Consistent Query Answering for Primary Keys and Conjunctive Queries with Negated Atoms

Paraschos Koutris
University of Wisconsin-Madison
Madison, WI, USA
paris@cs.wisc.edu

Jef Wijsen
Université de Mons
Mons, Belgium
jef.wijsen@umons.ac.be

ABSTRACT

This paper studies query answering on databases that may be inconsistent with respect to primary key constraints. A repair is any consistent database that is obtained by deleting a minimal set of tuples. Given a Boolean query q, the problem CERTAINTY(q) takes a database as input and asks whether q is true in every repair of the database. A significant complexity classification task is to determine, given q, whether CERTAINTY(q) is first-order definable (and thus solvable by a single SQL query). This problem has been extensively studied for self-join-free conjunctive queries. An important extension of this class of queries is to allow negated atoms. It turns out that if negated atoms are allowed, CERTAINTY(q) can express some classical matching problems. This paper studies the existence and construction of first-order definitions for CERTAINTY(q) for q in the class of self-join-free conjunctive queries with negated atoms.

CCS CONCEPTS

• **Information systems → Relational database query languages**; • **Theory of computation → Logic and databases**; **Incomplete, inconsistent, and uncertain databases**;

KEYWORDS

Conjunctive queries; consistent query answering; negation; primary keys

ACM Reference Format:
Paraschos Koutris and Jef Wijsen. 2018. Consistent Query Answering for Primary Keys and Conjunctive Queries with Negated Atoms. In *PODS'18: 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, June 10–15, 2018, Houston, TX, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3196959.3196982

1 INTRODUCTION

Consistent query answering for primary keys is the following problem. Fix a relational database schema with one primary key constraint per relation. A database is allowed to violate the primary key constraints of its schema, and is consistent if it satisfies these constraints. A *repair* of a database **db** is any consistent database that is obtained by deleting a minimal set of tuples from **db**. Given a Boolean query q, CERTAINTY(q) is the problem that takes a database as input, and asks whether q evaluates to true on every repair of the database.

R	$\underline{x}$	y
	Alice	Bob
	Alice	George
	Maria	Bob
	Maria	John

S	$\underline{y}$	x
	Bob	Alice
	Bob	Maria
	George	Alice
	George	Maria

Figure 1: Example of an inconsistent database violating the primary key constraints. Blocks of key-equal tuples are separated by dashed lines.

The complexity of the problem CERTAINTY(q) for self-join-free Boolean conjunctive queries has been settled in [19]. An important extension of conjunctive queries (CQs) is to allow negated atoms. This class of queries has received several names: CQs with atomic negation, CQs with negated atoms, CQs with safe negation, CQs with negation. These are first-order queries of the form:

$$\exists\vec{x}\left(R_1(\vec{x}_1) \land \cdots \land R_\ell(\vec{x}_\ell) \land \neg R_{\ell+1}(\vec{x}_{\ell+1}) \land \cdots \land \neg R_n(\vec{x}_n)\right),$$

subject to the restriction, called *safety*, that every variable that occurs within the scope of a negation, also occurs in an atom that is not negated. Such a query is Boolean if it contains no free variables, and self-join-free if all relation names are pairwise distinct. In this paper, we study the complexity of CERTAINTY(q) for self-join-free conjunctive queries with negated atoms. The extension with negation allows, among others, expressing some widely studied matching problems, as illustrated by Examples 1.1 and 1.2. Such expressiveness obviously does not come for free. It is a general observation—and our work is no exception—that whenever negation is added to a negation-free query language, reasoning problems become more involved.

Example 1.1. Let $q_1 = \exists x \exists y (R(\underline{x}, y) \land \neg S(\underline{y}, x))$, where primary keys are underlined. Intuitively, one can think of a fact $R(\underline{g}, b)$ as "girl g knows boy b." Conversely, $S(\underline{b}, g)$ means that "boy b knows girl g." See Fig. 1 for an example inconsistent database. The repairs of R correspond to all ways in which each girl can choose one boy she knows. Conversely, the repairs of S correspond to all ways in which each boy can choose one girl he knows. Then, given a database of R-facts and S-facts, the query q_1 is true in every repair if and only if it is impossible to match each girl to a distinct boy such that every girl is matched to a boy she knows and by whom she

ACM ISBN 978-1-4503-4706-8/18/06…$15.00
https://doi.org/10.1145/3196959.3196982

is known. For the database in Fig. 1, observe that such a matching is possible: we can pair Alice with George, and Maria with Bob. This pairing corresponds to the repair with facts $R(\underline{\text{Alice}}, \text{George})$, $R(\underline{\text{Maria}}, \text{Bob})$, $S(\underline{\text{George}}, \text{Alice})$, and $S(\underline{\text{Bob}}, \text{Maria})$, which does not satisfy q_1.

