $A \in f(\mathbf{db})$, we can assume an atom in q, denoted H_A, such that $A = \Theta^a_b(H_A)$ for some constant $a \in \mathsf{type}(x)$ and some constant $b \in \mathsf{type}(y)$. Then, for all $z \in \mathsf{vars}(H_A)$, $\Theta^a_b(z) \in \{\bot, a, b, \langle a, b\rangle\}$. The constants in the latter set allow to "trace back" A to some facts $R_0(\underline{a}, b)$ or $S_0(\underline{b}, a)$ in $\mathbf{db}$.

With this intuition in mind, it is easy to show $b = b'$ (the proof of $a = a'$ is symmetrical). Since $F \overset{q}{\rightsquigarrow} G$, there exists a sequence $F_0, F_1, \dots, F_n$ of atoms of q such that

- $F_0 = F$ and $F_n = G$; and
- for all $i \in \{0, \dots, n-1\}$, we can assume $u_i \in \mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1})$ such that $u_i \notin F^{+,q}$.

We show by induction on increasing i that for all $i \in \{0, \dots, n-1\}$, there exists constant a_i such that for all $w_i \in \mathsf{vars}(F_i)$, we have $\theta(w_i) \in \{\bot, a_i, b, \langle a_i, b\rangle\}$.

Basis $i = 0$**.** Since $\theta(F) \in \Theta^a_b(q \setminus \{G\})$, for all $w_0 \in \mathsf{vars}(F_0)$, we have $\theta(w_0) \in \{\bot, a, b, \langle a, b\rangle\}$.

Step $i \to i+1$**.** By the induction hypothesis, there exists constant a_i such that for all $w_i \in \mathsf{vars}(F_i)$, we have $\theta(w_i) \in \{\bot, a_i, b, \langle a_i, b\rangle\}$.

From $u_i \notin F^{+,q}$, it follows that $\theta(u_i) \in \{b, \langle a_i, b\rangle\}$.

Since $u_i \in \mathsf{vars}(F_{i+1})$, it follows that there exists constant a_{i+1} such that for all $w_{i+1} \in \mathsf{vars}(F_{i+1})$, we have $\theta(w_{i+1}) \in \{\bot, a_{i+1}, b, \langle a_{i+1}, b\rangle\}$.

It follows that for $u_{n-1} \in \mathsf{vars}(G)$, there exists constant a_{n-1} such that $\theta(u_{n-1}) \in \{b, \langle a_{n-1}, b\rangle\}$. From $\theta(G) \in \Theta_{b'}^{a'}(q \setminus \{F\})$, it follows $\theta(u_{n-1}) \in \{b', \langle a', b'\rangle\}$. Consequently, $b = b'$. □

5.2 Sufficient Condition

In this section, we show that $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$ if the attack graph of q is acyclic.

Lemma 8 *Let q be a self-join-free Boolean conjunctive query. Let F be an atom of q such that in the attack graph of q, the indegree of F is zero. Let $k = |\mathsf{key}(F)|$ and let $\vec{x} = (x_1, \dots, x_k)$ be a sequence containing (exactly once) each variable of $\mathsf{key}(F)$. Then the following are equivalent for every uncertain database $\mathbf{db}$:*

1. *q is true in every repair of $\mathbf{db}$;*
2. *for some $\vec{a} \in (\mathbf{adom}(\mathbf{db}))^k$, it is the case that $q_{[\vec{x}\mapsto\vec{a}]}$ is true in every repair of $\mathbf{db}$.*

Lemma 8 immediately leads to the following result.

Lemma 9 *Let q be a self-join-free Boolean conjunctive query. If the attack graph of q is acyclic, then $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$.*

Proof Assume that the attack graph of q is acyclic.

The proof runs by induction on $|q|$. If $|q| = 0$, then $\mathsf{CERTAINTY}(q)$ is obviously in $\mathbf{FO}$.

Let $\mathbf{db}$ be an instance of $\mathsf{CERTAINTY}(q)$. Since the attack graph of q is acyclic, we can assume an atom $R(\underline{\vec{x}}, \vec{y})$ that is not attacked in the attack graph of q. By Lemma 8, the following are equivalent:

1. q is true in every repair of $\mathbf{db}$.
2. For some fact $R(\underline{\vec{a}}, \vec{b}) \in \mathbf{db}$, there exists of a valuation θ over $\mathsf{vars}(\vec{x})$ such that $\theta(\vec{x}) = \vec{a}$ and such that for all key-equal facts $R(\underline{\vec{a}}, \vec{b}')$ in $\mathbf{db}$, the valuation θ can be extended to a valuation θ^+ over $\mathsf{vars}(\vec{x}) \cup \mathsf{vars}(\vec{y})$ such that $\theta^+(\vec{y}) = \vec{b}$ and $\theta^+(q')$ is true in every repair of $\mathbf{db}$, where $q' = q \setminus \{R(\underline{\vec{x}}, \vec{y})\}$.

From Lemma 5, it follows that the attack graph of $\theta^+(q')$ is acyclic, and hence $\mathsf{CERTAINTY}(\theta^+(q'))$ is in $\mathbf{FO}$ by the induction hypothesis. It is then clear that the latter condition (2) can be checked in $\mathbf{FO}$. □

For a self-join-free Boolean conjunctive query q, the problem $\mathsf{CERTAINTY}(q)$ can be equivalently defined as the set containing every uncertain database $\mathbf{db}$ such that every repair of $\mathbf{db}$ satisfies q. If $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$, then the set $\mathsf{CERTAINTY}(q)$ is definable in first-order logic (by definition of the complexity class $\mathbf{FO}$). If $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$, then its first-order definition is commonly called *first-order rewriting*. Such a first-order rewriting is actually an implementation, in first-order logic, of the algorithm in the proof of Lemma 9. This is illustrated next.

Example 5 Let $q = \{R(\underline{x}, y), S(\underline{y}, b)\}$, where b is a constant. The attack graph of q contains a single directed edge, from the R-atom to the S-atom. The first-order definition of $\mathsf{CERTAINTY}(q)$ is as follows:

$$\begin{aligned}&\exists x \exists y (R(\underline{x}, y) \wedge \\ &\quad \forall y \left(R(\underline{x}, y) \rightarrow \left(S(\underline{y}, b) \wedge \forall z \left(S(\underline{y}, z) \rightarrow z = b\right)\right)\right)) .\end{aligned}$$

◁

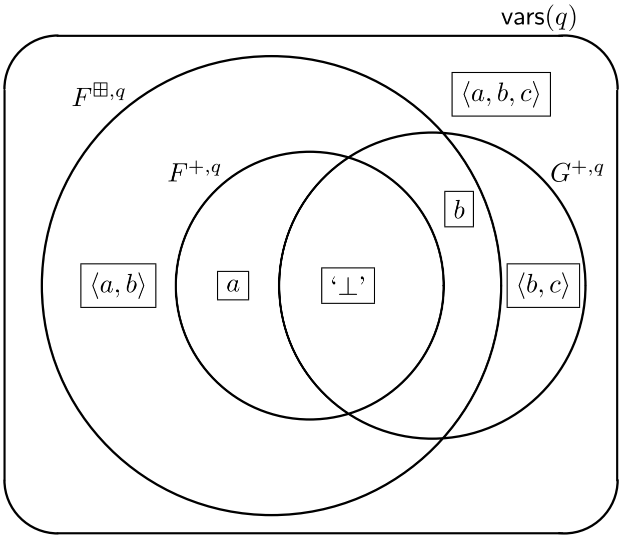

Figure 2: Help for the proof of Theorem 4.

6 Intractability Result

In this section, we prove the **coNP**-hard lower complexity bound stated in the third item of Theorem 2.

Theorem 4 *Let q be a self-join-free Boolean conjunctive query. If the attack graph of q contains a strong cycle, then* $\mathsf{CERTAINTY}(q)$ *is* **coNP***-hard.*

Proof Assume that the attack graph of q contains a strong cycle. By Lemma 4, we can assume $F, G \in q$ such that $F \overset{q}{\rightsquigarrow} G \overset{q}{\rightsquigarrow} F$ and the attack $F \overset{q}{\rightsquigarrow} G$ is strong. We will assume hereinafter that the relation names in F and G are R and S respectively.

Let $q_1 = \{R_1(\underline{x}, y), S_1(\underline{y, z}, x)\}$. We show hereinafter that there exists a polynomial-time (and even first-order) many-one reduction from $\mathsf{CERTAINTY}(q_1)$ to $\mathsf{CERTAINTY}(q)$. Since it is known [10] that $\mathsf{CERTAINTY}(q_1)$ is **coNP**-hard, it follows that $\mathsf{CERTAINTY}(q)$ is **coNP**-hard.

For all constants a, b, c, we define $\Theta^a_{b,c}$ as the following valuation over $\mathsf{vars}(q)$ (see Fig. 2 for a mnemonic). Let $\perp$ be some fixed constant.

1. If $u \in F^{+,q} \cap G^{+,q}$, then $\Theta^a_{b,c}(u) = \perp$;
2. if $u \in F^{+,q} \setminus G^{+,q}$, then $\Theta^a_{b,c}(u) = a$;
3. if $u \in G^{+,q} \setminus F^{\boxplus,q}$, then $\Theta^a_{b,c}(u) = \langle b, c\rangle$;
4. if $u \in \left(G^{+,q} \cap F^{\boxplus,q}\right) \setminus F^{+,q}$, then $\Theta^a_{b,c}(u) = b$;
5. if $u \in F^{\boxplus,q} \setminus (F^{+,q} \cup G^{+,q})$, then $\Theta^a_{b,c}(u) = \langle a, b\rangle$; and
6. if $u \notin F^{\boxplus,q} \cup G^{+,q}$, then $\Theta^a_{b,c}(u) = \langle a, b, c\rangle$.

Sublemma 3 *For all constants a, b, c, a', b', c', if $H \in q \setminus \{F, G\}$, then $\{\Theta^a_{b,c}(H), \Theta^{a'}_{b',c'}(H)\}$ is consistent.*

Proof of Sublemma 1 Assume that for all $u \in \mathsf{key}(H)$,

$$\Theta^a_{b,c}(u) = \Theta^{a'}_{b',c'}(u). \tag{3}$$

We distinguish four cases.

Case $a = a'$ **and** $b = b'$**.** If $c = c'$, then $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$. Assume next $c \neq c'$. From (3), it follows $\mathsf{key}(H) \subseteq F^{\boxplus,q}$. Consequently, $\mathsf{vars}(H) \subseteq F^{\boxplus,q}$. Since c does not occur inside $F^{\boxplus,q}$ in the Venn diagram of Fig. 2, we have $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$.

Case $a = a'$ **and** $b \neq b'$**.** From (3), it follows $\mathsf{key}(H) \subseteq F^{+,q}$, hence $\mathsf{vars}(H) \subseteq F^{+,q}$. Since b and c do not occur inside $F^{+,q}$ in the Venn diagram, $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$.

Case $a \neq a'$ **and** $b = b'$**.** First assume $c = c'$. From (3), it follows $\mathsf{key}(H) \subseteq G^{+,q}$, hence $\mathsf{vars}(H) \subseteq G^{+,q}$. Since c does not occur inside $G^{+,q}$ in the Venn diagram, $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$.

Next assume $c \neq c'$. From (3), it follows $\mathsf{key}(H) \subseteq F^{\boxplus,q} \cap G^{+,q}$, hence $\mathsf{vars}(H) \subseteq F^{\boxplus,q} \cap G^{+,q}$. Since a and c do not occur inside $F^{\boxplus,q} \cap G^{+,q}$ in the Venn diagram, $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$.

Case $a \neq a'$ **and** $b \neq b'$**.** From (3), it follows $\mathsf{key}(H) \subseteq F^{+,q} \cap G^{+,q}$, hence $\mathsf{vars}(H) \subseteq F^{+,q} \cap G^{+,q}$. Since a, b, c do not occur inside $F^{+,q} \cap G^{+,q}$ in the Venn diagram, $\Theta^{a}_{b,c}(H) = \Theta^{a'}_{b',c'}(H)$.

$\dashv$

Sublemma 4 *For all constants a, b, c, a', b', c',*

1. *$\Theta^{a}_{b,c}(F)$ and $\Theta^{a'}_{b',c'}(F)$ are key-equal iff $a = a'$.*
2. *$\Theta^{a}_{b,c}(F) = \Theta^{a'}_{b',c'}(F)$ iff $a = a'$ and $b = b'$.*
3. *$\Theta^{a}_{b,c}(G)$ and $\Theta^{a'}_{b',c'}(G)$ are key-equal iff $b = b'$ and $c = c'$.*
4. *$\Theta^{a}_{b,c}(G) = \Theta^{a'}_{b',c'}(G)$ iff $a = a'$ and $b = b'$ and $c = c'$.*

Proof of Sublemma 4

$\boxed{1. \implies}$ Consequence of $\mathsf{key}(F) \nsubseteq G^{+,q}$ (because $G \overset{q}{\rightsquigarrow} F$). $\boxed{1. \impliedby}$ Consequence of $\mathsf{key}(F) \subseteq F^{+,q}$.

$\boxed{2. \implies}$ Consequence of $\mathsf{vars}(F) \nsubseteq F^{+,q}$ (because $F \overset{q}{\rightsquigarrow} G$). $\boxed{2. \impliedby}$ Consequence of $\mathsf{vars}(F) \subseteq F^{\boxplus,q}$.

$\boxed{3. \implies}$ Consequence of $\mathsf{key}(G) \nsubseteq F^{\boxplus,q}$ (because $F \overset{q}{\rightsquigarrow} G$ is a strong attack). $\boxed{3. \impliedby}$ Consequence of $\mathsf{key}(G) \subseteq G^{+,q}$.

$\boxed{4. \implies}$ Consequence of item 3 and $\mathsf{vars}(G) \nsubseteq G^{+,q}$ (because $G \overset{q}{\rightsquigarrow} F$). $\boxed{4. \impliedby}$ Trivial. $\dashv$

Let $\mathbf{db}$ be uncertain database with R_1-facts and S_1-facts. In what follows, we assume that $\mathbf{db}$ is typed, as explained in Section 3. It will be understood that $a, a_1, a_2, \ldots$ belong to $\mathsf{type}(x)$, that $b, b_1, b_2, \ldots$ belong to $\mathsf{type}(y)$, and that $c, c_1, c_2, \ldots$ belong to $\mathsf{type}(z)$.

Let $h(\mathbf{db})$ be the subset of $\mathbf{db}$ such that

1. $h(\mathbf{db})$contains all S_1-facts of $\mathbf{db}$; and
2. $h(\mathbf{db})$ contains every R_1-block $\mathbf{b}$ of $\mathbf{db}$ such that for every fact $R_1(\underline{a}, b)$ in $\mathbf{b}$, there exists some constant c such that $S_1(\underline{b, c}, a)$ is in $\mathbf{db}$.

Clearly, the computation of $h(\mathbf{db})$ from $\mathbf{db}$ is in $\mathbf{FO}$, and the following are equivalent:

1. every repair of $\mathbf{db}$ satisfies q_1;
2. every repair of $h(\mathbf{db})$ satisfies q_1.

We define $f(\mathbf{db})$ as the following uncertain database:

1. for every pair $\{R_1(\underline{a}, b), S_1(\underline{b, c}, a)\}$ contained in $h(\mathbf{db})$, $f(\mathbf{db})$ contains $\Theta^{a}_{b,c}(q \setminus \{G\})$; and
2. for every $S_1(\underline{b, c}, a)$ in $h(\mathbf{db})$, $f(\mathbf{db})$ contains $\Theta^{a}_{b,c}(q \setminus \{F\})$.

It is easy to see that f is computable in $\mathbf{FO}$.

Let $g(\mathbf{db})$ be the subset of $f(\mathbf{db})$ containing all facts of $f(\mathbf{db})$ that are neither R-facts nor S-facts.

By Sublemmas 3 and 4,

$$\mathsf{rset}(f(\mathbf{db})) = \{f(\mathbf{r}) \cup g(\mathbf{db}) \mid \mathbf{r} \in \mathsf{rset}(\mathbf{db})\}. \tag{4}$$

Let $\mathbf{db}$ be an arbitrary database with R_1-facts and S_1-facts. It suffices to show that the following are equivalent for every repair $\mathbf{r}$ of $\mathbf{db}$:

1. $\mathbf{r}$ satisfies q_1;
2. $f(\mathbf{r}) \cup g(\mathbf{db})$ satisfies q.

$\boxed{1 \Longrightarrow 2}$ This is the easier part.

$\boxed{2 \Longrightarrow 1}$ Let θ be a substitution over $\mathsf{vars}(q)$ such that $\theta(q) \subseteq f(\mathbf{r}) \cup g(\mathbf{db})$. By our construction, we can assume $R_1(\underline{a}, b) \in \mathbf{r}$ and some constant c such that $\theta(F) \in \Theta^{a}_{b,c}(q \setminus \{G\})$. Likewise, we can assume $S_1(\underline{b', c'}, a') \in \mathbf{r}$ such that $\theta(G) \in \Theta^{a'}_{b',c'}(q \setminus \{F\})$. It suffices to show that $a = a'$ and $b = b'$.

$\boxed{b = b'}$ Since $F \overset{q}{\rightsquigarrow} G$, there exists a sequence $F_0, F_1, \ldots, F_n$ of distinct atoms of q such that

- $F_0 = F$ and $F_n = G$; and
- for all $i \in \{0, \ldots, n-1\}$, we can assume $u_i \in \mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1})$ such that $u_i \notin F^{+,q}$.

We show by induction on increasing i that for all $i \in \{0, \ldots, n-1\}$, there exist constants a_i and c_i such that for all $w_i \in \mathsf{vars}(F_i)$, we have $\theta(w_i) \in \{\bot, a_i, b, \langle a_i, b\rangle, \langle b, c_i\rangle, \langle a_i, b, c_i\rangle\}$.

Basis $i = 0$. Since $\theta(F) \in \Theta^{a}_{b,c}(q \setminus \{G\})$, for all $w_0 \in \mathsf{vars}(F_0)$, we have $\theta(w_0) \in \{\bot, a, b, \langle a, b\rangle, \langle b, c\rangle, \langle a, b, c\rangle\}$.

Step $i \rightarrow i+1$. By the induction hypothesis, there exist constants a_i and c_i such that for all $w_i \in \mathsf{vars}(F_i)$, we have $\theta(w_i) \in \{\bot, a_i, b, \langle a_i, b\rangle, \langle b, c_i\rangle, \langle a_i, b, c_i\rangle\}$.

From $u_i \notin F^{+,q}$, it follows that $\theta(u_i) \in \{b, \langle a_i, b\rangle, \langle b, c_i\rangle, \langle a_i, b, c_i\rangle\}$.

Since $u_i \in \mathsf{vars}(F_{i+1})$, it follows that there exist constants a_{i+1} and c_{i+1} such that for all $w_{i+1} \in \mathsf{vars}(F_{i+1})$, we have $\theta(w_{i+1}) \in \{\bot, a_{i+1}, b, \langle a_{i+1}, b\rangle, \langle b, c_{i+1}\rangle, \langle a_{i+1}, b, c_{i+1}\rangle\}$.