If every R-fact $R(\underline{g}, b)$ has a corresponding S-fact $S(\underline{b}, g)$, and vice versa, and if our databases have the same number of girls and boys, then CERTAINTY(q_1) is equivalent to the complement of the Bipartite Perfect Matching (BPM) problem. □

In general, our work concerns the following complexity classification question: given a Boolean conjunctive query q with negated atoms, what is the complexity of the problem CERTAINTY(q)? A fine-grained complexity classification is too ambitious when one realizes that the exact complexity of BPM is already an open problem (see, for example, [9, 10]). In this paper, we concentrate on deciding whether or not CERTAINTY(q) is in the complexity class **FO**. In our setting, **FO** is the class of decision problems that take as input a relational database (over a fixed schema) and can be solved by a single first-order query. Membership of a database-related problem in **FO** is of practical interest, because this means that the problem can be solved using standard SQL database technology. A first-order query φ that solves CERTAINTY(q) is called a *consistent first-order rewriting* for q.

For example, for the query q_1 of Example 1.1, we can conclude that CERTAINTY(q_1) is not in **FO**, because BPM is known not to be in **FO** (see the proof of Lemma 5.2 for a more detailed argumentation). Thus, q_1 has no consistent first-order rewriting. Example 1.2 introduces another matching problem that can be reduced to the complement of CERTAINTY(q_{Hall}), where q_{Hall} is a conjunctive query with negated atoms that does have a consistent first-order rewriting.

Example 1.2. Consider the following problem, which we call S-Covering: given a set S and a list $T_1, T_2, \ldots, T_\ell$ of (possibly empty) subsets of S, can we pick *at most* one element from each T_i such that every element of S is picked once? That is, we are asking whether there exists an injective function $f : S \to \{1, 2, \ldots, \ell\}$ such that for every element $a \in S$, $a \in T_{f(a)}$. Hall's Marriage Theorem [14] gives a sufficient and necessary condition for the existence of such function.

S-Covering can be first-order reduced to the complement of CERTAINTY(q_{Hall}) with

$$q_{Hall} = \exists x(S(\underline{x}) \land \neg N_1(\underline{c}, x) \land \cdots \land \neg N_\ell(\underline{c}, x)),$$

where c is a constant. The reduction constructs a database **db** that contains $S(\underline{a})$ for every $a \in S$, and that contains $N_i(\underline{c}, a)$ whenever $a \in T_i$. The repairs correspond to all ways of picking elements from the ℓ sets. A repair of **db** that falsifies q_{Hall} will satisfy $\neg q_{Hall}$:

$$\neg q_{Hall} \equiv \forall x(S(\underline{x}) \to (N_1(\underline{c}, x) \lor \cdots \lor N_\ell(\underline{c}, x))).$$

Obviously, such a repair, if it exists, picks at most one element from each T_i, and picks every element of S.

An immediate corollary of our results will be that if ℓ is fixed, CERTAINTY(q_{Hall}) can be solved by a single first-order query. Figure 2 shows the query for solving the case $\ell = 3$. The details are postponed until Example 6.12. □

We are not aware of other existing work on theoretical aspects of CERTAINTY(q) for conjunctive queries with negated atoms. This paper makes a first significant contribution in this direction by establishing the following result (see Theorem 4.3):

> Given a self-join-free Boolean conjunctive query q with weakly-guarded negation, it is decidable whether the problem CERTAINTY(q) is in **FO**. Moreover, if CERTAINTY(q) is in **FO**, then a first-order query that solves CERTAINTY(q), called a *consistent first-order rewriting*, can be effectively constructed.

Negation is called *weakly-guarded* if whenever some variables x and y occur together in a negated atom, they also occur together in some non-negated atom. Weakly-guarded negation is safe (take x and y to be the same), and strictly weaker than *guarded negation* (where for each negated atom, there exists a non-negated atom containing all the variables of the negated atom). To prove the above result, we significantly extend existing techniques in consistent query answering.

We briefly discuss the restrictions we impose on the queries we consider. The restriction to Boolean queries allows us to define CERTAINTY(q) as a decision problem. The extension to queries with free variables is easy, essentially because free variables can be treated as constants. This is argued in [19, Section 3.3] for queries without negated atoms, and the same argumentation holds if negated atoms are allowed. The restriction to self-join-free queries is used in several proofs. The complexity classification of problems CERTAINTY(q) remains an open task if self-joins are allowed, even in the absence of negated atoms. The restriction to weakly-guarded negation is also used in several proofs, and will be discussed in Section 7. If we look beyond conjunctive queries, it is natural to consider classes of queries that contain disjunctions of (restricted) Boolean conjunctive queries. The complexity classification of CERTAINTY(q) for queries q with disjunction is still largely open.

Organization. This paper is organized as follows. Section 2 discusses related work. Section 3 introduces our theoretical framework. Section 4 introduces our main theorem, which establishes a sufficient and necessary condition for CERTAINTY(q) to be in **FO**, where q is a self-join-free Boolean conjunctive query with weakly-guarded negation. Section 5 shows that the condition is necessary; and Section 6 shows that the condition is sufficient. Section 7 discusses in more depth the notion of weakly-guarded negation. Finally, Section 8 concludes the paper.

2 RELATED WORK

Consistent query answering (CQA) goes back to the seminal work by Arenas, Bertossi, and Chomicki [1], and is the topic of the monograph [4]. The term CERTAINTY(q) was coined in [28] to refer to CQA for Boolean queries q on databases that violate primary keys, one per relation, which are fixed by q's schema. The complexity classification of CERTAINTY(q) for the class of self-join-free Boolean conjunctive queries has attracted much research; see, for example, [12, 15, 17]. These works were concluded by [18, 19], where it was shown that for every query q in the class of self-join-free Boolean conjunctive queries, CERTAINTY(q) is either in **P** or **coNP**-complete. Furthermore, it was shown that, given a query q

in this class, it is decidable whether CERTAINTY(q) is in **FO**; and if CERTAINTY(q) is in **FO**, a consistent first-order rewriting for q (i.e., a first-order query that solves CERTAINTY(q)) can be effectively constructed.

Little is known about CERTAINTY(q) beyond self-join-free conjunctive queries. Fontaine [11] obtained the following interesting result for UCQ, the class of Boolean queries that can be expressed as disjunctions of Boolean conjunctive queries (possibly with self-joins). A daring conjecture is that for every query q in UCQ, CERTAINTY(q) is either in **P** or **coNP**-complete. Fontaine showed that this conjecture implies Bulatov's dichotomy theorem for conservative CSP [6], which has an involved proof. This relationship between CQA and CSP was further explored in [22]. The complexity of CQA for aggregation queries with respect to violations of functional dependencies has been studied in [2].

The counting variant of the problem CERTAINTY(q), denoted ♯CERTAINTY(q), asks to determine the number of repairs that satisfy some Boolean query q. In [25], it was shown that for every self-join-free Boolean conjunctive query q, the counting problem ♯CERTAINTY(q) is either in **FP** or **♯P**-complete. For conjunctive queries q with self-joins, the complexity of ♯CERTAINTY(q) has been established under the restriction that all primary keys consist of a single attribute [26].

The work on CQA has inspired work on inconsistency-tolerant query answering in ontology-based data access [5, 20, 21]. It is common to assume that the ontological theory (usually a TBox in some description logic) is correct, while the database (ABox) may contain erroneous facts that are not consistent with the ontological theory. A repair is defined as a maximal (with respect to set inclusion) subset of the ABox that is consistent with respect to the TBox. Restricted forms of negation show up in different studies that deal with inconsistent and incomplete information; see, for example, [13] and the references therein.

Guarded-Negation First-Order Logic (GNFO) [3] restricts first-order logic by requiring that all occurrences of negation are of the form $\alpha \wedge \neg\phi$, where α is an atomic formula, a guard, containing all free variables of ϕ. Note that our results cover first-order queries that are not in GNFO. For example, negation is weakly-guarded in $\exists x \exists y \exists z(R(\underline{x}, y) \wedge S(\underline{y}, z) \wedge T(\underline{z}, x) \wedge \neg N(\underline{x}, y, z))$, a formula not in GNFO.

In practice, CQA problems have been solved by means of expressive solvers for ASP or BIP [16, 23, 24]. These solvers may generally not outperform the best SQL engines on large database problems that can be solved in **FO**.