It follows that for $u_{n-1} \in \mathsf{vars}(G)$, there exist constants a_{n-1} and c_{n-1} such that $\theta(u_{n-1}) \in \{b, \langle a_{n-1}, b\rangle, \langle b, c_{n-1}\rangle, \langle a_{n-1}, b, c_{n-1}\rangle\}$. From $\theta(G) \in \Theta^{a'}_{b',c'}(q \setminus \{F\})$, it follows $\theta(u_{n-1}) \in \{b', \langle a', b'\rangle, \langle b', c'\rangle, \langle a', b', c'\rangle\}$. Consequently, $b = b'$.

$\boxed{a = a'}$ Analogous. $\square$

7 Polynomial Tractability

In this section, we prove the $\mathbf{P}$ upper complexity bound stated in the second item of Theorem 2.

Theorem 5 *Let q be a self-join-free Boolean conjunctive query. If the attack graph of q contains no strong cycle, then $\mathsf{CERTAINTY}(q)$ is in $\mathbf{P}$.*

Road map The proof of Theorem 5 is technically involved. We start by introducing in Section 7.1 an extension of the data model that allows some syntactic simplifications, expressed in Section 7.2. In Section 7.3, we introduce the notion of *Markov cycle*, and show how the "dissolution" of Markov cycles is helpful in the proof of Theorem 5, which is given in Section 7.4. The dissolution of Markov cycles is explained in detail in Section 7.5.

7.1 Relations Known to Be Consistent

We conservatively extend our data model. We first distinguish between two kinds of relation names: those that can be inconsistent, and those that cannot.

Relations known to be consistent Every relation name has a unique and fixed *mode*, which is an element in $\{i, c\}$. It will come in handy to think of i and c as inconsistent and consistent respectively. We often write R^c to denote that R is a relation name with mode c. If q is a self-join-free Boolean conjunctive query, then $[\![q]\!]$ denotes the subset of q containing each atom whose relation name has mode c. The *inconsistency count* of q, denoted $\mathsf{incnt}(q)$, is the number of relation names with mode i in q. Modes carry over to atoms and facts: the mode of an atom $R(\underline{\vec{x}}, \vec{y})$ or a fact $R(\underline{\vec{a}}, \vec{b})$ is the mode of R.

The intended semantics is that if a relation name R has mode c, then the set of R-facts of an uncertain database will always be consistent.

Certain query answering with consistent and inconsistent relations The problem $\mathsf{CERTAINTY}(q)$ now takes as input an uncertain database $\mathbf{db}$ such that for every relation name R in q, if R has mode c, then the set of R-facts of $\mathbf{db}$ is consistent. The problem is to determine whether every repair of $\mathbf{db}$ satisfies q.

All results shown in previous sections carry over to the new setting, by assuming that all relation names used so far had mode i. Furthermore, as stated by Proposition 1 (which has an easy proof), relation names with mode c can be simulated by means exclusively of relation names with mode i. Therefore, having relation names with mode c will be convenient, but is not fundamental.

Proposition 1 *Let q be a self-join free Boolean conjunctive query. Let $R^c(\underline{\vec{x}}, \vec{y})$ be an atom with mode c in q. Let R_1 and R_2 be two relation names, both with mode i and with the same signature as R, such that neither R_1 nor R_2 occurs in q. Let $q' = (q \setminus \{R^c(\underline{\vec{x}}, \vec{y})\}) \cup \{R_1(\underline{\vec{x}}, \vec{y}), R_2(\underline{\vec{x}}, \vec{y})\}$. Then $\mathsf{CERTAINTY}(q)$ and $\mathsf{CERTAINTY}(q')$ are equivalent under first-order reductions.*

If relation names with mode c are allowed for syntactic convenience, the definition of $F^{+,q}$ needs slight change:

$$F^{+,q} := \{x \in \mathsf{vars}(q) \mid \mathcal{K}((q \setminus F) \cup [\![q]\!]) \models \mathsf{key}(F) \rightarrow x\}$$

Modulo this redefinition, the notion of attack graph remains unchanged.

Proposition 1 explains how to replace atoms with mode c. Conversely, the following lemma states that in pursuing a proof for Theorem 5, there are cases where a self-join-free Boolean conjunctive query can be extended with atoms of mode c.

Lemma 10 *Let q be a self-join-free Boolean conjunctive query. Let $x, z \in \mathsf{vars}(q)$ such that $\mathcal{K}(q) \models x \rightarrow z$ and for every $F \in q$, if $\mathcal{K}(q) \models x \rightarrow \mathsf{key}(F)$, then $F \overset{q}{\not\rightsquigarrow} x$ and $F \overset{q}{\not\rightsquigarrow} z$. Let $q' = q \cup \{T^c(\underline{x}, z)\}$, where T is a fresh relation name with mode c. Then,*

1. *there exists a polynomial-time many-one reduction from $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q')$; and*
2. *if the attack graph of q contains no strong cycle, then the attack graph of q' contains no strong cycle either.*

Saturated queries Given a self-join-free Boolean conjunctive query, the reduction of Lemma 10 can be repeated until it can no longer be applied. The query so obtained will be called *saturated*.

Definition 4 Let q be a self-join-free Boolean conjunctive query. We say that q is *saturated* if whenever $x, z \in \mathsf{vars}(q)$ such that $\mathcal{K}(q) \models x \rightarrow z$ and $\mathcal{K}([\![q]\!]) \not\models x \rightarrow z$, then there exists an atom $F \in q$ with $\mathcal{K}(q) \models x \rightarrow \mathsf{key}(F)$ such that $F \overset{q}{\rightsquigarrow} x$ or $F \overset{q}{\rightsquigarrow} z$. ◁

Example 6 Consider the query $q = \{R(\underline{x}, y), S_1(\underline{y}, z), S_2(\underline{y}, z), T^c(\underline{x, z}, w), U(\underline{w}, x)\}$. We have $\mathcal{K}(q) \models y \rightarrow z$ and $\mathcal{K}([\![q]\!]) \not\models y \rightarrow z$. The set $\{F \in q \mid \mathcal{K}(q) \models y \rightarrow \mathsf{key}(F)\}$ equals $\{S_1, S_2\}$. We have neither $S_1 \overset{q}{\rightsquigarrow} y$ nor $S_1 \overset{q}{\rightsquigarrow} z$. Likewise, neither $S_2 \overset{q}{\rightsquigarrow} y$ nor $S_2 \overset{q}{\rightsquigarrow} z$. Hence, q is not saturated. By Lemma 10, there exists a polynomial-time many-one reduction from $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q')$ with $q' = q \cup \{S^c(\underline{y}, z)\}$, where S is a fresh relation name with mode c. It can be verified that the query q' is saturated. ◁

7.2 Syntactic Simplifications

The following lemma shows that any proof of Theorem 5 can assume some syntactic simplifications without loss of generality.

Lemma 11 *Let q be a self-join-free Boolean conjunctive query. There exists a polynomial-time many-one reduction from $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q')$ for some self-join-free Boolean conjunctive query q' with the following properties:*

- $\mathsf{incnt}(q') \leq \mathsf{incnt}(q)$*;*
- *no atom in q' contains two occurrences of the same variable;*
- *constants occur in q' exclusively at the primary-key position of simple-key atoms;*
- *every atom with mode i in q' is simple-key;*
- *q' is saturated; and*
- *if the the attack graph of q contains no strong cycle, then the attack graph of q' contains no strong cycle either.*

7.3 Dissolving Markov Cycles

The following definition introduces Markov graphs.

Definition 5 Let q be a self-join-free Boolean conjunctive query such that every atom with mode i in q is simple-key. For every $x \in \mathsf{vars}(q)$, we define

$$\mathsf{C}_q(x) := \{F \in q \mid F \text{ has mode } i \text{ and } \mathsf{key}(F) = \{x\}\}.$$

Notice that $\mathsf{C}_q(x)$ can be empty.

The *Markov graph* of q is a directed graph whose vertex set is $\mathsf{vars}(q)$. There is a directed edge from x to y, denoted $x \xrightarrow{q,\mathsf{M}} y$, if $x \neq y$ and $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!]) \models x \rightarrow y$. If the query q is clear from the context, then $x \xrightarrow{q,\mathsf{M}} y$ can be shortened into $x \xrightarrow{\mathsf{M}} y$. We write $x \xrightarrow{q,\mathsf{M}*} y$ (or $x \xrightarrow{\mathsf{M}*} y$ if q is clear from the context) if the Markov graph of q contains a directed path from x to y.[2] Notice that for every $x \in \mathsf{vars}(q)$, $x \xrightarrow{q,\mathsf{M}*} x$.

An elementary directed cycle $\mathcal{C}$ in the Markov graph of q is said to be *premier* if there exists a variable $x \in \mathsf{vars}(q)$ such that

1. $\{x\} = \mathsf{key}(F_0)$ for some atom F_0 with mode i that belongs to an initial strong component of the attack graph of q; and
2. for some y in $\mathcal{C}$, we have $x \xrightarrow{q,\mathsf{M}*} y$ and $\mathcal{K}(q) \models y \rightarrow x$.

The term *Markov edge* is used for an edge in the Markov graph; likewise for *Markov path* and *Markov cycle*. $\triangleleft$

Example 7 Let $q = \{R(\underline{x}, y, v),\ S(\underline{y}, x),\ V_1^c(\underline{v}, w),\ W(\underline{w}, v)\ V_2^c(\underline{w}, y)\}$. All atoms in q are simple-key. Then, $[\![q]\!] = \{V_1^c(\underline{v}, w),\ V_2^c(\underline{w}, y)\}$.

We have $\mathsf{C}_q(x) = \{R(\underline{x}, v, y)\}$. Since $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!]) \models x \rightarrow \{y, v, w\}$, the Markov graph of q contains directed edges from x to each of y, v, and w.

We have $\mathsf{C}_q(v) = \emptyset$. Since $\mathcal{K}(\mathsf{C}_q(v) \cup [\![q]\!]) \models v \rightarrow \{y, w\}$, the Markov graph of q contains directed edges from v to both y and w. The complete Markov graph of q is shown in Fig. 3 (right).

The attack graph of q is shown in Fig. 3 (left). The atoms $R(\underline{x}, y, v)$ and $S(\underline{y}, x)$ together constitute an initial strong component of the attack graph. It is then straightforward that each cycle in the Markov graph of q that contains x or y, must be premier. Further, the cycle v, w, v in the Markov graph of q is also premier, because there is a Markov path from x to v, and $\mathcal{K}(q) \models v \rightarrow x$. $\triangleleft$

[2] The term Markov refers to the intuition that in a Markov path, each variable functionally determines the next variable in the path, independently of preceding variables.

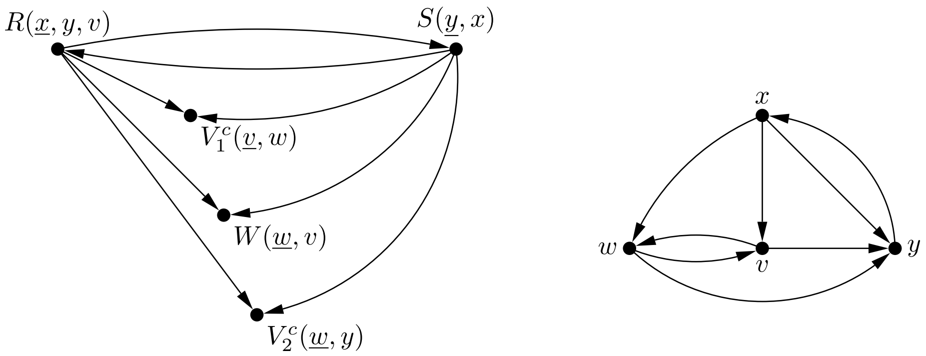

Figure 3: Attack graph (left) and Markov graph (right) of the query $\{R(\underline{x}, y, v),\ S(\underline{y}, x),\ V_1^c(\underline{v}, w),\ W(\underline{w}, v)\ V_2^c(\underline{w}, y)\}$.

Let q be like in Definition 5 and assume that the Markov graph of q contains an elementary directed cycle $\mathcal{C}$. Lemma 12 states that $\mathsf{CERTAINTY}(q)$ can be reduced in polynomial time to $\mathsf{CERTAINTY}(q^*)$, where q^* is obtained from q by "dissolving" the Markov cycle $\mathcal{C}$ as defined in Definition 6. Moreover, we will show (Lemma 13) that if $\mathcal{C}$ is premier and the attack graph of q contains no strong cycle, then the attack graph of q^* will contain no strong cycle either. The reduction that "dissolves" Markov cycles will be the central idea in our polynomial-time algorithm for $\mathsf{CERTAINTY}(q)$ when the attack graph of q contains no strong cycle.

Definition 6 Let q be a self-join-free Boolean conjunctive query such that every atom with mode i in q is simple-key. Let $\mathcal{C}$ be an elementary directed cycle of length $k \geq 2$ in the Markov graph of q. Then, $\mathsf{dissolve}(\mathcal{C}, q)$ denotes the self-join-free Boolean conjunctive query defined next. Let $x_0, \ldots, x_{k-1}$ be the variables in $\mathcal{C}$, and let $q_0 = \bigcup_{i=0}^{k-1} \mathsf{C}_q(x_i)$. Let $\vec{y}$ be a sequence of variables containing exactly once each variable of $\mathsf{vars}(q_0) \setminus \{x_0, \ldots, x_{k-1}\}$. Let $q_1 = \{T(\underline{u}, x_0, \ldots, x_{k-1}, \vec{y})\} \cup \{U_i^c(\underline{x_i}, u)\}_{i=0}^{k-1}$, where u is a fresh variable, T is a fresh relation name with mode i, and $U_1, \ldots, U_{k-1}$ are fresh relation names with mode c. Then, we define

$$\mathsf{dissolve}(\mathcal{C}, q) := (q \setminus q_0) \cup q_1.$$

Notice that $\mathsf{dissolve}(\mathcal{C}, q)$ is unique up to a renaming of the variable u and the relation names in q_1. $\lhd$

Example 8 Let q be the query of Fig. 3. Let $\mathcal{C}$ be the cycle x, w, y, x in the Markov graph of q. Using the notation of Definition 6, we have

$$\begin{aligned} q_0 &= \{R(\underline{x}, y, v), S(\underline{y}, x), W(\underline{w}, v)\} \\ q_1 &= \{T(\underline{u}, x, w, y, v), U_1^c(\underline{x}, u), U_2^c(\underline{w}, u), U_3^c(\underline{y}, u)\} \end{aligned}$$

Hence, $\mathsf{dissolve}(\mathcal{C}, q) = \{V_1^c(\underline{v}, w),\ V_2^c(\underline{w}, y),\ T(\underline{u}, x, w, y, v),\ U_1^c(\underline{x}, u),\ U_2^c(\underline{w}, u),\ U_3^c(\underline{y}, u)\}$. $\lhd$

Lemma 12 *Let q be a self-join-free Boolean conjunctive query such that every atom with mode i in q is simple-key. Let $\mathcal{C}$ be an elementary directed cycle in the Markov graph of q, and let $q^* = \mathsf{dissolve}(\mathcal{C}, q)$. Then, there exists a polynomial-time many-one reduction from $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q^*)$.*

The reduction of Lemma 12 will be explained in Section 7.5. To use the reduction in a proof of Theorem 5, two more results are needed:

- First, we need to show that the "dissolution" of Markov cycles can be done while keeping the attack graph free of strong cycles (this is Lemma 13). This turns out to be true only for Markov cycles that are premier (as defined in Definition 5).
- Second, we need to show the existence of premier Markov cycles that can be "dissolved" (this is Lemma 14).

Lemma 13 *Let q be a self-join-free Boolean conjunctive query such that every atom with mode i in q is simple-key. Let $\mathcal{C}$ be an elementary directed cycle in the Markov graph of q such that $\mathcal{C}$ is premier, and let $q^* = \mathsf{dissolve}(\mathcal{C}, q)$. If the attack graph of q contains no strong cycle, then the attack graph of q^* contains no strong cycle either.*

Lemma 14 *Let q be a self-join-free Boolean conjunctive query such that*

- *for every atom $F \in q$, if F has mode i, then F is simple-key and $\mathsf{key}(F) \neq \emptyset$;*
- *q is saturated;*
- *the attack graph of q contains no strong cycle; and*
- *the attack graph of q contains an initial strong component with two or more atoms.*

Then, the Markov graph of q contains an elementary directed cycle that is premier and such that for every y in $\mathcal{C}$, $\mathsf{C}_q(y) \neq \emptyset$.

The condition $\mathsf{C}_q(y) \neq \emptyset$, for every y in $\mathcal{C}$, guarantees that $\mathsf{dissolve}(\mathcal{C}, q)$ will contain strictly less atoms of mode i than q. This condition will be used in the proof of Theorem 5 which runs by induction on the number of atoms with mode i. The following example shows that Lemma 14 is no longer true if q is not saturated.

Example 9 Continuing Example 6. The query q of Example 6 is not saturated, but satisfies all other conditions in the statement of Lemma 14. In particular, the attack graph of q contains a weak cycle $R \overset{q}{\rightsquigarrow} U \overset{q}{\rightsquigarrow} R$, which is part of an initial strong component. The Markov graph of q consists of a single path $w \xrightarrow{q,\mathsf{M}} x \xrightarrow{q,\mathsf{M}} y \xrightarrow{q,\mathsf{M}} z$, and hence is acyclic.

The query q' of Example 6 is saturated, and we have $x \xrightarrow{q',\mathsf{M}} w \xrightarrow{q',\mathsf{M}} x$, a Markov cycle which can be shown to be premier. $\lhd$

7.4 The Proof of Theorem 5

Proof of Theorem 5 Assume that the attack graph of q contains no strong cycle. The proof runs by induction on increasing $\mathsf{incnt}(q)$. The desired result is obvious if $\mathsf{incnt}(q) = 0$. Assume that $\mathsf{incnt}(q) > 0$ in the remainder of the proof. Let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$.

First, we reduce in polynomial time $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q')$ with q' like in Lemma 11. We now distinguish two cases.

Case q' contains an atom F with mode i that has zero indegree in the attack graph of q. We can assume either $F = R(\underline{x}, \vec{y})$ or $F = R(\underline{a}, \vec{y})$, where $\vec{y}$ is a sequence of distinct variables. In the remainder, we treat the case $F = R(\underline{x}, \vec{y})$ (the case $F = R(\underline{a}, \vec{y})$ is even simpler).

Let $q'' = q' \setminus \{R(\underline{x}, \vec{y})\}$. By Lemma 8, every repair of $\mathbf{db}$ satisfies q' if and only if $\mathbf{db}$ includes an R-block $\mathbf{b}$ (there are only polynomially many such blocks) such for every $R(\underline{a}, \vec{b}) \in \mathbf{b}$, every repair of $\mathbf{db}$ satisfies ${q''}_{[x,\vec{y}\mapsto a,\vec{b}]}$. By Lemma 5, the attack graph of ${q''}_{[x,\vec{y}\mapsto a,\vec{b}]}$ contains no strong cycle. From $\mathsf{incnt}({q''}_{[x,\vec{y}\mapsto a,\vec{b}]}) = \mathsf{incnt}(q') - 1 < \mathsf{incnt}(q)$, it follows that $\mathsf{CERTAINTY}({q''}_{[x,\vec{y}\mapsto a,\vec{b}]})$ is in $\mathbf{P}$ by the induction hypothesis. It follows that $\mathsf{CERTAINTY}(q)$ is in $\mathbf{P}$ as well.

Case every atom F with mode i in q' has an incoming attack in the attack graph of q'. It will be the case that no constant occurs in an atom of mode i in q'.

Then, the attack graph of q' must contain an initial strong component with two or more atoms. By Lemma 14, the Markov graph of q' contains an elementary directed cycle $\mathcal{C}$ that is premier and such that for every y in $\mathcal{C}$, $\mathsf{C}_{q'}(y) \neq \emptyset$. By Lemma 12, we can reduce in polynomial time $\mathsf{CERTAINTY}(q')$ to $\mathsf{CERTAINTY}(q^*)$ where $q^* = \mathsf{dissolve}(\mathcal{C}, q')$. Since the attack graph of q' contains no strong cycle, it follows by Lemma 13 that the attack graph of q^* contains no strong cycle either.

Let $k \geq 2$ be the size of $\mathcal{C}$. It can be easily verified that $\mathsf{incnt}(q^*) \leq (\mathsf{incnt}(q') - k) + 1 < \mathsf{incnt}(q')$. By the induction hypothesis, $\mathsf{CERTAINTY}(q^*)$ is in $\mathbf{P}$. Since there exists a polynomial-time reduction from $\mathsf{CERTAINTY}(q)$ to $\mathsf{CERTAINTY}(q^*)$, we conclude that $\mathsf{CERTAINTY}(q)$ is in $\mathbf{P}$ as well. $\square$

7.5 The Reduction of Lemma 12

This section first describes the reduction of Lemma 12, and then proves the lemma.

Relevance of subsets of repairs In Section 3, we distinguished database facts that are relevant for a query from those that are not. This notion is extended next.

Definition 7 Let q be a self-join-free Boolean conjunctive query, and let $\mathbf{db}$ be an uncertain database. A consistent subset $\mathbf{s}$ of $\mathbf{db}$ is said to be *grelevant for q in* $\mathbf{db}$ (generalized relevant) if it can be extended into a repair $\mathbf{r}$ of $\mathbf{db}$ such that some fact of $\mathbf{s}$ is relevant for q in $\mathbf{r}$. $\triangleleft$

It can be seen that $A \in \mathbf{db}$ is relevant for q in $\mathbf{db}$ if and only if $\{A\}$ is grelevant for q in $\mathbf{db}$. Therefore, "grelevant" is a notion that generalises "relevant."

Lemma 15 *Let q be a self-join-free Boolean conjunctive query, and let $\mathbf{db}$ be an uncertain database. Let $\mathbf{s}$ be a consistent subset of $\mathbf{db}$ that is not grelevant for q in $\mathbf{db}$. Let $\mathbf{db}_0 = \bigcup\{\mathsf{block}(A, \mathbf{db}) \mid A \in \mathbf{s}\}$. Then, the following are equivalent:*

1. *every repair of $\mathbf{db}$ satisfies q;*
2. *every repair of $\mathbf{db} \setminus \mathbf{db}_0$ satisfies q.*

Proof $\boxed{1 \Longrightarrow 2}$ By contraposition. Let $\mathbf{r}$ be a repair of $\mathbf{db} \setminus \mathbf{db}_0$ that falsifies q. Then, $\mathbf{r} \cup \mathbf{s}$ is a repair of $\mathbf{db}$. If $\mathbf{r} \cup \mathbf{s} \models q$, then it must be the case that $\mathbf{s}$ is grelevant for q in $\mathbf{db}$, a contradiction. We conclude by contradiction that $\mathbf{r} \cup \mathbf{s} \not\models q$. $\boxed{2 \Longrightarrow 1}$ Trivial. $\square$

Introductory example The following example illustrates the main ideas behind the reduction of Lemma 12.

Example 10 Let q be a self-join-free Boolean conjunctive query. Assume that q includes $q_0 = \{R(\underline{x}, y), S(\underline{y}, z), V(\underline{z}, x)\}$. Then, the Markov graph of q contains a cycle $x \xrightarrow{\mathsf{M}} y \xrightarrow{\mathsf{M}} z \xrightarrow{\mathsf{M}} x$. Let $\mathbf{db}$ be an uncertain database that is purified relative to q. Let $\mathbf{db}_0$ be the subset of $\mathbf{db}$ containing all R-facts, S-facts, and V-facts of $\mathbf{db}$. Assume that the following three tables represent all facts of $\mathbf{db}_0$ (for convenience, we use variables as attribute names, and we blur the distinction between a relation name R and a table representing a set of R-facts).

R: $\underline{x}$	y	S: $\underline{y}$	z	V: $\underline{z}$	x	
1	a	a	α	α	1	$\mathbf{db}_{01}$
		a	κ	κ	1	
2	b	b	β	β	2	$\mathbf{db}_{02}$
2	c	c	γ	γ	2	
3	d	d	δ	δ	3	$\mathbf{db}_{03}$
3	e	e	ϵ	ϵ	3	
4	e	e	δ	δ	4	
4	f	f	ϕ	ϕ	4	

As indicated, we can partition $\mathbf{db}_0$ into three subsets $\mathbf{db}_{01}$, $\mathbf{db}_{02}$, and $\mathbf{db}_{03}$ whose active domains have, pairwise, no constants in common. Consider each of these three subsets in turn.

1. $\mathbf{db}_{01}$ has two repairs, each of which satisfies q_0. For every repair $\mathbf{r}$ of $\mathbf{db}$, either $\mathbf{r} \models q_{0[x,y,z \mapsto 1,a,\alpha]}$ or $\mathbf{r} \models q_{0[x,y,z \mapsto 1,a,\kappa]}$.
2. $\mathbf{db}_{02}$ has two repairs, each of which satisfies q_0. For every repair $\mathbf{r}$ of $\mathbf{db}$, either $\mathbf{r} \models q_{0[x,y,z \mapsto 2,b,\beta]}$ or $\mathbf{r} \models q_{0[x,y,z \mapsto 2,c,\gamma]}$.
3. $\mathbf{db}_{03}$ has 16 repairs, and for $\mathbf{s} := \{R(\underline{3}, d), S(\underline{d}, \delta), V(\underline{\delta}, 4), R(\underline{4}, e), S(\underline{e}, \epsilon), V(\underline{\epsilon}, 3), S(\underline{f}, \phi), V(\underline{\phi}, 4)\}$, we have that $\mathbf{s}$ is a repair of $\mathbf{db}_{03}$ that falsifies q_0. It can be seen that $\mathbf{s}$ is not grelevant for q in $\mathbf{db}$. Then, by Lemma 15, every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db} \setminus \mathbf{db}_{03}$ satisfies q. That is, $\mathbf{db}_{03}$ can henceforth be ignored.

The following table T summarizes our findings. In the first column (named with a fresh variable u), the values 01 and 02 refer to $\mathbf{db}_{01}$ and $\mathbf{db}_{02}$ respectively. The table includes two blocks (separated by a dashed line for clarity). The first block indicates that for every repair $\mathbf{r}$ of $\mathbf{db}$, either $\mathbf{r} \models q_{0[x,y,z \mapsto 1,a,\alpha]}$ or $\mathbf{r} \models q_{0[x,y,z \mapsto 1,a,\kappa]}$. Likewise for the second block.

T	$\underline{u}$	x	y	z
	01	1	a	α
	01	1	a	κ
	02	2	b	β
	02	2	c	γ

The table U_x shown below is the projection of T on attributes x and u. This table must be consistent, because by construction, the active domains of $\mathbf{db}_{01}$ and $\mathbf{db}_{02}$ are disjoint. Likewise for U_y and U_z.

U_x	$\underline{x}$	u
	1	01
	2	02

U_y	$\underline{y}$	u
	a	01
	b	02
	c	02

U_z	$\underline{z}$	u
	α	01
	κ	01
	β	02
	γ	02

Let $\mathbf{db}'$ be the database that extends $\mathbf{db}$ with all the facts shown in the tables T, U_x, U_y, and U_z.[3] Let $q^* = (q \setminus q_0) \cup \{T(\underline{u}, x, y, z), U_x^c(\underline{x}, u), U_y^c(\underline{y}, u), U_z^c(\underline{z}, u)\}$. From our construction, it follows that every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}'$ satisfies q^*. $\triangleleft$

Gblocks and gpurification The following definition strengthens the notion of purification introduced earlier in Section 3.

Definition 8 Let q be a self-join-free Boolean conjunctive query such that all atoms with mode i in q are simple-key. Let $\mathbf{db}$ be an uncertain database that is purified and typed relative to q. A *gblock* (generalized block) of $\mathbf{db}$ relative to q is a maximal (with respect to $\subseteq$) subset $\mathbf{g}$ of $\mathbf{db}$ such that all facts in $\mathbf{g}$ have mode i and agree on their primary-key position (but may disagree on their relation name). Notice that a gblock has at most polynomially many repairs (in the size of $\mathbf{db}$).[4] We say that $\mathbf{db}$ is *gpurified relative to* q if for every gblock $\mathbf{g}$ of $\mathbf{db}$, every repair of $\mathbf{g}$ is grelevant for q in $\mathbf{db}$. $\triangleleft$

Clearly, every gblock is the union of one or more blocks. Two facts of the same gblock have the same primary-key value, but can have distinct relation names.

Example 11 Let $q = \{R(\underline{x}, y), S(\underline{x}, y)\}$. Let $\mathbf{db} = \{R(\underline{a}, 1), R(\underline{a}, 2), S(\underline{a}, 1), S(\underline{a}, 2)\}$. Then, $\mathbf{db}$ is purified and typed relative to q. All facts of $\mathbf{db}$ together constitute a gblock. The uncertain database $\mathbf{db}$ is not gpurified, since $\mathbf{s} = \{R(\underline{a}, 1), S(\underline{a}, 2)\}$ is a repair of the gblock, and also a repair of $\mathbf{db}$. However, neither $R(\underline{a}, 1)$ nor $S(\underline{a}, 2)\}$ is relevant for q in $\mathbf{s}$. $\triangleleft$

Example 12 Let $q = \{R_1(\underline{x}, y), R_2(\underline{x}, z), S(\underline{y, z})\}$, where the signature of S is $[2, 2]$. Let $\mathbf{db}$ be the uncertain database containing the following facts.

R_1	$\underline{x}$	y
	a	1
	a	2

R_2	$\underline{x}$	z
	a	3
	a	4

S	$\underline{y}$	$\underline{z}$
	1	3
	2	4

Then, $\mathbf{db}$ is purified and typed relative to q. All R_1-facts and R_2-facts together constitute a gblock. A repair of this gblock is $\mathbf{s} = \{R_1(\underline{a}, 1), R_2(\underline{a}, 4)\}$. The uncertain database $\mathbf{db}$ is not gpurified. Indeed, the only repair of $\mathbf{db}$ that extends $\mathbf{s}$ is $\{R_1(\underline{a}, 1), R_2(\underline{a}, 4), S(\underline{1, 3}), S(\underline{2, 4})\}$ (call it $\mathbf{r}$). Neither $R_1(\underline{a}, 1)$ nor $R_2(\underline{a}, 4)$ is relevant for q in $\mathbf{r}$. $\triangleleft$

[3] Facts of $\mathbf{db}_0$ can be omitted from $\mathbf{db}'$, but that is not important.

[4] Indeed, since $\mathbf{db}$ is purified relative to q, every gblock of $\mathbf{db}$ contains at most $|q|$ distinct relation names, and hence has at most $|\mathbf{db}|^{|q|}$ distinct repairs.

The following lemma is similar to Lemma 1 and has an easy proof.

Lemma 16 *Let q be a self-join-free Boolean conjunctive query such that all atoms with mode i in q are simple-key. Let $\mathbf{db}$ be an uncertain database that is purified and typed relative to q. It is possible to compute in polynomial time an uncertain database $\mathbf{db}'$ that is gpurified relative to q such that every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}'$ satisfies q.*

Specification of the reduction of Lemma 12 Let q and $\mathcal{C}$ be as in the statement of Lemma 12. Assume that the elementary directed cycle $\mathcal{C}$ in the Markov graph of q is $x_0 \xrightarrow{\mathsf{M}} x_1 \cdots \xrightarrow{\mathsf{M}} x_{k-1} \xrightarrow{\mathsf{M}} x_0$. In what follows, let $\mathsf{dissolve}(\mathcal{C}, q)$ be as in Definition 6, with q_0, q_1, $\vec{y}$, u, T, and $U_0, \dots, U_{k-1}$ as defined there. Moreover, we write $\oplus$ for addition modulo k, and $\ominus$ for subtraction modulo k. For every $i \in \{0, \dots, k-1\}$, we define X_i as follows:

$$X_i := \mathsf{vars}(\mathsf{C}_q(x_i)).$$

The reduction of Lemma 12 will be described under the following simplifying assumptions which can be made without loss of generality:

- every uncertain database $\mathbf{db}$ that is input to $\mathsf{CERTAINTY}(q)$ is typed, purified, and gpurified relative to q. This assumption is without loss of generality as argued in Section 3, and by Lemmas 1 and 16; and
- for every $i \in \{0, \dots, k-1\}$, no atom of $\mathsf{C}_q(x_i)$ contains constants or double occurrences of the same variable. This assumption is without loss of generality by Lemma 11.

Under these notations and assumptions, we describe the reduction of Lemma 12. Let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$. Define a directed k-partite graph, denoted $\mathcal{G}(\mathbf{db})$, as follows:

1. the vertex set of $\mathcal{G}(\mathbf{db})$ is $\bigcup_{i=0}^{k-1} \mathsf{type}(x_i)$; and
2. there is a directed edge from $a \in \mathsf{type}(x_i)$ to $b \in \mathsf{type}(x_{i\oplus 1})$ if for some valuation θ over $\mathsf{vars}(q)$, we have that $\theta(q) \subseteq \mathbf{db}$ and $\theta(x_i) = a$ and $\theta(x_{i\oplus 1}) = b$. In this case, we say that $\theta[X_i]$ *realizes* the edge (a, b), where $\theta[X_i]$ denotes the restriction of θ on X_i.

Notice that distinct valuations can realize the same edge of $\mathcal{G}(\mathbf{db})$ (but if $\mathbf{db}$ is consistent, then every edge in $\mathcal{G}(\mathbf{db})$ is realized at most once).

Example 13 Let $q = \{R_1(\underline{x_0}, y_1),\ R_2(\underline{x_0}, y_2),\ S^c(\underline{y_1, y_2}, x_1),\ R_3(\underline{x_0}, y_3),\ V(\underline{x_1}, x_0)\}$. Then, $x_0 \xrightarrow{\mathsf{M}} x_1$ and $X_0 = \{x_0, y_1, y_2, y_3\}$. Assume an uncertain database $\mathbf{db}$ containing, among others, the following facts.

R_1	$\underline{x_0}$	y_1
	a	c_1

R_2	$\underline{x_0}$	y_2
	a	c_2
	a	c_3

S	$\underline{y_1}$	$\underline{y_2}$	x_1
	c_1	c_2	1
	c_1	c_3	1

R_3	$\underline{x_0}$	y_3
	a	β
	a	γ

The graph $\mathcal{G}(\mathbf{db})$ contains a directed edge $(a, 1)$, which is realized by $\{x_0 \mapsto a,\ y_1 \mapsto c_1,\ y_2 \mapsto c_2,\ y_3 \mapsto \beta\}$. The edge $(a, 1)$ is also realized by $\{x_0 \mapsto a,\ y_1 \mapsto c_1,\ y_2 \mapsto c_3,\ y_3 \mapsto \gamma\}$. $\lhd$

Let $[\![\mathbf{db}]\!]$ be the subset of $\mathbf{db}$ that contains all facts with mode c. Significantly, the edges in $\mathcal{G}(\mathbf{db})$ outgoing from some constant $a \in \mathsf{type}(x_j)$ (for some $j \in \{0, \dots, k-1\}$) are fully determined by $[\![\mathbf{db}]\!]$ and the gblock of $\mathbf{db}$ containing all facts whose relation name is in $\mathsf{C}_q(x_j)$ and whose primary-key position contains the constant a (call this gblock $\mathbf{g}_a$). Since $\mathbf{db}$ is gpurified, for every repair $\mathbf{s}$ of $\mathbf{g}_a$, there exists a unique constant $b \in \mathsf{type}(x_{j\oplus 1})$ such that

$$\mathbf{s} \cup [\![\mathbf{db}]\!] \models (\mathsf{C}_q(x_j) \cup [\![q]\!])_{[x_j, x_{j\oplus 1} \mapsto a, b]},$$

in which case $\mathcal{G}(\mathbf{db})$ will contain a directed edge from a to b. Uniqueness of b follows from $\mathcal{K}(\mathsf{C}_q(x_j) \cup [\![q]\!]) \models x_j \rightarrow x_{j\oplus 1}$ and [16, Lemma 4.3].

Since $\mathbf{db}$ is gpurified, $\mathcal{G}(\mathbf{db})$ is a vertex-disjoint union of strong components such that no edge leads from one strong component to another strong component (i.e., all strong components are initial).[5] In what follows, let D be a strong component of $\mathcal{G}(\mathbf{db})$. Since $\mathcal{G}(\mathbf{db})$ is k-partite, the length of any cycle in $\mathcal{G}(\mathbf{db})$ must be a multiple of k, i.e., must be in $\{k, 2k, 3k, \dots\}$. Let $\mathbf{db}_D$ be the subset of $\mathbf{db}$ that contains $R(\underline{a}, \vec{b})$ whenever R is of mode i

[5] Strong components are defined by Definition 1.

and the constant a is a vertex in D (and $\vec{b}$ is any sequence of constants). Obviously, every block of $\mathbf{db}$ is either included in $\mathbf{db}_D$ or disjoint with $\mathbf{db}_D$.

Clearly, D must contain a cycle. Among the cycles in D of length exactly k, we now distinguish the cycles that *support* q from those that do not, as defined next. Let such cycle in D be

$$a_0, a_1, \dots, a_{k-1}, a_0 \tag{5}$$

where for $i \in \{0, \dots, k-1\}$, $a_i \in \mathsf{type}(x_i)$. For $i \in \{0, \dots, k-1\}$, let Δ_i be the set of all valuations over X_i that realize $(a_i, a_{i\oplus 1})$. We say that the cycle (5) *supports* q if for for all $i, j \in \{0, \dots, k-1\}$, for all $\mu_i \in \Delta_i$ and $\mu_j \in \Delta_j$, it is the case that μ_i and μ_j agree on all variables in $X_i \cap X_j$. Notice that $X_i \cap X_j$ can be empty. The cycle (5) may not support q, because μ_i and μ_j can disagree on variables in $X_i \cap X_j \cap \mathsf{vars}(\vec{y})$, as illustrated next.

Example 14 Let $q = \{R(\underline{x_0}, x_1, y), S(\underline{x_1}, x_0, y)\}$. We have $x_0 \xrightarrow{\mathsf{M}} x_1 \xrightarrow{\mathsf{M}} x_0$. Let $\mathbf{db}$ be the uncertain database containing the following facts.

R	$\underline{x_0}$	x_1	y
	a	1	α
	a	1	β

S	$\underline{x_1}$	x_0	y
	1	a	α
	1	a	β

The edge set of $\mathcal{G}(\mathbf{db})$ is $\{(a, 1), (1, a)\}$. Both $(a, 1)$ and $(1, a)$ are realized by the valuations $\{x_0 \mapsto a, x_1 \mapsto 1, y \mapsto \alpha\}$ and $\{x_0 \mapsto a, x_1 \mapsto 1, y \mapsto \beta\}$, which disagree on y. Hence, the cycle $a, 1, a$ does not support q. $\triangleleft$

On the other hand, we can assume without loss of generality that μ_i and μ_j agree on all variables in $X_i \cap X_j \cap \{x_0, \dots, x_{k-1}\}$. In particular, if $x_i \in X_j$, then $\mu_j(x_i) = \mu_i(x_i) = a_i$. To see why this is the case, assume that $x_i \in X_j$, where $i, j \in \{0, \dots, k-1\}$ and $i \neq j$. Then, it must be that $x_j \xrightarrow{\mathsf{M}} x_i$. Two cases can occur:

- if $j = i \ominus 1$, then μ_j realizes the edge $(a_{i\ominus 1}, a_i)$ and $\mu_j(x_i) = a_i$; and
- if $j \neq i \ominus 1$, then $x_j \xrightarrow{\mathsf{M}} x_i \xrightarrow{\mathsf{M}} x_{i\oplus 1} \cdots \xrightarrow{\mathsf{M}} x_{j\ominus 1} \xrightarrow{\mathsf{M}} x_j$ is a shorter Markov cycle.

The second case can be avoided by picking $\mathcal{C}$ to be the shorter cycle, as illustrated by Example 15. It can be seen that such choice of $\mathcal{C}$ is without loss of generality. In particular, in Lemma 14, if $\mathcal{C}$ was premier, then the shorter cycle will also be premier.

Example 15 Let $q = \{R(\underline{x_0}, x_1), S(\underline{x_1}, x_2, x_0), V(\underline{x_2}, x_0)\}$. Then, $x_0 \xrightarrow{\mathsf{M}} x_1 \xrightarrow{\mathsf{M}} x_2 \xrightarrow{\mathsf{M}} x_0$. We have $X_0 = \{x_0, x_1\}$, $X_1 = \{x_1, x_2, x_0\}$, and $X_2 = \{x_2, x_0\}$. Assume an uncertain database $\mathbf{db}$ with the following facts.

R	$\underline{x_0}$	x_1
	a	1
	b	1

S	$\underline{x_1}$	x_2	x_0
	1	β	a
	1	β	b

V	$\underline{x_2}$	x_0
	β	a
	β	b

The graph $\mathcal{G}(\mathbf{db})$ contains an elementary directed cycle $a, 1, \beta, a$. The edge $(a, 1)$ is realized by $\mu_0 = \{x_0 \mapsto a, x_1 \mapsto 1\}$. The edge $(1, \beta)$ is realized, among others, by $\mu_1 = \{x_1 \mapsto 1, x_2 \mapsto \beta, x_0 \mapsto b\}$. Notice that μ_0 and μ_1 disagree on x_0. Although it is easy to deal with this situation where two valuations disagree on a variable in the Markov cycle, it is even easier to avoid this situation by working with the shorter Markov cycle $x_0 \xrightarrow{\mathsf{M}} x_1 \xrightarrow{\mathsf{M}} x_0$. $\triangleleft$

We now distinguish two cases.

Case D contains either an elementary directed cycle of size k that does not support q, or an elementary directed cycle of size strictly greater than k. We show in the next paragraph how to construct a repair $\mathbf{s}$ of $\mathbf{db}_D$ such that $\mathbf{s}$ is not grelevant for q in $\mathbf{db}$. Then, by Lemma 15, every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db} \setminus \mathbf{db}_D$ satisfies q. In this case, the reduction deletes from $\mathbf{db}$ all facts of $\mathbf{db}_D$.

The construction of $\mathbf{s}$ proceeds as follows. Pick an elementary cycle in D that has size strictly greater than k, or that has size k but does not support q. The cycle picked will henceforth be denoted by $\mathcal{E}$. Construct a maximal sequence

$$(V_0, E_0), b_1, (V_1, E_1), b_2, (V_2, E_2), \dots, b_n, (V_n, E_n)$$

where

1. V_0 is the set of vertices in $\mathcal{E}$, and E_0 is the set of directed edges in $\mathcal{E}$; and
2. for every $i \in \{1, \dots, n\}$,
   (a) $b_i \notin V_{i-1}$ and for some $c \in V_{i-1}$, (b_i, c) is a directed edge in $\mathcal{G}(\mathbf{db})$; and
   (b) $V_i = V_{i-1} \cup \{b_i\}$ and $E_i = E_{i-1} \cup \{(b_i, c)\}$.

The resulting graph (V_n, E_n) is such that V_n is equal to the vertex set of D, and E_n contains exactly one outgoing edge for each vertex in V_n. The graph (V_n, E_n) contains no directed cycle other than $\mathcal{E}$. To construct $\mathbf{s}$, for each $j \in \{0, \dots, k-1\}$, for each vertex $a \in V_n \cap \mathsf{type}(x_j)$, select some valuation μ that realizes the edge in E_n outgoing from a, and add $\mu(\mathsf{C}_q(x_j))$ to $\mathbf{s}$. If $\mathcal{E}$ has size k, then the valuations μ should be selected such that for some vertices a, b in $\mathcal{E}$, the valuations chosen for a and b disagree on some variable of $\mathsf{vars}(\vec{y})$. It is not hard to see that the set $\mathbf{s}$ so obtained is a repair of $\mathbf{db}_D$ that is not grelevant for q in $\mathbf{db}$.

We illustrate the above construction by two examples.

Example 16 In Example 14, one can choose $\mathbf{s} = \{R(\underline{a}, 1, \alpha),\ S(\underline{1}, a, \beta)\}$. The treatment of a directed cycle of size strictly greater than k is illustrated by $\mathbf{db}_{03}$ in Example 10. $\triangleleft$

Example 17 Let $q = \{R(\underline{x_0}, y_1, y_2),\ V(\underline{x_1}, y_2),\ S_1^c(\underline{y_1}, \underline{y_2}, x_1),\ S_2^c(\underline{y_2}, x_0)\}$. We have $x_0 \xrightarrow{\mathsf{M}} x_1 \xrightarrow{\mathsf{M}} x_0$, $X_0 = \{x_0, y_1, y_2\}$, and $X_1 = \{x_1, y_2\}$. Let $\mathbf{db}$ be an uncertain database with the following facts.

R	$\underline{x_0}$	y_1	y_2
	a	1	2
	a	3	4
	a	1	6

V	$\underline{x_1}$	y_2
	γ	2
	γ	4
	β	6

S_1^c	$\underline{y_1}$	$\underline{y_2}$	x_1
	1	2	γ
	3	4	γ
	1	6	β

S_2^c	$\underline{y_2}$	x_0
	2	a
	4	a
	6	a

The following table lists the edges in $\mathcal{G}(\mathbf{db})$, by type, along with the valuations that realize each edge.

Edges in $\mathsf{type}(x_0) \times \mathsf{type}(x_1)$

Edge	Realized by
(a, γ)	$\{x_0 \mapsto a, y_1 \mapsto 1, y_2 \mapsto 2\} = \mu_1$
	$\{x_0 \mapsto a, y_1 \mapsto 3, y_2 \mapsto 4\} = \mu_2$
(a, β)	$\{x_0 \mapsto a, y_1 \mapsto 1, y_2 \mapsto 6\} = \mu_3$

Edges in $\mathsf{type}(x_1) \times \mathsf{type}(x_0)$

Edge	Realized by
(γ, a)	$\{x_1 \mapsto \gamma, y_2 \mapsto 2\} = \mu_4$
	$\{x_1 \mapsto \gamma, y_2 \mapsto 4\} = \mu_5$
(β, a)	$\{x_1 \mapsto \beta, y_2 \mapsto 6\} = \mu_6$

Then, $\mathcal{G}(\mathbf{db})$ contains two elementary cycles, a, γ, a and a, β, a, both of length 2. The cycle a, β, a supports q. The cycle a, γ, a does not support q, because μ_1 and μ_5 disagree on y_2. Therefore, the edges (a, γ) and (γ, a), along with μ_1 and μ_5, will be used in the construction of a consistent set $\mathbf{s}$ that is not grelevant for q in $\mathbf{db}$. For the remaining vertex β, we add the edge (β, a), which is only realized by μ_6. Then, $\mathbf{s}$ contains the R-fact $R(\underline{a}, 1, 2)$ (because of μ_1), and the V-facts $V(\underline{\gamma}, 4)$ and $V(\underline{\beta}, 6)$ (because of μ_5 and μ_6 respectively). In this example, there is only one repair that contains $\mathbf{s}$, and this repair falsifies q. $\triangleleft$

Case every elementary directed cycle in D has length k and supports q. In this case, we will encode each cycle of D as a set of T-facts, as follows. Consider any cycle of the form (5) in D, and take the cross product

$$\Delta_0 \times \Delta_2 \times \cdots \times \Delta_{k-1}, \tag{6}$$

which is of polynomial size (in the size of $\mathbf{db}$). Since we are in the case where any cycle of the form (5) supports q, for every tuple $(\mu_0, \mu_1, \dots, \mu_{k-1})$ in the cross product (6), the set $\mu := \bigcup_{i=0}^{k-1} \mu_i$ is a well defined valuation over $\{x_0, \dots, x_{k-1}\} \cup \mathsf{vars}(\vec{y})$. In this case, for each such tuple, the reduction adds the following $k+1$ facts:

$$\begin{array}{l} T(\underline{D}, a_0, \dots, a_{k-1}, \mu(\vec{y})) \\ U_0^c(\underline{a_0}, D) \\ \quad\vdots \\ U_{k-1}^c(\underline{a_{k-1}}, D) \end{array}$$

in which D is used as a constant. Recall that $a_i = \mu(x_i)$ for $i \in \{0, \dots, k-1\}$. Notice that if the sequence $\vec{y}$ is empty, then the reduction will add exactly one T-fact for every cycle of the form (5). Otherwise, the reduction may add multiple T-facts for the same cycle, as illustrated next.

Example 18 Let $q = \{R(\underline{x_0}, x_1, y),\ S(\underline{x_1}, x_0)\}$. We have $x_0 \xrightarrow{\mathsf{M}} x_1 \xrightarrow{\mathsf{M}} x_0$, $X_0 = \{x_0, x_1, y\}$ and $X_1 = \{x_0, x_1\}$. Let $\mathbf{db}$ be the uncertain database containing the following facts.

R	$\underline{x_0}$	x_1	y
	a	1	α
	a	1	β

S	$\underline{x_1}$	x_0
	1	a

The edge set of $\mathcal{G}(\mathbf{db})$ is $\{(a, 1),\ (1, a)\}$. The edge $(a, 1)$ is realized by both $\{x_0 \mapsto a,\ x_1 \mapsto 1,\ y \mapsto \alpha\}$ and $\{x_0 \mapsto a,\ x_1 \mapsto 1,\ y \mapsto \beta\}$. The edge $(1, a)$ is realized only by $\{x_0 \mapsto a,\ x_1 \mapsto 1\}$. The cycle $a, 1, a$ in $\mathcal{G}(\mathbf{db})$ supports q. The reduction will add the following T-facts (for some identifier D):

T	$\underline{u}$	x_0	x_1	y
	D	a	1	α
	D	a	1	β

$\triangleleft$

Example 19 Take the query q of Example 17, with the following uncertain database $\mathbf{db}$.

R	$\underline{x_0}$	y_1	y_2
	a	1	2
	a	1	6
	a	3	6

V	$\underline{x_1}$	y_2
	γ	2
	β	6

S_1^c	$\underline{y_1}$	$\underline{y_2}$	x_1
	1	2	γ
	1	6	β
	3	6	β

S_2^c	$\underline{y_2}$	x_0
	2	a
	6	a

Then, $\mathcal{G}(\mathbf{db})$ contains two elementary cycles, a, γ, a and a, β, a, both of length 2 and both supporting q. The reduction will add the following T-facts (for some identifier D):

T	$\underline{u}$	x_0	x_1	y_1	y_2
	D	a	γ	1	2
	D	a	β	1	6
	D	a	β	3	6

$\triangleleft$

Each relation U_i^c encodes that each constant in $\mathsf{type}(x_i) \cap \mathbf{adom}(\mathbf{db})$ occurs in a unique strong component of $\mathcal{G}(\mathbf{db})$. The meaning of the T-facts is as follows. Let $V = \{x_0, \dots, x_{k-1}\} \cup \mathsf{vars}(\vec{y})$. Let Θ_D be the set of all valuations over V such that

$$T(\underline{D}, \mu(x_1), \dots, \mu(x_{k-1}), \mu(\vec{y}))$$

has been added by the reduction. Then the following hold (recall $q_0 = \bigcup_{i=0}^{k-1} \mathsf{C}_q(x_i)$):

- for every repair $\mathbf{r}$ of $\mathbf{db}$, there exists $\mu \in \Theta_D$ such that $\mathbf{r} \models \mu(q_0)$; and
- for every $\mu \in \Theta_D$, there exists a repair $\mathbf{r}$ of $\mathbf{db}$ such that
  1. $\mathbf{r} \models \mu(q_0)$; and
  2. for each $\mu' \in \Theta_D$, if $\mu' \neq \mu$, then $\mathbf{r} \not\models \mu'(q_0)$.

The cycles in D can be found in polynomial time by solving reachability problems, as explained in [17, Theorem 4] and [11]. The crux is that the number of cycles in $\mathcal{G}(\mathbf{db})$ of length exactly k is polynomially bounded. Any longer cycle consists of an elementary path $a_0, a_1, \dots, a_{k-1}, a'_0$ of length k ($a_0 \neq a'_0$), concatenated with an elementary path from a'_0 to a_0 that contains no vertex in $\{a_1, \dots, a_{k-1}\}$. Notice incidentally that the reduction needs to know the existence (or not) of cycles of size strictly greater than k in any strong component D, but the vertices on such cycle need not be remembered.

It can now be seen that, in general, the above reduction results in a database $\mathbf{db}'$ that is as in the following lemma.

Lemma 17 *Let q and $\mathcal{C}$ be as in the statement of Lemma 12. Let $q^* = \mathsf{dissolve}(q, \mathcal{C})$, and let the variable u be as in Definition 6. Let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$. We can compute in polynomial time an uncertain database $\mathbf{db}'$ that is a legal input to $\mathsf{CERTAINTY}(q^*)$ such that the following hold:*

1. *for every repair* $\mathbf{r}$ *of* $\mathbf{db}$*, there exists a repair* $\mathbf{r}'$ *of* $\mathbf{db}'$ *such that for every valuation* θ *over* $\mathsf{vars}(q^*)$*, if* $\theta(q^*) \subseteq \mathbf{r}'$*, then* $\theta(q) \subseteq \mathbf{r}$*; and*
2. *for every repair* $\mathbf{r}'$ *of* $\mathbf{db}'$*, there exists a repair* $\mathbf{r}$ *of* $\mathbf{db}$ *such that for every valuation* θ *over* $\mathsf{vars}(q)$*, if* $\theta(q) \subseteq \mathbf{r}$*, then there exists a constant* D *such that* $\theta_{[u \mapsto D]}(q^*) \subseteq \mathbf{r}'$*.*

We can now prove Lemma 12.

Proof of Lemma 12 Let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$. By Lemma 17, we can compute in polynomial time an uncertain database $\mathbf{db}'$ that is a legal input to $\mathsf{CERTAINTY}(q^*)$ such that $\mathbf{db}'$ satisfies conditions 1 and 2 in the statement of Lemma 17. It suffices to show that the following are equivalent.

1. Every repair of $\mathbf{db}$ satisfies q.
2. Every repair of $\mathbf{db}'$ satisfies q^*.

$\boxed{1 \Longrightarrow 2}$ Proof by contraposition. Assume a repair $\mathbf{r}'$ of $\mathbf{db}'$ such that $\mathbf{r}' \not\models q^*$. By item 2 in the statement of Lemma 17, we can assume a repair $\mathbf{r}$ of $\mathbf{db}$ such that for every valuation θ over $\mathsf{vars}(q)$, if $\theta(q) \subseteq \mathbf{r}$, then there exists a constant D such that $\theta_{[u \mapsto D]}(q^*) \subseteq \mathbf{r}'$. Obviously, if $\mathbf{r} \models q$, then $\mathbf{r}' \models q^*$, a contradiction. We conclude by contradiction that $\mathbf{r} \not\models q$. $\boxed{2 \Longrightarrow 1}$ Proof by contraposition. Assume a repair $\mathbf{r}$ of $\mathbf{db}$ such that $\mathbf{r} \not\models q$. By item 1 in the statement of Lemma 17, we can assume a repair $\mathbf{r}'$ of $\mathbf{db}'$ such that for every valuation θ over $\mathsf{vars}(q^*)$, if $\theta(q^*) \subseteq \mathbf{r}'$, then $\theta(q) \subseteq \mathbf{r}$. Obviously, $\mathbf{r}' \not\models q^*$. □

8 Conclusion

This paper settles a long-standing open question in certain query answering, by establishing an effective complexity trichotomy in the set containing $\mathsf{CERTAINTY}(q)$ for each self-join-free Boolean conjunctive query q. In particular, we show that, given q, there exists a procedure that looks at the structure of the attack graph of q and decides whether $\mathsf{CERTAINTY}(q)$ is in $\mathbf{FO}$, in $\mathbf{P} \setminus \mathbf{FO}$, or $\mathbf{coNP}$-complete.

The exciting question that still remains open is whether the above trichotomy can be extended beyond self-join-free conjunctive queries, to conjunctive queries with self-joins and unions of conjunctive queries.

References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *PODS*, pages 68–79. ACM Press, 1999.

[3] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979.

[4] Catriel Beeri, Ronald Fagin, David Maier, and Mihalis Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30(3):479–513, 1983.

[5] Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[6] Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24, 2011.

[7] Stephen A. Cook and Pierre McKenzie. Problems complete for deterministic logarithmic space. *J. Algorithms*, 8(3):385–394, 1987.

[8] Gaëlle Fontaine. Why is it hard to obtain a dichotomy for consistent query answering? In *LICS*, pages 550–559. IEEE Computer Society, 2013.

[9] Ariel Fuxman and Renée J. Miller. First-order query rewriting for inconsistent databases. In Thomas Eiter and Leonid Libkin, editors, *ICDT*, volume 3363 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2005.

[10] Phokion G. Kolaitis and Enela Pema. A dichotomy in the complexity of consistent query answering for queries with two atoms. *Inf. Process. Lett.*, 112(3):77–85, 2012.

[11] Paraschos Koutris and Dan Suciu. A dichotomy on the complexity of consistent query answering for atoms with simple keys. In Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy, editors, *ICDT*, pages 165–176. OpenProceedings.org, 2014.

[12] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[13] George J. Minty. On maximal independent sets of vertices in claw-free graphs. *J. Comb. Theory, Ser. B*, 28(3):284–304, 1980.

[14] Jef Wijsen. On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In Jan Paredaens and Dirk Van Gucht, editors, *PODS*, pages 179–190. ACM, 2010.

[15] Jef Wijsen. A remark on the complexity of consistent conjunctive query answering under primary key violations. *Inf. Process. Lett.*, 110(21):950–955, 2010.

[16] Jef Wijsen. Certain conjunctive query answering in first-order logic. *ACM Trans. Database Syst.*, 37(2):9, 2012.

[17] Jef Wijsen. Charting the tractability frontier of certain conjunctive query answering. In Richard Hull and Wenfei Fan, editors, *PODS*, pages 189–200. ACM, 2013.

[18] Jef Wijsen. A survey of the data complexity of consistent query answering under key constraints. In Christoph Beierle and Carlo Meghini, editors, *Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings*, volume 8367 of *Lecture Notes in Computer Science*, pages 62–78. Springer, 2014.

A Proofs for Section 4

A.1 Proof of Lemma 3

We use the following helping lemma.

Lemma 18 *Let q be a self-join-free Boolean conjunctive query. Let $F, G \in q$ such that $F \overset{q}{\rightsquigarrow} G$. Then, for every $x \in F^{+,q} \setminus G^{+,q}$, there exists a sequence $F_0, F_1, \ldots, F_n$ of atoms of q such that*

- $F_0 = F$;
- *for all* $i \in \{0, \ldots, n-1\}$, $\mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1}) \nsubseteq G^{+,q}$; *and*
- $x \in \mathsf{vars}(F_n)$.

Proof Consider a maximal sequence

$$\begin{array}{rcll} \mathsf{key}(F) & = & S_0 & H_1 \\ & & S_1 & H_2 \\ & & \vdots & \vdots \\ & & S_{k-1} & H_k \\ & & S_k & \end{array}$$

where

1. $S_0 \subsetneq S_1 \subsetneq \cdots \subsetneq S_{k-1} \subsetneq S_k$; and
2. for every $i \in \{1, 2, \ldots, k\}$,
   (a) $H_i \in q \setminus \{F\}$. Thus, $\mathcal{K}(q \setminus \{F\})$ contains the functional dependency $\mathsf{key}(H_i) \rightarrow \mathsf{vars}(H_i)$.

(b) $\mathsf{key}(H_i) \subseteq S_{i-1}$ and $S_i = S_{i-1} \cup \mathsf{vars}(H_i)$.

Then, $S_k = F^{+,q}$. From $F \overset{q}{\rightsquigarrow} G$, it follows $G \notin \{H_1, \ldots, H_k\}$. For every $v \in S_k$, define $d(v)$ as the smallest integer i such that $v \in S_i$. Let $x \in F^{+,q} \setminus G^{+,q}$. We define the desired result by induction on $d(x)$.

Basis: $d(x) = 0$. Then the desired sequence is F.

Step: $d(x) = i$. Hence, $x \in S_i$ and $x \notin S_{i-1}$. Then, $x \notin \mathsf{key}(H_i) \subseteq S_{i-1}$ and $x \in \mathsf{vars}(H_i)$. Since $H_i \neq G$, we have $\mathsf{key}(H_i) \nsubseteq G^{+,q}$, or else $x \in G^{+,q}$, a contradiction. Therefore, we can assume some variable $y \in \mathsf{key}(H_i) \setminus G^{+,q}$. Since $y \in S_{i-1}$, we have $d(y) < d(x)$. By the induction hypothesis, there exists a sequence $F_0, F_1, \ldots, F_n$ of atoms of q such that

- $F_0 = F$;
- for all $i \in \{0, \ldots, n-1\}$, $\mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1}) \nsubseteq G^{+,q}$; and
- $y \in F_n$.

The desired sequence is $F_0, F_1, \ldots, F_n, H_i$. □

The proof of Lemma 3 is given next.

Proof of Lemma 3 Assume $F \overset{q}{\rightsquigarrow} G$, $G \overset{q}{\rightsquigarrow} H$, and $F \overset{q}{\not\rightsquigarrow} H$.

Since $F \overset{q}{\rightsquigarrow} G$, there exists a sequence $F_0, F_1, \ldots, F_n$ of atoms of q such that

- $F_0 = F$ and $F_n = G$; and
- for all $i \in \{0, \ldots, n-1\}$, $\mathsf{vars}(F_i) \cap \mathsf{vars}(F_{i+1}) \nsubseteq F^{+,q}$.

Since $G \overset{q}{\rightsquigarrow} H$, there exists a sequence $G_0, G_1, \ldots, G_m$ of atoms of q such that

- $G_0 = G$ and $G_m = H$; and
- for all $i \in \{0, \ldots, m-1\}$, $\mathsf{vars}(G_i) \cap \mathsf{vars}(G_{i+1}) \nsubseteq G^{+,q}$.

Consider the path

$$F_0, F_1, \ldots, F_n, G_1, G_2, \ldots, G_m$$

where $F_0 = F$, $F_n = G = G_0$, and $G_m = H$. Since $F \overset{q}{\not\rightsquigarrow} H$, we can assume $j \in \{0, \ldots, m-1\}$ such that $\mathsf{vars}(G_j) \cap \mathsf{vars}(G_{j+1}) \subseteq F^{+,q}$. Since $\mathsf{vars}(G_j) \cap \mathsf{vars}(G_{j+1}) \nsubseteq G^{+,q}$, we can assume $x \in \mathsf{vars}(G_j) \cap \mathsf{vars}(G_{j+1})$ such that $x \in F^{+,q} \setminus G^{+,q}$.

By Lemma 18, there exists a sequence $H_0, H_1, \ldots, H_k$ of atoms of q such that

- $H_0 = F$;
- for all $i \in \{0, \ldots, k-1\}$, $\mathsf{vars}(H_i) \cap \mathsf{vars}(H_{i+1}) \nsubseteq G^{+,q}$; and
- $x \in H_k$.

Consider the sequence

$$G_0, G_1, \ldots, G_j, H_k, H_{k-1}, \ldots, H_0,$$

where $G_0 = G$ and $H_0 = F$. Every two consecutive atoms in this sequence share a variable not in $G^{+,q}$. In particular, G_j and H_k share the variable x. It follows $G \overset{q}{\rightsquigarrow} F$. □

A.2 Proof of Lemma 4

Proof of Lemma 4 The first item is an immediate consequence of Lemma 3. In what follows, we show the second item.

We show that if the attack graph of q contains a strong cycle of length n with $n \geq 3$, then it contains a strong cycle of some length m with $m < n$.

Let $H_0 \overset{q}{\rightsquigarrow} H_1 \overset{q}{\rightsquigarrow} H_2 \overset{q}{\rightsquigarrow} \cdots \overset{q}{\rightsquigarrow} H_{n-1} \overset{q}{\rightsquigarrow} H_0$ be a strong cycle of length n ($n \geq 3$) in the attack graph of q, where $i \neq j$ implies $H_i \neq H_j$. Assume without loss of generality that the attack $H_0 \overset{q}{\rightsquigarrow} H_1$ is strong. Thus, $\mathcal{K}(q) \not\models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$.

We write $i \oplus j$ as shorthand for for $(i+j) \mod n$. If $H_1 \overset{q}{\rightsquigarrow} H_{1\oplus 2}$, then $H_0 \overset{q}{\rightsquigarrow} H_1 \overset{q}{\rightsquigarrow} H_{1\oplus 2} \overset{q}{\rightsquigarrow} \cdots \overset{q}{\rightsquigarrow} H_{n-1} \overset{q}{\rightsquigarrow} H_0$ is a strong cycle of length $n-1$, and the desired result holds. Assume next $H_1 \overset{q}{\not\rightsquigarrow} H_{1\oplus 2}$. By Lemma 3, $H_2 \overset{q}{\rightsquigarrow} H_1$. We distinguish two cases.

Case $H_2 \overset{q}{\rightsquigarrow} H_1$ is a strong attack. Then $H_1 \overset{q}{\rightsquigarrow} H_2 \overset{q}{\rightsquigarrow} H_1$ is a strong cycle of length $2 < n$.

Case $H_2 \overset{q}{\rightsquigarrow} H_1$ is a weak attack. If $H_1 \overset{q}{\rightsquigarrow} H_0$, then $H_0 \overset{q}{\rightsquigarrow} H_1 \overset{q}{\rightsquigarrow} H_0$ is a strong cycle of length $2 < n$. Assume next $H_1 \overset{q}{\not\rightsquigarrow} H_0$. Then, from $H_0 \overset{q}{\rightsquigarrow} H_1 \overset{q}{\rightsquigarrow} H_2$ and Lemma 3, it follows $H_0 \overset{q}{\rightsquigarrow} H_2$. The cycle $H_0 \overset{q}{\rightsquigarrow} H_2 \overset{q}{\rightsquigarrow} H_{2\oplus 1} \overset{q}{\rightsquigarrow} \cdots \overset{q}{\rightsquigarrow} H_{n-1} \overset{q}{\rightsquigarrow} H_0$ has length $n-1$. It suffices to show that the attack $H_0 \overset{q}{\rightsquigarrow} H_2$ is strong. Assume towards a contradiction that the attack $H_0 \overset{q}{\rightsquigarrow} H_2$ is weak. Then, $\mathcal{K}(q) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_2)$. Since $H_2 \overset{q}{\rightsquigarrow} H_1$ is a weak attack, $\mathcal{K}(q) \models \mathsf{key}(H_2) \rightarrow \mathsf{key}(H_1)$. By transitivity, $\mathcal{K}(q) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$, a contradiction. This concludes the proof. □

A.3 Proof of Lemma 5

Proof of Lemma 5 Let $q' = q_{[x \mapsto a]}$. For every $F \in q'$, there exists a (unique) atom $\widehat{F} \in q$ such that $F = \widehat{F}_{[x \mapsto a]}$. It can be easily shown that for every $F \in q'$, we have $\widehat{F}^{+,q} \setminus \{x\} \subseteq F^{+,q'}$.

Assume $F \overset{q'}{\rightsquigarrow} G$. Then, there exists a witness $F_0 \overset{z_1}{\frown} F_1 \overset{z_2}{\frown} F_2 \ldots \overset{z_n}{\frown} F_n$ for $F \overset{q'}{\rightsquigarrow} G$ where $F_0 = F$ and $F_n = G$. It can now be easily seen that $\widehat{F_0} \overset{z_1}{\frown} \widehat{F_1} \overset{z_2}{\frown} \widehat{F_2} \ldots \overset{z_n}{\frown} \widehat{F_n}$ is a witness for $\widehat{F} \overset{q}{\rightsquigarrow} \widehat{G}$. Therefore, if the attack graph of q' is cyclic, then the attack graph of q is cyclic.

The second item in the statement of Lemma 5 follows from the observation that for all $F, G \in q'$, if $\mathcal{K}(q) \models \mathsf{key}(\widehat{F}) \rightarrow \mathsf{key}(\widehat{G})$, then $\mathcal{K}(q') \models \mathsf{key}(F) \rightarrow \mathsf{key}(G)$. □

B Proofs for Section 5

B.1 Proof of Lemma 6

Proof of Lemma 6 We show a first-order reduction from the problem UFA (Undirected Forest Accessibility) [7] to $\mathsf{CERTAINTY}(q_0)$. In UFA, we are given an acyclic undirected graph, and nodes u, v. The problem is to determine whether there is a path between u and v. The problem is Ł-complete, and remains Ł-complete when the given graph has exactly two connected components. Moreover, we can assume in the reduction that the two connected components each contain at least one edge.

Given an acyclic undirected graph $G = (V, E)$ with exactly two connected components, and two nodes u, v, we construct an uncertain database $\mathbf{db}$ as follows:

1. for every edge $\{a, b\}$ in E, the uncertain database $\mathbf{db}$ contains the facts $R_0(\underline{a}, \{a, b\})$, $R_0(\underline{b}, \{a, b\})$, $S_0(\underline{\{a, b\}}, a)$, and $S_0(\underline{\{a, b\}}, b)$, in which $\{a, b\}$ is treated as a constant; and
2. $\mathbf{db}$ contains $R_0(\underline{u}, t)$ and $R_0(\underline{v}, t)$, where t is a new value not occurring elsewhere.

Clearly, the computation of $\mathbf{db}$ from G is in $\mathbf{FO}$.

We next show that there exists a path between u and v in G if and only if every repair of $\mathbf{db}$ satisfies q_0.

Assume first that u, v belong to the same connected component. Let $\mathbf{db}'$ be the uncertain database that is constructed from the connected component not containing u, v. Let $a_0, b_0, a_1, b_1, \ldots, a_{n-1}, b_{n-1}, a_n$ be a sequence of distinct constants such that

1. $a_0 = a_n$ and for $0 \leq i < j \leq n-1$, $a_i \neq a_j$ and $b_i \neq b_j$; and
2. for $i \in \{0, \ldots, n-1\}$, $\mathbf{db}'$ contains $R_0(\underline{a_i}, b_i)$ and $S_0(\underline{b_i}, a_{i+1})$.

Since G is acyclic, any such sequence satisfies $n = 1$. An existing algorithm for $\mathsf{CERTAINTY}(q_0)$ [17, 11] will return that every repair of $\mathbf{db}'$ satisfies q_0. Consequently, every repair of $\mathbf{db}$ satisfies q_0.

For the opposite implication, assume that one connected component contains u, and the other contains v. By Lemma 1, there exists an uncertain database $\mathbf{db}'$ that is purified relative to q_0 such that q_0 is true in every repair of $\mathbf{db}'$ if and only if q_0 is true in every repair of $\mathbf{db}$. It is easy to see that if u and v belong to distinct connected components, then this purified uncertain database $\mathbf{db}'$ will be the empty database, whose only repair is the empty repair which falsifies q_0. It follows that q_0 is not true in every repair of $\mathbf{db}$. □

B.2 Proof of Lemma 8

We first show two helping lemmas.

Lemma 19 *Let q be a self-join-free Boolean conjunctive query. Let $X \subseteq \mathsf{vars}(q)$ and let $G \in q$ be an R-atom such for every $x \in X$, $G \overset{q}{\not\rightsquigarrow} x$. Let $\mathbf{r}$ be a repair of some database such that $\mathbf{r} \models q$. Let $A \in \mathbf{r}$ be an R-fact that is relevant for q in $\mathbf{r}$. Let B be key-equal to A and $\mathbf{r}_B = (\mathbf{r} \setminus \{A\}) \cup \{B\}$. Then, for every valuation ζ over X, if $\mathbf{r}_B \models \zeta(q)$, then $\mathbf{r} \models \zeta(q)$.*

Proof Let ζ be a valuation over X such that $\mathbf{r}_B \models \zeta(q)$. We can assume a valuation ζ^+ over $\mathsf{vars}(q)$ such that $\zeta^+[X] = \zeta[X]$ and $\zeta^+(q) \subseteq \mathbf{r}_B$. Thus, ζ^+ extends ζ to $\mathsf{vars}(q)$. We need to show $\mathbf{r} \models \zeta(q)$, which is obvious if $B \notin \zeta^+(q)$. Assume next $B \in \zeta^+(q)$. Since A is relevant for q in $\mathbf{r}$, we can assume a valuation μ over $\mathsf{vars}(q)$ such that $A \in \mu(q) \subseteq \mathbf{r}$. Let $q' = q \setminus \{G\}$. Let $\mathbf{r}' = \mathbf{r}_B \setminus \{B\} = \mathbf{r} \setminus \{A\}$. Since q' contains no R-atom (no self-join), $\zeta^+(q') \subseteq \mathbf{r}'$ and $\mu(q') \subseteq \mathbf{r}'$. Moreover, $\zeta^+[\mathsf{key}(G)] = \mu[\mathsf{key}(G)]$, because A and B are key-equal.

From $\mathcal{K}(q') \models \mathsf{key}(G) \rightarrow G^{+,q}$ and [16, Lemma 4.3], it follows $\zeta^+[G^{+,q}] = \mu[G^{+,q}]$.

Let τ be the complete edge-labeled undirected graph whose vertices are the atoms of q; an edge between H and H' is labeled by $\mathsf{vars}(H) \cap \mathsf{vars}(H')$.

Let τ' be the graph obtained from τ by cutting every edge whose label is included in $G^{+,q}$. Let q_G be the subset of q containing all atoms that are in τ''s strong component that contains G. Let $q_X = q \setminus q_G$.

Let κ be the valuation over $\mathsf{vars}(q)$ such that for every $x \in \mathsf{vars}(q)$,

$$\kappa(x) = \begin{cases} \mu(x) & \text{if } x \in \mathsf{vars}(q_G) \\ \zeta^+(x) & \text{if } x \in \mathsf{vars}(q_X) \end{cases}$$

We show that κ is well defined. Assume $x \in \mathsf{vars}(q_X) \cap \mathsf{vars}(q_G)$. Then, there exist atoms $F' \in q_X$ and $G' \in q_G$ such that $x \in \mathsf{vars}(F') \cap \mathsf{vars}(G')$. Since F' and G' belong to distinct strong components of τ', it follows $\mathsf{vars}(F') \cap \mathsf{vars}(G') \subseteq G^{+,q}$. Consequently, $x \in G^{+,q}$. Since $\zeta^+[G^{+,q}] = \mu[G^{+,q}]$, it follows that $\mu(x) = \zeta^+(x)$.

Obviously, $\kappa(q) \subseteq \mathbf{r}$. Finally, we show that for every $u \in X$, $\kappa(u) = \zeta(u)$. This is obvious if $u \in X \cap G^{+,q}$. Assume next that $u \in X \setminus G^{+,q}$. Since $G \overset{q}{\not\rightsquigarrow} u$ by the assumption in the statement of Lemma 19, it must be the case $u \in \mathsf{vars}(q_X)$, hence $\kappa(u) = \zeta^+(u) = \zeta(u)$. It follows $\mathbf{r} \models \zeta(q)$. This concludes the proof. □

The following helping lemma extends [16, Lemma B.1].

Lemma 20 *Let q be a self-join-free Boolean conjunctive query. Let $F \in q$ such that F has zero indegree in the attack graph of q. Let $\mathbf{r}$ be a repair of some database. Let $A \in \mathbf{r}$ such that A is relevant for q in $\mathbf{r}$.*[6] *Let B be key-equal to A and $\mathbf{r}_B = (\mathbf{r} \setminus \{A\}) \cup \{B\}$. Then, for every valuation ζ over $\mathsf{key}(F)$, if $\mathbf{r}_B \models \zeta(q)$, then $\mathbf{r} \models \zeta(q)$.*

Proof The proof is obvious if A has the same relation name as F. Assume next that relation names in A and F are distinct. We can assume some atom $G \in q \setminus \{F\}$ such that A has the same relation name as G. Since $G \overset{q}{\not\rightsquigarrow} F$, we have that for each $x \in \mathsf{key}(F)$, $G \overset{q}{\not\rightsquigarrow} x$. The desired result then follows by Lemma 19. □

Assume that a query q contains an R-atom that has no incoming attack in the attack graph of q. Paraphrasing Lemma 20, if one replaces, in a repair $\mathbf{r}$, some relevant fact A with another fact B that belongs to the same block as A, then every R-fact of $\mathbf{r}$ that was not relevant in $\mathbf{r}$, will remain non-relevant in $(\mathbf{r} \setminus \{A\}) \cup \{B\}$. Notice, however, that the fact B may be non-relevant in the new repair $(\mathbf{r} \setminus \{A\}) \cup \{B\}$.

The proof of Lemma 8 can now be given.

Proof of Lemma 8 Let $X = \mathsf{key}(F)$. Let $\mathbf{db}$ be an uncertain database. Let $\mathbf{r}$ be a repair of $\mathbf{db}$ that is $\preceq_q^X$-frugal. Let $\mathbf{s}$ be any repair of $\mathbf{db}$. Construct a maximal sequence

$$(\mathbf{r}_0, \mathbf{s}_0), (\mathbf{r}_1, \mathbf{s}_1), \ldots, (\mathbf{r}_n, \mathbf{s}_n) \tag{7}$$

where

1. $\mathbf{r}_0 = \mathbf{r}$ and $\mathbf{s}_0 = \mathbf{s}$;
2. for every $i \in \{1, \ldots, n\}$, one of the following holds:
   (a) $\mathbf{r}_i = \mathbf{r}_{i-1}$ and $\mathbf{s}_i = (\mathbf{s}_{i-1} \setminus \{A\}) \cup \{B\}$ for distinct, key-equal facts A, B such that $A \in \mathbf{s}_{i-1}$, $B \in \mathbf{r}_{i-1}$, and A is relevant for q in $\mathbf{s}_{i-1}$; or
   (b) $\mathbf{s}_i = \mathbf{s}_{i-1}$ and $\mathbf{r}_i = (\mathbf{r}_{i-1} \setminus \{A\}) \cup \{B\}$ for distinct, key-equal facts A, B such that $A \in \mathbf{r}_{i-1}$, $B \in \mathbf{s}_{i-1}$, and A is relevant for q in $\mathbf{r}_{i-1}$.

That is, the construction repeatedly replaces a fact that is relevant in one repair with its distinct, key-equal fact in the other repair. The sequence (7) is finite, since the total number of distinct relevant facts distinguishes at each step. For the last element $(\mathbf{r}_n, \mathbf{s}_n)$, it holds that the set of facts that are relevant for q in $\mathbf{r}_n$ is equal the set of facts that are relevant for q in $\mathbf{s}_n$. It follows that for every valuation θ over X,

$$\mathbf{r}_n \models \theta(q) \iff \mathbf{s}_n \models \theta(q). \tag{8}$$

By Lemma 20, for every valuation θ over X,

$$\mathbf{r}_n \models \theta(q) \quad \Longrightarrow \quad \mathbf{r} \models \theta(q) \tag{9}$$

$$\mathbf{s}_n \models \theta(q) \quad \Longrightarrow \quad \mathbf{s} \models \theta(q) \tag{10}$$

From (9) and since $\mathbf{r}$ is $\preceq_q^X$-frugal, it follows that for every valuation θ over X,

$$\mathbf{r}_n \models \theta(q) \iff \mathbf{r} \models \theta(q) \tag{11}$$

From (11), (10), and (8), it follows that for every valuation θ over X,

$$\mathbf{r} \models \theta(q) \implies \mathbf{s} \models \theta(q)$$

Since $\mathbf{s}$ is an arbitrary repair, the desired result follows. □

[6] Recall from Section 3 that $A \in \mathbf{r}$ is *relevant* for q in $\mathbf{r}$ if $A \in \theta(q) \subseteq \mathbf{r}$ for some valuation θ over $\mathsf{vars}(q)$.

C Proofs for Section 7

This section contains helping lemmas and proofs that are used in the proof of Theorem 5.

C.1 Helping Lemmas

Lemma 21 *Let q be a self-join-free Boolean conjunctive query. Let $G \in q$ and $x, y \in \mathsf{vars}(q)$ such that $\mathcal{K}(q \setminus \{G\}) \models x \rightarrow y$ and $y \notin G^{+,q}$. Then, there exists a sequence $G_1, \ldots, G_n$ of distinct atoms in q such that $x \in \mathsf{vars}(G_1)$, $y \in \mathsf{vars}(G_n)$, and for every $i \in \{1, \ldots, n-1\}$, $\mathsf{vars}(G_i) \cap \mathsf{vars}(G_{i+1}) \nsubseteq G^{+,q}$.*

Proof If $x = y$, then the desired sequence that proves the lemma is any atom that contains x. In the remainder, we treat the case $x \neq y$.

Since $\mathcal{K}(q \setminus \{G\}) \models x \rightarrow y$, we can assume a shortest sequence $F_1, F_2, \ldots, F_m$ (call it π) that is a sequential proof of $\mathcal{K}(q \setminus \{G\}) \models x \rightarrow y$, as defined by Definition 3. Note that $G \notin \{F_1, \ldots, F_m\}$. It will be the case that y occurs at a non-primary-key position in F_m.

The proof runs by induction on the length m of the proof.

Basis If $m = 1$, then the sequential proof π is F_1 with $\mathsf{key}(F_1) = \{x\}$. Notice that $\mathsf{key}(F_1) \neq \emptyset$, or else $y \in G^{+,q}$, a contradiction. The desired sequence that proves the lemma is F_1.

Induction Assume $m > 1$. Consider the last atom F_m in π. We have $\mathsf{key}(F_m) \nsubseteq G^{+,q}$, or else $y \in G^{+,q}$, a contradiction. If $x \in \mathsf{vars}(F_m)$, then the desired sequence is F_m. In the remainder, we treat the case $x \notin \mathsf{vars}(F_m)$. We can assume a variable $u \in \mathsf{key}(F_m)$ such that $u \notin G^{+,q}$. There exists an integer $k < m$ such that u occurs at a non-primary-key position in F_k. Then, $F_1, F_2, \ldots, F_k$ contains a shortest subsequence that is a sequential proof of $\mathcal{K}(q \setminus \{G\}) \models x \rightarrow u$, where $u \notin G^{+,q}$. By the induction hypothesis, there exists a sequence $G_1, \ldots, G_\ell$ of distinct atoms in q such that $x \in \mathsf{vars}(G_1)$, $u \in \mathsf{vars}(G_\ell)$, and for every $i \in \{1, \ldots, \ell-1\}$, $\mathsf{vars}(G_i) \cap \mathsf{vars}(G_{i+1}) \nsubseteq G^{+,q}$. The desired sequence that proves the lemma is $G_1, \ldots, G_\ell, F_m$. Notice that $u \in \mathsf{vars}(G_\ell) \cap \mathsf{vars}(F_m)$ and $u \notin G^{+,q}$. □

The following two lemmas are important tools for inferring attacks.

Lemma 22 *Let q be a self-join-free Boolean conjunctive query. Let $G \in q$ and $y \in \mathsf{vars}(q)$ such that $G \overset{q}{\rightsquigarrow} y$. Let $x \in \mathsf{vars}(q)$ such that $\mathcal{K}(q \setminus \{G\}) \models x \rightarrow y$. Then, $G \overset{q}{\rightsquigarrow} x$.*

Proof From $G \overset{q}{\rightsquigarrow} y$, it follows $y \notin G^{+,q}$. A witness for $G \overset{q}{\rightsquigarrow} x$ can be obtained by concatenating the sequence $G_1, \ldots, G_n$ like in the statement of Lemma 21, where $y \in \mathsf{vars}(G_n)$, with a witness of $G \overset{q}{\rightsquigarrow} y$. □

Lemma 23 *Let q be a self-join-free Boolean conjunctive query. Let $G \in q$ and $y \in \mathsf{vars}(q)$ such that $G \overset{q}{\rightsquigarrow} y$ and $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow y$. If $\mathcal{K}(q) \models x \rightarrow y$, then $G \overset{q}{\rightsquigarrow} x$.*

Proof The desired result is obvious in case $x = y$. In the remainder of the proof, we treat the case $x \neq y$. Assume $\mathcal{K}(q) \models x \rightarrow y$. Then, we can assume a shortest sequence $F_1, F_2, \ldots, F_n$ that is a sequential proof of $\mathcal{K}(q) \models x \rightarrow y$ as defined by Definition 3.

Let $V = \left(\bigcup_{j=1}^{n} \mathsf{vars}(F_j)\right) \cup \{x\}$. For every $u \in V \setminus \{x\}$, we define the *depth* of u, denoted $d(u)$, as the smallest integer j such that $u \in \mathsf{vars}(F_j)$. Furthermore, we define $d(x) = 0$. Clearly, $d(y) = n$.

We show next that if G attacks some variable $u \in V$ with $d(u) > 0$ and $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow u$, then also G attacks some variable $u' \in V$ with $d(u') < d(u)$ and $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow u'$.

Assume $G \overset{q}{\rightsquigarrow} u$ with $d(u) = k > 0$ and $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow u$. It must be the case that $u \in \mathsf{vars}(F_k) \setminus \mathsf{key}(F_k)$. Also, $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow \mathsf{key}(F_k)$ (otherwise, $\mathcal{K}(q) \models \mathsf{key}(G) \rightarrow u$, a contradiction). Then, there must be some $w \in \mathsf{key}(F_k)$ such that $\mathcal{K}(q) \not\models \mathsf{key}(G) \rightarrow w$, which implies $w \notin G^{+,q}$. Clearly, $d(w) < k$ and $G \overset{q}{\rightsquigarrow} w$.

It follows $G \overset{q}{\rightsquigarrow} x$. $\square$

C.2 Proof of Lemma 10

Proof of Lemma 10 $\boxed{\text{Item 1}}$ Let $\pi = H_1, H_2, \ldots, H_n$ be a shortest sequence that is a sequential proof of $\mathcal{K}(q) \models x \to z$. Clearly, for $i \in \{1, \ldots, n\}$, we have $\mathcal{K}(q) \models x \to \mathsf{key}(H_i)$, hence $H_i \overset{q}{\not\rightsquigarrow} x$ and $H_i \overset{q}{\not\rightsquigarrow} z$, by the assumption in the statement of Lemma 10.

Let $\mathbf{db}$ be an uncertain database that is the input to $\mathsf{CERTAINTY}(q)$.

Sublemma 5 *Let a, b be constants. If some $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}$ satisfies $q_{[x,z \mapsto a,b]}$, then for every repair $\mathbf{r}_B$ of $\mathbf{db}$, for every valuation θ over $\mathsf{vars}(q)$ such that $\theta(q) \subseteq \mathbf{r}_B$, if $\theta(x) = a$, then $\theta(z) = b$.*

Proof Let $\mathbf{r}_A$ be a $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}$. Let θ_A be a valuation over $\mathsf{vars}(q)$ such that $\theta_A(q) \subseteq \mathbf{r}_A$, and $\theta_A(x) = a$ and $\theta_A(z) = b$. That is, $\mathbf{r}_A \models q_{[x,z \mapsto a,b]}$. Let $\mathbf{r}_B$ be a repair of $\mathbf{db}$ such that for some valuation θ_B over $\mathsf{vars}(q)$, we have $\theta_B(q) \subseteq \mathbf{r}_B$ and $\theta_B(x) = a$. We need to show $\theta_B(z) = b$.

We show how to inductively construct a maximal sequence

$$(p_0, \mathbf{r}_0, \zeta_0), (p_1, \mathbf{r}_1, \zeta_1), \ldots, (p_m, \mathbf{r}_m, \zeta_m)$$

where for every $j \geq 0$,

1. $\mathbf{r}_j$ is a $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}$;
2. ζ_j is a valuation over $\mathsf{vars}(q)$ such that $\zeta_j(q) \subseteq \mathbf{r}_j$;
3. $\zeta_j(x) = a$ and $\zeta_j(z) = b$, i.e., $\mathbf{r}_j \models q_{[x,z \mapsto a,b]}$;
4. $p_j \in \{0, 1, \ldots, n\}$ and for all $i \in \{1, \ldots, p_j\}$, $\zeta_j(H_i) = \theta_B(H_i)$;
5. $p_0 < p_1 < \cdots < p_j$.

Intuitively, one can think of p_j as an index in π indicating that ζ_j and θ_B agree on all variables in $H_1, H_2, \ldots, H_{p_j}$.

For the basis of the induction, we choose $(p_0, \mathbf{r}_0, \zeta_0) = (0, \mathbf{r}_A, \theta_A)$. In this way, the above conditions are obviously satisfied for $j = 0$.

For the induction step $j \to j+1$, let p_{j+1} be be the smallest integer k such that $\zeta_j(H_k) \neq \theta_B(H_k)$. It can be seen that $\zeta_j(H_k)$ and $\theta_B(H_k)$ must be key-equal. Let $\mathbf{r}_{j+1} = (\mathbf{r}_j \setminus \{\zeta_j(H_k)\}) \cup \{\theta_B(H_k)\}$. By Lemma 19 and since $\mathbf{r}_j$ is $\preceq_q^{\{x,z\}}$-frugal, it follows $\mathbf{r}_{j+1} \models q_{[x,z \mapsto a,b]}$. So there exists a valuation μ over $\mathsf{vars}(q)$ such that $\mu(q) \subseteq \mathbf{r}_{j+1}$, and $\mu(x) = a$ and $\mu(z) = b$. From $\mathbf{r}_j \setminus \{\zeta_j(H_k)\} = \mathbf{r}_{j+1} \setminus \{\theta_B(H_k)\}$ and $\mu(x) = \zeta_j(x)$, it will be that case that $\mu(H_i) = \zeta_j(H_i)$ for all $i \in \{1, \ldots, p_j\}$. By the condition 4, $\mu(H_i) = \theta_B(H_i)$ for all $i \in \{1, \ldots, p_j\}$. Then by our choice of p_{j+1} and our construction of $\mathbf{r}_{j+1}$, we have $\mu(H_i) = \theta_B(H_i)$ for all $i \in \{1, \ldots, p_{j+1}\}$. We choose $\zeta_{j+1} = \mu$. With these choices, the above conditions 1–5 are satisfied for $j+1$.

For $j = m$, we will have that ζ_m and θ_B agree on all variables in $\bigcup_{i=1}^{n} \mathsf{vars}(H_i)$. Since $\zeta_m(z) = b$, it follows $\theta_B(z) = b$. This concludes the proof of Sublemma 5. $\dashv$

Sublemma 6 *Let a, b_1, b_2 be constants such that $b_1 \neq b_2$. If $\mathbf{db} \models q_{[x,z \mapsto a,b_1]}$ and $\mathbf{db} \models q_{[x,z \mapsto a,b_2]}$, then for every $\preceq_q^{\{x,z\}}$-frugal repair $\mathbf{r}_f$ of $\mathbf{db}$, $\mathbf{r}_f \not\models q_{[x \mapsto a]}$.*

Proof Assume the existence of two valuations θ_1, θ_2 over $\mathsf{vars}(q)$ such that $\theta_1(q) \subseteq \mathbf{db}$, $\theta_2(q) \subseteq \mathbf{db}$, $\theta_1(x) = \theta_2(x) = a$, and $b_1 = \theta_1(z) \neq \theta_2(z) = b_2$. Then, there exist two repairs $\mathbf{r}_1, \mathbf{r}_2$ such that $\theta_1(q) \subseteq \mathbf{r}_1$ and $\theta_2(q) \subseteq \mathbf{r}_2$.

Assume towards a contradiction the existence of a $\preceq_q^{\{x,z\}}$-frugal repair $\mathbf{r}_f$ of $\mathbf{db}$ such that $\mathbf{r}_f \models q_{[x \mapsto a]}$. Then, we can assume a valuation μ over $\mathsf{vars}(q)$ such that $\mu(q) \subseteq \mathbf{r}_f$ and $\mu(x) = a$. By Sublemma 5, $\theta_1(z) = \mu(z)$ and $\theta_2(z) = \mu(z)$, hence $\theta_1(z) = \theta_2(z)$, a contradiction. This concludes the proof of Sublemma 6. $\dashv$

Construct a maximal sequence

$$\mathbf{db}_0, a_1, \mathbf{db}_1, a_2, \mathbf{db}_2, \ldots, a_\ell, \mathbf{db}_\ell \tag{12}$$

where $\mathbf{db}_0 = \mathbf{db}$ and for $i \in \{1, \ldots, \ell\}$,

1. there exist two constants b_i, c_i such that $b_i \neq c_i$, $\mathbf{db}_{i-1} \models q_{[x,z \mapsto a_i, b_i]}$, and $\mathbf{db}_{i-1} \models q_{[x,z \mapsto a_i, c_i]}$; and
2. $\mathbf{db}_i = \mathbf{db}_{i-1} \setminus \widehat{\mathbf{db}}_{i-1}$, where $\widehat{\mathbf{db}}_{i-1}$ is the smallest subset of $\mathbf{db}_{i-1}$ that includes every block $\mathbf{b}$ of $\mathbf{db}_{i-1}$ such that a_i occurs in some fact of $\mathbf{b}$. Recall from Section 3 that we assume uncertain databases to be typed.

Then, the following are equivalent:

1. every repair of $\mathbf{db}$ satisfies q;
2. every $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}$ satisfies q; and
3. every $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}_\ell$ satisfies q.

Equivalence of items 1 and 2 follows from Lemma 2. Equivalence of items 2 and 3 follows from Sublemma 6, using induction on increasing $i \in \{0, \ldots, \ell\}$.

Since the sequence (12) is maximal, it must be that $\mathbf{db}_\ell \Vdash_q x \rightarrow z$. Let $\mathbf{db}'$ be the database that includes $\mathbf{db}_\ell$ and such that for every valuation θ, if $\theta(q) \subseteq \mathbf{db}_\ell$, then $\mathbf{db}'$ contains $T^c(\underline{\theta(x)}, \theta(z))$. Clearly, the set of T-facts of $\mathbf{db}'$ is consistent, and the following are equivalent:

1. every $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}_\ell$ satisfies q;
2. every $\preceq_q^{\{x,z\}}$-frugal repair of $\mathbf{db}'$ satisfies $q \cup \{T^c(\underline{x}, z)\}$; and
3. every repair of $\mathbf{db}'$ satisfies $q \cup \{T^c(\underline{x}, z)\}$.

Finally, it can be easily seen that $\mathbf{db}'$ can be computed from $\mathbf{db}$ in polynomial time. This concludes the proof of the first item.

$\boxed{\text{Item 2}}$ Define $q' = q \cup \{T^c(\underline{x}, z)\}$. We show that for all $F, G \in q$, if $F \overset{q'}{\rightsquigarrow} G$, then $F \overset{q}{\rightsquigarrow} G$. For every attack $F \overset{q'}{\rightsquigarrow} G$, we distinguish two cases depending on F.

Case $\mathcal{K}(q \setminus \{F\}) \models x \rightarrow z$. Then clearly, $F^{+,q} = F^{+,q'}$. The only hard case is where a witness for the attack $F \overset{q'}{\rightsquigarrow} G$ contains the atom $T^c(\underline{x}, z)$. Then, $z \notin F^{+,q'}$, hence $z \notin F^{+,q}$. From Lemma 21, it follows that there exists a witness for $F \overset{q}{\rightsquigarrow} G$.

Case $\mathcal{K}(q \setminus \{F\}) \not\models x \rightarrow z$. Since $\mathcal{K}(q) \models x \rightarrow z$, it must be the case that every sequential proof of $\mathcal{K}(q) \models x \rightarrow z$ contains F. Then $\mathcal{K}(q) \models x \rightarrow \mathsf{key}(F)$. By the assumption in the statement of Lemma 10, $F \overset{q}{\not\rightsquigarrow} x$ and $F \overset{q}{\not\rightsquigarrow} z$. Assume towards a contradiction that a witness of $F \overset{q'}{\rightsquigarrow} G$ contains $T^c(\underline{x}, z)$. Then, since $F^{+,q} \subseteq F^{+,q'}$, it must be the case that $F \overset{q}{\rightsquigarrow} x$ or $F \overset{q}{\rightsquigarrow} z$, a contradiction. We conclude by contradiction that no witness of $F \overset{q'}{\rightsquigarrow} G$ contains $T^c(\underline{x}, z)$. Since $F^{+,q} \subseteq F^{+,q'}$, it follows $F \overset{q}{\rightsquigarrow} G$.

Assume that the attack graph of q' contains a strong cycle C. Since the atom $T^c(\underline{x}, z)$ cannot be in C (since it has no outgoing attacks), the attack graph of q contains the same cycle C. It can be easily seen that C is strong in the attack graph of q. $\square$

C.3 Proof of Lemma 11

We first show two helping lemmas.

Lemma 24 *Let q be a self-join-free Boolean conjunctive query. Let F be an atom of q. Let G be an atom with a fresh relation name such that* $\mathsf{key}(G) = \mathsf{key}(F)$ *and* $\mathsf{vars}(G) = \mathsf{vars}(F)$*. Let $q' = (q \setminus \{F\}) \cup \{G\}$. Then,*

1. *there exists a polynomial-time many-one reduction from* $\mathsf{CERTAINTY}(q)$ *to* $\mathsf{CERTAINTY}(q')$*; and*[7]

2. *if the attack graph of* q *contains no strong cycle, then the attack graph of* q' *contains no strong cycle either.*

Proof The proof of the second item is straightforward.

For the first item, let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$. By Lemma 1, we can compute in polynomial time a database $\mathbf{db}_p$ such that $\mathbf{db}_p$ is purified relative to q and such that every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}_p$ satisfies q.

Let $\mathbf{db}'$ be the uncertain database that includes $\mathbf{db}_p$ and such that whenever $\mathbf{db}_p$ contains $\theta(F)$ for some valuation θ over $\mathsf{vars}(F)$, then $\mathbf{db}'$ contains $\theta(G)$. Notice here that $\mathsf{vars}(F) = \mathsf{vars}(G)$ and, since $\mathbf{db}_p$ is purified, whenever $A \in \mathbf{db}_p$ has the same relation name as F, then there exists a valuation θ over $\mathsf{vars}(F)$ such that $A = \theta(F)$. It can now be easily verified that every repair of $\mathbf{db}_p$ satisfies q if and only if every repair of $\mathbf{db}'$ satisfies q'. $\square$

Notice that the roles of F and G can be switched in the statement of Lemma 24, showing that $\mathsf{CERTAINTY}(q)$ and $\mathsf{CERTAINTY}(q')$ are polynomially equivalent.

Example 20 If $F = R(\underline{a, x, x, y, y, z, z, b}, u)$ and $G = S(\underline{x, y, z}, u)$, then $\mathsf{key}(F) = \mathsf{key}(G)$ and $\mathsf{vars}(F) = \mathsf{vars}(G)$. So Lemma 24 implies that we can replace F with G in the study of $\mathsf{CERTAINTY}(q)$. $\lhd$

Lemma 25 *Let* q *be a self-join-free Boolean conjunctive query. Let* $R(\underline{\vec{x}}, \vec{y})$ *be an atom of* q *with mode* i*. Let* $q_0 = \{R_1^c(\underline{\vec{x}}, w)\ R_2^c(\underline{w}, \vec{x}),\ S(\underline{w}, \vec{y})\}$*, where* R_1*,* R_2 *are fresh relation names of mode c,* S *is a fresh relation name of mode i, and* w *is a variable such that* $w \notin \mathsf{vars}(q)$*. Let* $q' = (q \setminus \{R(\underline{\vec{x}}, \vec{y})\}) \cup q_0$*. Then,*

1. *there exists a polynomial-time many-one reduction from* $\mathsf{CERTAINTY}(q)$ *to* $\mathsf{CERTAINTY}(q')$*; and*

2. *if the attack graph of* q *contains no strong cycle, then the attack graph of* q' *contains no strong cycle either.*

Proof $\boxed{\text{Item 1}}$ Assume that the signature of R is $[n, k]$. Let $\mathbf{db}$ be an uncertain database that is input to $\mathsf{CERTAINTY}(q)$. Define an injective function h that maps every element in $(\mathbf{adom}(\mathbf{db}))^k$ to a fresh constant not occurring elsewhere. Let $\mathbf{db}'$ be the database obtained from $\mathbf{db}$ by replacing each fact $R(\underline{\vec{a}}, \vec{b})$ with the following three facts:

$$R_1^c(\underline{\vec{a}}, h(\vec{a})), R_2^c(\underline{h(\vec{a})}, \vec{a}), \text{and } S(\underline{h(\vec{a})}, \vec{b}).$$

Since the function h is injective, the set of R_1-facts and R_2-facts of $\mathbf{db}'$ is consistent. Hence, $\mathbf{db}'$ is a legal input to $\mathsf{CERTAINTY}(q')$. Intuitively, R_1-facts encode the function h, and R_2-facts affirm that h is injective. It remains to be shown that every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}'$ satisfies q'.

Define $f : \mathsf{rset}(\mathbf{db}) \to \mathsf{rset}(\mathbf{db}')$ such that for every $\mathbf{r} \in \mathsf{rset}(\mathbf{db})$,

- if $\mathbf{r}$ contains $R(\underline{\vec{a}}, \vec{b})$, then $f(\mathbf{r})$ contains $S(\underline{h(\vec{a})}, \vec{b})$;
- $f(\mathbf{r})$ contains all R_1-facts and all R_2-facts of $\mathbf{db}'$; and
- if T is a relation name in q such that $T \neq R$, then $f(\mathbf{r})$ contains exactly the same T-facts as $\mathbf{r}$.

The following can be easily verified for every $\mathbf{r} \in \mathsf{rset}(\mathbf{db})$:

- $f(\mathbf{r})$ is indeed a repair of $\mathbf{db}'$; and
- q is true in $\mathbf{r}$ if and only if q' is true in $f(\mathbf{r})$.

The desired result follows from the easy observation that f is bijective.

$\boxed{\text{Item 2}}$ By a little abuse of notation, we will denote atoms by their relation name. First, observe that $\mathcal{K}([\![q']\!]) \models w \to \mathsf{vars}(\vec{x})$ and $\mathcal{K}([\![q']\!]) \models \mathsf{vars}(\vec{x}) \to w$. This implies that for any atom $F \in q \setminus \{R\}$, we have $F^{+,q} = F^{+,q'} \setminus \{w\}$. Furthermore, $R^{+,q} = S^{+,q'} \setminus \{w\}$.

Notice that atoms R_1 and R_2 have mode c, and hence have no outgoing attacks in the attack graph of q'. We will now show that for all $F, G \in q \setminus \{R\}$,

- if $S \stackrel{q'}{\rightsquigarrow} G$, then $R \stackrel{q}{\rightsquigarrow} G$;

[7] We know that there exists such a first-order reduction. However, polynomial-time is sufficient here and allows for an easier proof.

- if $F \overset{q'}{\rightsquigarrow} S$, then $F \overset{q}{\rightsquigarrow} R$; and
- if $F \overset{q'}{\rightsquigarrow} G$, then $F \overset{q}{\rightsquigarrow} G$.

To this extent, assume an attack $F \overset{q'}{\rightsquigarrow} G$ where $F, G \in (q \setminus \{R\}) \cup \{S\}$. We can assume a witness

$$F_0 \overset{z_1}{\frown} F_1 \overset{z_2}{\frown} F_2 \dots \overset{z_n}{\frown} F_n \tag{13}$$

for $F \overset{q'}{\rightsquigarrow} G$ where $F_0 = F$ and $F_n = G$. We can assume without loss of generality that $1 \leq i < j \leq n$ implies $z_i \neq z_j$, and that $0 \leq i < j \leq n$ implies $F_i \neq F_j$. Moreover, since $\mathsf{vars}(R_1) = \mathsf{vars}(R_2)$, we can assume that R_2 does not occur in (13). We distinguish two cases.

Case $F_0 = S$. Since $\{w\} \cup \mathsf{vars}(\vec{x}) \subseteq S^{+,q'}$, we have that R_1 and R_2 do not occur in the sequence (13), and that $w \notin \{z_1, \dots, z_n\}$. Then, $R \overset{z_1}{\frown} F_1 \overset{z_2}{\frown} F_2 \dots \overset{z_n}{\frown} F_n$ is a witness for $R \overset{q}{\rightsquigarrow} F_n$.

Case $F_n = S$. It may be the case that $w \in \{z_1, \dots, z_n\}$. Then, by the form of q_0, we can assume a smallest integer i such that $z_i \in \mathsf{vars}(\vec{x}) \cup \mathsf{vars}(\vec{y})$. Then, $F_0 \overset{z_1}{\frown} F_1 \overset{z_2}{\frown} F_2 \dots \overset{z_i}{\frown} R$ is a witness for $F_0 \overset{q}{\rightsquigarrow} R$.

Case $F_0 \neq S \neq F_n$. The only hard case is when the sequence (13) is of one of the following forms:

$$\begin{array}{l} F_0 \dots \overset{x}{\frown} R_1^c \overset{w}{\frown} S \overset{y}{\frown} \dots F_n, \quad \text{or} \\ F_0 \dots \overset{y}{\frown} S \overset{w}{\frown} R_1^c \overset{x}{\frown} \dots F_n, \end{array}$$

where $x \in \mathsf{vars}(\vec{x})$ and $y \in \mathsf{vars}(\vec{y})$. Then, $y \notin {F_0}^{+,q'}$ and $x \notin {F_0}^{+,q'}$. It follows $y \notin {F_0}^{+,q}$ and $x \notin {F_0}^{+,q}$, which implies that we can replace the subsequence $R_1^c \overset{w}{\frown} S$ (or $S \overset{w}{\frown} R_1^c$) with R to obtain a witness for $F_0 \overset{q}{\rightsquigarrow} F_n$.

It follows that every cycle in the attack graph of q' is present in the attack graph of q modulo a replacement of S with R.

Assume that the attack graph of q contains no strong cycle. Let C' be an elementary directed cycle in the attack graph of q'. Let C be the directed cycle in the attack graph of q obtained from C' by replacing S with R. The attack cycle C must be weak. Then, the attack cycle C' will be weak, because for every $F, G \in q \setminus \{R\}$,

- if $\mathcal{K}(q) \models \mathsf{key}(F) \rightarrow \mathsf{key}(G)$, then $\mathcal{K}(q') \models \mathsf{key}(F) \rightarrow \mathsf{key}(G)$;
- if $\mathcal{K}(q) \models \mathsf{key}(F) \rightarrow \mathsf{key}(R)$, then $\mathcal{K}(q') \models \mathsf{key}(F) \rightarrow \mathsf{key}(S)$; and
- if $\mathcal{K}(q) \models \mathsf{key}(R) \rightarrow \mathsf{key}(G)$, then $\mathcal{K}(q') \models \mathsf{key}(S) \rightarrow \mathsf{key}(G)$.

This concludes the proof. □

The proof of Lemma 11 is now straightforward.

Proof of Lemma 11 Apply the reductions of Lemmas 24 and 25. Then repeatedly apply the reduction of Lemma 10 until it can no longer be applied. Notice that the reduction of Lemma 10 consists in adding atoms of the form $T^c(\underline{x}, z)$. □

C.4 Proof of Lemma 13

Proof of Lemma 13 Assume that $k, x_0, \dots, x_{k-1}, \vec{y}, q_0, q_1$ are as in Definition 6. Let $K = T(\underline{u}, x_0, \dots, x_{k-1}, \vec{y})$.

Since the Markov cycle $\mathcal{C}$ is premier, we can assume an atom $F_0 \in q$ with mode i and $x \in \mathsf{vars}(q)$ such that $\mathsf{key}(F_0) = \{x\}$ and $x \overset{q,\mathsf{M}*}{\longrightarrow} x_0$ and $\mathcal{K}(q) \models x_0 \rightarrow x$.

Assume that the attack graph of q contains no strong cycle.

Sublemma 7 $\mathcal{K}(q_0 \cup [\![q]\!]) \cup \{u \rightarrow x_0, x_0 \rightarrow u\} \models \mathcal{K}(q_1)$.

Proof $\mathcal{K}(q_1)$ is logically equivalent to $\{u \to z \mid z \in \mathsf{vars}(q_0)\} \cup \{x_i \to u \mid 0 \leq i \leq k-1\}$.

Let $z \in \mathsf{vars}(q_0)$. Clearly, for all $i, j \in \{0, \ldots, k-1\}$, $\mathcal{K}(q_0 \cup [\![q]\!]) \models x_i \to x_j$. It is then obvious that $\mathcal{K}(q_0 \cup [\![q]\!]) \models x_0 \to z$. Hence, $\mathcal{K}(q_0 \cup [\![q]\!]) \cup \{u \to x_0, x_0 \to u\} \models u \to z$.

Let $i \in \{0, \ldots, k-1\}$. As argued before, $\mathcal{K}(q_0 \cup [\![q]\!]) \models x_i \to x_0$. Hence, $\mathcal{K}(q_0 \cup [\![q]\!]) \cup \{u \to x_0, x_0 \to u\} \models x_i \to u$.

It follows that every functional dependency of $\mathcal{K}(q_1)$ is logically implied by $\mathcal{K}(q_0 \cup [\![q]\!]) \cup \{u \to x_0, x_0 \to u\}$. $\dashv$

Sublemma 8 $\mathcal{K}(q_1) \models \mathcal{K}(q_0) \cup \{u \to x_0, x_0 \to u\}$.

Proof Obviously, $\mathcal{K}(q_1) \models u \to x_0$, $\mathcal{K}(q_1) \models x_0 \to u$, and for every $i \in \{0, \ldots, k-1\}$, $\mathcal{K}(q_1) \models x_i \to \mathsf{vars}(q_0)$. Every atom of q_0 is of the form $R(\underline{x_i}, \vec{z})$ where $i \in \{0, \ldots, k-1\}$ and $\mathsf{vars}(\vec{z}) \subseteq \mathsf{vars}(q_0)$. Since $\mathcal{K}(q_1) \models x_i \to \mathsf{vars}(q_0)$, we have $\mathcal{K}(q_1) \models x_i \to \mathsf{vars}(\vec{z})$. $\dashv$

Sublemmas 7 and 8 immediately lead to the following results.

Sublemma 9 $\mathcal{K}(q^*) \equiv \mathcal{K}(q) \cup \{u \to x_0, x_0 \to u\}$.

Sublemma 10 *For every $F \in q \setminus q_0$ such that F has mode i, we have $\mathcal{K}(q^* \setminus \{F\}) \equiv \mathcal{K}(q \setminus \{F\}) \cup \{x_0 \to u, u \to x_0\}$.*

Sublemma 11 *For every $F \in q \setminus q_0$ such that F has mode i, we have $F^{+,q} = F^{+,q^*} \setminus \{u\}$.*

Proof Let $F \in q \setminus q_0$ such that the mode of F is i. From Sublemma 10, it follows that $F^{+,q} \subseteq F^{+,q^*}$. Since $u \notin \mathsf{vars}(q)$, it follows $F^{+,q} \subseteq F^{+,q^*} \setminus \{u\}$.

The inclusion $F^{+,q^*} \setminus \{u\} \subseteq F^{+,q}$ follows from Sublemma 10 and the observation that in the computation of F^{+,q^*}, the functional dependencies $x_0 \to u$ and $u \to x_0$ are useless, except for inferring $u \in F^{+,q^*}$ from $x_0 \in F^{+,q^*}$. $\dashv$

All U_i-atoms have mode c and hence have no outgoing attacks in the attack graph of q^*. The following lemma states that all attacks among atoms of $q \setminus q_0$ in the attack graph of q^* are also present in the attack graph of q.

Sublemma 12 *For all $F, G \in q \setminus q_0$, if $F \overset{q^*}{\rightsquigarrow} G$, then $F \overset{q}{\rightsquigarrow} G$.*

Proof Let $F, G \in q \setminus q_0$ such that $F \overset{q^*}{\rightsquigarrow} G$. Then, we can assume a witness for $F \overset{q^*}{\rightsquigarrow} G$ of the following form:

$$H_0 \overset{z_1}{\frown} H_1 \overset{z_2}{\frown} H_2 \ldots \overset{z_n}{\frown} H_n, \tag{14}$$

where $H_0 = F$ and $H_n = G$. We can assume without loss of generality that $1 \leq i < j \leq n$ implies $z_i \neq z_j$, and that $0 \leq i < j \leq n$ implies $H_i \neq H_j$. Since ${H_0}^{+,q} \subseteq {H_0}^{+,q^*}$ by Sublemma 11, it follows that $\{z_1, \ldots, z_n\} \cap {H_0}^{+,q} = \emptyset$.

If the sequence (14) contains no atom of q_1, then it is also a witness for $F \overset{q}{\rightsquigarrow} G$, and the desired result holds. In the remainder, assume that the sequence (14) contains an atom of q_1. Because of the structure of q_1, we can assume without loss of generality that K is the only atom of q_1 that occurs in the sequence (14). So we can assume $\ell \in \{1, \ldots, n-1\}$ such that $H_\ell = K$. Clearly, $z_\ell, z_{\ell+1} \in \mathsf{vars}(q_0)$ and by Sublemma 11, $z_\ell, z_{\ell+1} \notin F^{+,q}$.

For the variable $z_{\ell+1}$, there exists some $i \in \{0, \ldots, k-1\}$ such that either $z_{\ell+1} = x_i$ or the atom $R(\underline{x_i}, z_{\ell+1})$ belongs to q_0. Since $\mathcal{K}(q_0 \cup [\![q]\!]) \models x_i \to x_j$ for all $i, j \in \{0, \ldots, k-1\}$, it follows $\mathcal{K}(q \setminus \{F\}) \models x_i \to z_\ell$. From $F \overset{q}{\rightsquigarrow} z_\ell$, it follows $F \overset{q}{\rightsquigarrow} x_i$ by Lemma 22, and hence $F \overset{q}{\rightsquigarrow} z_{\ell+1}$. It can then be easily seen that there exists a witness for $F \overset{q}{\rightsquigarrow} G$. $\dashv$

We finally focus on attacks in the attack graph of q^* that involve the atom K.

Sublemma 13 *For every $H \in q^*$, if $H \overset{q^*}{\rightsquigarrow} K$, then $H \in q \setminus q_0$, and both $\mathcal{K}(q) \models \mathsf{key}(F_0) \to \mathsf{key}(H)$ and $\mathcal{K}(q) \models \mathsf{key}(H) \to \mathsf{key}(F_0)$.*

Proof Let $H \in q^*$ such that $H \overset{q^*}{\rightsquigarrow} K$. Since U_i-atoms have no outgoing attacks in the attack graph of q^*, it must be the case that $H \in q \setminus q_0$. The Markov graph of q contains a directed path from x to x_0 (recall $\{x\} = \mathsf{key}(F_0)$); let M be the set of variables on this path. We now distinguish two cases.

- If $\mathsf{key}(H) \subseteq M$, then clearly $\mathcal{K}(q) \models \mathsf{key}(F_0) \rightarrow \mathsf{key}(H)$. Since $\mathcal{K}(q) \models \mathsf{key}(H) \rightarrow x_0$ and $\mathcal{K}(q) \models x_0 \rightarrow \mathsf{key}(F_0)$, we obtain $\mathcal{K}(q) \models \mathsf{key}(H) \rightarrow \mathsf{key}(F_0)$.

- Otherwise, $\mathcal{K}(q \setminus \{H\}) \models \mathsf{key}(F_0) \rightarrow z$ for every $z \in \mathsf{vars}(q_0)$. Since $H \overset{q^*}{\rightsquigarrow} K$, it must be that $H \overset{q}{\rightsquigarrow} z$ for some $z \in \mathsf{vars}(q_0)$. Then, $H \overset{q}{\rightsquigarrow} x$ by Lemma 22, and consequently $H \overset{q}{\rightsquigarrow} F_0$. Then, it must be the case that H belongs to the initial strong component of the attack graph of q that also contains F_0. Since the attack graph of q contains no strong cycle, we have $\mathcal{K}(q) \models \mathsf{key}(F_0) \rightarrow \mathsf{key}(H)$ and $\mathcal{K}(q) \models \mathsf{key}(H) \rightarrow \mathsf{key}(F_0)$.

This concludes the proof of Sublemma 13. $\dashv$

We can now complete the proof of Lemma 13. Assume towards a contradiction that the attack graph of q^* contains a strong cycle. By Lemma 4, the attack graph of q^* contains a strong cycle of size 2. So we can assume atoms $H_0, H_1 \in q^*$ such that $H_0 \overset{q^*}{\rightsquigarrow} H_1 \overset{q^*}{\rightsquigarrow} H_0$, and at least one of the attacks is strong.

Case $H_0, H_1 \in q \setminus q_0$. By Sublemma 12, $H_0 \overset{q}{\rightsquigarrow} H_1 \overset{q}{\rightsquigarrow} H_0$. Since the attack graph of q contains no strong attack cycles, we have $\mathcal{K}(q) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$ and $\mathcal{K}(q) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$. From Sublemma 9, it follows $\mathcal{K}(q^*) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$ and $\mathcal{K}(q^*) \models \mathsf{key}(H_0) \rightarrow \mathsf{key}(H_1)$, contradicting that $H_0 \overset{q^*}{\rightsquigarrow} H_1 \overset{q^*}{\rightsquigarrow} H_0$ is a strong attack cycle.

Case $H_0 = K$ (the case $H_1 = K$ is symmetrical). Then, $\mathsf{key}(H_0) = \{u\}$. By Sublemma 13, $H_1 \in q \setminus q_0$, and both $\mathcal{K}(q) \models \mathsf{key}(F_0) \rightarrow \mathsf{key}(H_1)$ and $\mathcal{K}(q) \models \mathsf{key}(H_1) \rightarrow \mathsf{key}(F_0)$. From Sublemma 9 and $\mathcal{K}(q) \models x_0 \rightarrow \mathsf{key}(F_0)$, it follows $\mathcal{K}(q^*) \models u \rightarrow \mathsf{key}(H_1)$. From Sublemma 9 and $\mathcal{K}(q) \models \mathsf{key}(F_0) \rightarrow x_0$ (because there is a Markov path from x to x_0), it follows $\mathcal{K}(q^*) \models \mathsf{key}(H_1) \rightarrow u$. But then $H_0 \overset{q^*}{\rightsquigarrow} H_1 \overset{q^*}{\rightsquigarrow} H_0$ is a weak attack cycle, a contradiction.

In both cases, we conclude by contradiction that the attack graph of q^* contains no strong attack cycle. $\square$

C.5 Proof of Lemma 14

We use the following helping lemma.

Lemma 26 *Let q be a self-join-free Boolean conjunctive query such that*

- *for every atom $F \in q$, if F has mode i, then F is simple-key and* $\mathsf{key}(F) \neq \emptyset$*;*
- *q is saturated; and*
- *the attack graph of q contains no strong cycle.*

Let F_0 be an atom of q that belongs to an initial strong component of the attack graph of q, and let $\mathsf{key}(F_0) = \{y\}$. *Let $x \in \mathsf{vars}(q)$ such that $\mathcal{K}(q) \models x \rightarrow y$ and $\mathcal{K}(q) \models y \rightarrow x$. Then, there exists $z \in \mathsf{vars}(q)$ with $\mathsf{C}_q(z) \neq \emptyset$ such that $x \xrightarrow{\mathsf{M}} z$ and $\mathcal{K}(q) \models z \rightarrow y$.*

Proof If $x \xrightarrow{\mathsf{M}} y$, then the desired result holds for $z = y$. In the remainder of the proof, we treat the case $x \not\xrightarrow{\mathsf{M}} y$.

Let q_0 be a minimal (with respect to $\subseteq$) subset of q such that $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!] \cup q_0) \models x \rightarrow y$. Obviously, $q_0 \cap \mathsf{C}_q(x) = \emptyset$ and $q_0 \cap [\![q]\!] = \emptyset$. Let p be a minimal (with respect to $\subseteq$) subset of $\mathsf{C}_q(x) \cup [\![q]\!] \cup q_0$ such that the atoms of p can be sequentially ordered into a sequential proof (call it π) of $\mathcal{K}(q) \models x \rightarrow y$. Clearly, π must contain all atoms of q_0.

From $x \not\xrightarrow{\mathsf{M}} y$, it follows $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!]) \not\models x \rightarrow y$. Hence, $q_0 \neq \emptyset$. Let G be the leftmost atom in π such that $G \in q_0$. Notice that $\mathsf{key}(G) \neq \emptyset$ by the premise in the statement of Lemma 26. We can assume $z \in \mathsf{vars}(q)$ such that $G \in \mathsf{C}_q(z)$. Since G is chosen leftmost, $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!]) \models x \rightarrow z$, hence $x \xrightarrow{\mathsf{M}} z$ and $\mathsf{C}_q(z) \neq \emptyset$. It remains to be shown that $\mathcal{K}(q) \models z \rightarrow y$.

Assume towards a contradiction that $\mathcal{K}(q) \not\models z \rightarrow y$. In the next paragraph, we show that π contains an atom H such that for some $w_1, w_2 \in \mathsf{key}(H)$,

1. $\mathcal{K}(q) \models z \rightarrow w_1$ but $\mathcal{K}([\![q]\!]) \not\models z \rightarrow w_1$; and
2. $\mathcal{K}(q) \not\models z \rightarrow w_2$.

Existence of H, w_1, and w_2. Let $V = \mathsf{vars}(p) \cup \{x\}$ and let the sequential proof π be $H_1, H_2, \ldots, H_\ell$. For every $u \in V \setminus \{x\}$, we define the *depth* of u, denoted $d(u)$, as the smallest integer j such that $u \in \mathsf{vars}(H_j)$. Furthermore, we define $d(x) = 0$. Clearly, $d(y) = \ell$.

For $u \in V$ and $i, j \in \{0, \ldots, \ell\}$, we write $i \overset{u}{\rightarrowtail} j$ if $d(u) = i$ and $j \in \{i+1, \ldots, \ell\}$ such that $u \in \mathsf{key}(H_j)$. Intuitively, if $i > 0$, then $i \overset{u}{\rightarrowtail} j$ says that the variable u is introduced in the sequential proof by H_i, and "used" later on by H_j. We can assume $k \in \{1, \ldots, \ell\}$ such that $G = H_k$. Clearly, $d(z) < k$. It can be easily seen that the following can be assumed without loss of generality.

Simple-Things-First Condition: for every $u \in V$, if $\mathcal{K}([\![q]\!]) \models z \rightarrow u$, then $d(u) < k$.

Since no atom of π is redundant, there exists a sequence

$$k_0 \overset{u_1}{\rightarrowtail} k_1 \overset{u_2}{\rightarrowtail} k_2 \cdots \overset{u_m}{\rightarrowtail} k_m$$

where $k_0 = k$ and $k_m = \ell$. Thus, y occurs at a non-primary-key position in H_{k_m}. For all $i \in \{1, \ldots, m\}$, $d(u_i) \geq k$, hence $\mathcal{K}([\![q]\!]) \not\models z \rightarrow u_i$ by the *Simple-Things-First Condition*.

Since $\mathcal{K}(q) \not\models z \rightarrow y$, we have $\mathcal{K}(q) \not\models z \rightarrow \mathsf{key}(H_{k_m})$. Hence, we can assume a smallest integer $j \in \{1, 2, \ldots, m\}$ such that $\mathcal{K}(q) \not\models z \rightarrow \mathsf{key}(H_{k_j})$. Then obviously, $\mathcal{K}(q) \models z \rightarrow \mathsf{key}(H_{k_{j-1}})$, hence $\mathcal{K}(q) \models z \rightarrow u_j$. We can choose $w_1 = u_j$ and $H = H_{k_j}$. Further, since $\mathcal{K}(q) \not\models z \rightarrow \mathsf{key}(H_{k_j})$, we can choose $w_2 \in \mathsf{key}(H_{k_j})$ such that $\mathcal{K}(q) \not\models z \rightarrow w_2$. We conclude that H, w_1, and w_2 indeed exist.

Since q is saturated, from $\mathcal{K}(q) \models z \rightarrow w_1$ and $\mathcal{K}([\![q]\!]) \not\models z \rightarrow w_1$, it follows that there exists an atom $G' \in q$ such that $\mathcal{K}(q) \models z \rightarrow \mathsf{key}(G')$ and such that either $G' \overset{q}{\rightsquigarrow} z$ or $G' \overset{q}{\rightsquigarrow} w_1$. Clearly, G' is an atom with mode i.

We show $\mathcal{K}(q \setminus \{G'\}) \models x \rightarrow z$. Assume towards a contradiction that $\mathcal{K}(q \setminus \{G'\}) \not\models x \rightarrow z$. Since $\mathcal{K}(\mathsf{C}_q(x) \cup [\![q]\!]) \models x \rightarrow z$, it must be the case that $G' \in \mathsf{C}_q(x)$, hence $\mathsf{key}(G') = \{x\}$. Then, from $\mathcal{K}(q) \models z \rightarrow \mathsf{key}(G')$ and $\mathcal{K}(q) \models x \rightarrow y$, it follows $\mathcal{K}(q) \models z \rightarrow y$, a contradiction. We conclude by contradiction that $\mathcal{K}(q \setminus \{G'\}) \models x \rightarrow z$.

Two cases can occur.

Case $G' \overset{q}{\rightsquigarrow} w_1$. Since $\mathcal{K}(q) \not\models \mathsf{key}(G') \rightarrow w_2$ (or otherwise $\mathcal{K}(q) \models z \rightarrow w_2$, a contradiction), we have $w_2 \notin G'^{+,q}$, hence $G' \overset{q}{\rightsquigarrow} w_2$. Since $\mathcal{K}(q) \models x \rightarrow w_2$, it follows by Lemma 23 that $G' \overset{q}{\rightsquigarrow} x$.

Case $G' \overset{q}{\rightsquigarrow} z$. Since $\mathcal{K}(q \setminus \{G'\}) \models x \rightarrow z$, we have that $G' \overset{q}{\rightsquigarrow} x$ by Lemma 22.

Thus, at this part of the proof, we have $G' \overset{q}{\rightsquigarrow} x$. We now distinguish two cases.

Case $\mathcal{K}(q) \models \mathsf{key}(G') \rightarrow x$. From $\mathcal{K}(q) \models z \rightarrow \mathsf{key}(G')$ and $\mathcal{K}(q) \models x \rightarrow y$, we have $\mathcal{K}(q) \models z \rightarrow y$, a contradiction.

Case $\mathcal{K}(q) \not\models \mathsf{key}(G') \rightarrow x$. From $\mathcal{K}(q) \models y \rightarrow x$ and $G' \overset{q}{\rightsquigarrow} x$, it follows from Lemma 23 that $G' \overset{q}{\rightsquigarrow} y$, which implies $G' \overset{q}{\rightsquigarrow} F_0$. Since F_0 belongs to an initial strong component of q's attack graph and since the attack graph of q contains no strong cycle, the attack $G' \overset{q}{\rightsquigarrow} F_0$ must be weak, so $\mathcal{K}(q) \models \mathsf{key}(G') \rightarrow y$. Since $\mathcal{K}(q) \models z \rightarrow \mathsf{key}(G')$, we obtain $\mathcal{K}(q) \models z \rightarrow y$, a contradiction.

We conclude by contradiction that $\mathcal{K}(q) \models z \rightarrow y$. □

The proof of Lemma 14 is given next.

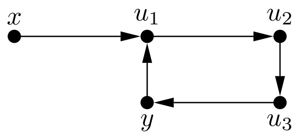

Figure 4: Markov graph of the query in Example 21.

Proof of Lemma 14 By repeated application of Lemma 3, the initial strong component with two or more atoms will contain two atoms F_0, G such that $F_0 \stackrel{q}{\rightsquigarrow} G \stackrel{q}{\rightsquigarrow} F_0$.

Let $\{w_0\} = \mathsf{key}(F_0)$ (and thus $\mathsf{C}_q(w_0) \neq \emptyset$) and $\{y\} = \mathsf{key}(G)$. Since the attack graph of q contains no strong cycle, we have $\mathcal{K}(q) \models w_0 \rightarrow y$ and $\mathcal{K}(q) \models y \rightarrow w_0$. By Lemma 26, there exists $w_1 \in \mathsf{vars}(q)$ such that $w_0 \xrightarrow{\mathsf{M}} w_1$, $\mathsf{C}_q(w_1) \neq \emptyset$, and $\mathcal{K}(q) \models w_1 \rightarrow y$. The latter implies that $\mathcal{K}(q) \models w_1 \rightarrow w_0$ as well.

By repeated application of Lemma 26, for every $k > 0$, there exists a Markov path $w_0 \xrightarrow{\mathsf{M}} w_1 \cdots \xrightarrow{\mathsf{M}} w_k$, where $\mathsf{C}_q(w_i) \neq \emptyset$ for every $i \in \{0, \dots, k\}$, and $\mathcal{K}(q) \models w_k \rightarrow w_0$. Since $\mathsf{vars}(q)$ is a finite set, at some point we will have $w_k = w_i$ for some i with $i < k$, at which point we have found the desired Markov cycle. $\square$

The proof of Lemma 14 actually shows a slightly stronger result than the statement of Lemma 14. The proof shows that whenever $R(\underline{x}, \vec{z})$ belongs to an attack cycle of size 2 that is part of an initial strong component of the attack graph, then the Markov graph contains a directed path from x to a Markov cycle with the desired properties. This is illustrated by the following example.

Example 21 Let $q = \{R_1(\underline{x}, u_1),\ R_2(\underline{u_1}, u_2),\ R_3(\underline{u_2}, u_3),\ R_4(\underline{u_3}, y),\ R_5(\underline{y}, u_1),\ S^c(\underline{u_2, y}, x)\}$. In the attack graph of q, every R_i-atom attacks every other atom of q, and all these attacks are weak.

The Markov graph of q is shown in Figure 4. As predicted by the proof of Lemma 14, for every variable among x, y, u_1, u_2, u_3, there is a path that starts from the variable and ends in a Markov cycle. Notice, however, that x itself is not part of a Markov cycle. $\lhd$

C.6 Proof of Lemma 16

Proof of Lemma 16 Construct a maximal sequence

$$\mathbf{db}_0, \mathbf{g}_1, \mathbf{db}_1, \mathbf{g}_2, \mathbf{db}_2, \dots, \mathbf{g}_n, \mathbf{db}_n \tag{15}$$

such that $\mathbf{db}_0 = \mathbf{db}$ and for every $i \in \{1, \dots, n\}$,

1. $\mathbf{g}_i$ is a gblock of $\mathbf{db}_{i-1}$ such that some repair of $\mathbf{g}_i$ is not grelevant for q in $\mathbf{db}_{i-1}$;
2. $\mathbf{db}_i = \mathbf{db}_{i-1} \setminus \mathbf{g}_i$.

Clearly, $\mathbf{db}_n$ is gpurified relative to q, and by repeated application of Lemma 15, every repair of $\mathbf{db}$ satisfies q if and only if every repair of $\mathbf{db}_n$ satisfies q.

It remains to be shown that $\mathbf{db}_n$ can be computed in polynomial time. Clearly, the above sequence (15) satisfies $n \leq |\mathbf{db}|$. The condition 1 can be tested in polynomial time, as argued in the sequel of this proof.

First, every uncertain database that is purified relative to q has at most polynomially many gblocks, and every gblock has at most polynomially many repairs. Further, for any repair $\mathbf{s}$ of some gblock $\mathbf{g}_i$, the following are equivalent:

1. $\mathbf{s}$ is grelevant for q in $\mathbf{db}_{i-1}$;
2. there exists a repair $\mathbf{r}$ of $\mathbf{db}$ such that $\mathbf{s} \subseteq \mathbf{r}$ and for some valuation θ over $\mathsf{vars}(q)$ and some fact $A \in \mathbf{s}$, $A \in \theta(q) \subseteq \mathbf{r}$; and

3. $(\mathbf{db}_{i-1} \setminus \mathbf{db}_{\mathbf{s}}) \cup \mathbf{s} \models q$, where $\mathbf{db}_{\mathbf{s}}$ is the subset of $\mathbf{db}$ that contains all facts whose relation name occurs in $\mathbf{s}$.

The first two items are equivalent by definition. Equivalence of the last two items follows from the observation that if some atom $A \in \mathbf{s}$ is relevant for q in $\mathbf{r}$, then every atom of $\mathbf{s}$ must be relevant for q in $\mathbf{r}$. The latter test is obviously in polynomial time. $\square$