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.

Atoms and key-equal facts. Each *relation name* R of arity n, where $n \geq 1$, 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 $k = 1$, and *all-key* if $n = k$. Elements of the primary key are called *primary-key positions*, while $k + 1, k + 2, \ldots, n$ 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 the expression $R(s_1, \ldots, s_n)$ 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*, denoted $R_1(\underline{\vec{a}_1}, \vec{b}_1) \sim R_2(\underline{\vec{a}_2}, \vec{b}_2)$, if $R_1 = R_2$ and $\vec{a}_1 = \vec{a}_2$. An R-atom or an R-fact is called *simple-key* if R is simple-key, and is called *all-key* if R is all-key.

We will denote atoms by F, G, H, N, P. 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 $\mathsf{vars}(F) = \mathsf{vars}(\vec{x}) \cup \mathsf{vars}(\vec{y})$.

(Possibly inconsistent) databases, 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. A *database* is a finite set **db** of facts using only the relation names of the schema.

A *block* of **db** is a maximal set of key-equal facts of **db**. The term R-block refers to a block of R-facts, i.e., facts with relation name R. A database **db** is *consistent* if no two distinct facts are key-equal (i.e., if every block of **db** is a singleton). A *repair* of **db** is a maximal (with respect to set inclusion) consistent subset of **db**. We write rset(**db**) for the set of repairs of **db**.

For every Boolean query q, CERTAINTY(q) is the problem that takes a database **db** as input and asks whether q evaluates to true on every repair of **db**. If CERTAINTY(q) is in **FO**, then a first-order query that solves CERTAINTY(q) is called a *consistent first-order rewriting* for q. In this paper, the complexity class **FO** stands for the class of decision problems that take a database as input and that can be solved in first-order logic with equality and constants, but without other built-in predicates or function symbols.

Boolean conjunctive queries with negated atoms. A self-join-free Boolean conjunctive query with negated atoms is a first-order sentence q of the form

$$\exists \vec{u}(F_1 \wedge F_2 \wedge \cdots \wedge F_\ell \wedge \neg F_{\ell+1} \wedge \neg F_{\ell+2} \wedge \cdots \wedge \neg F_m),$$

where $0 \leq \ell \leq m$, each F_i is an atom ($1 \leq i \leq m$), no two atoms have the same relation name, and $\bigcup_{j=\ell+1}^{m} \mathsf{vars}(F_j) \subseteq \bigcup_{i=1}^{\ell} \mathsf{vars}(F_i)$. The latter condition is called the *safety condition*. When a self-join-free conjunctive query is understood, we will often use a relation name at places where an atom is expected; for example, we will use R instead of $R(\underline{x}, y)$. This simplifies the technical treatment and causes no confusion, because q cannot have two distinct atoms with the same relation name R.

We denote by sjfBCQ$^\neg$ the set of self-join-free Boolean conjunctive queries with negated atoms. For a query q in sjfBCQ$^\neg$, we denote by $\mathsf{vars}(q)$ the set of variables that occur in q. If $\vec{x} = \langle x_1, \ldots, x_\ell \rangle$ is a sequence of distinct variables in $\mathsf{vars}(q)$, and $\vec{c} = \langle c_1, \ldots, c_\ell \rangle$ is a sequence of constants, then $q_{[\vec{x} \mapsto \vec{c}]}$ denotes the query obtained from q by replacing all occurrences of x_i with c_i, for all $1 \leq i \leq \ell$.

A *literal* is an atom or the negation of an atom. Queries q in sjfBCQ$^\neg$ will be denoted as sets of literals: $q = \{F_1, \ldots, F_\ell, \neg F_{\ell+1}, \ldots, \neg F_m\}$. We define $q^+ := \{F_1, \ldots, F_\ell\}$, the set of all positive literals in q, and $q^- := \{F_{\ell+1}, \ldots, F_m\}$, the set of atoms whose negation belongs to q. Then, q is satisfied by a database **db** if there exists a valuation θ over $\text{vars}(q)$ such that for every $P \in q^+$, $\theta(P) \in$ **db**, and for every $N \in q^-$, $\theta(N) \notin$ **db**.

Example 3.1. For $q = \{R(\underline{x, y}), \neg S(\underline{x}, y), \neg T(\underline{y}, x)\}$, we have $q^+ = \{R(\underline{x, y})\}$ and $q^- = \{S(\underline{x}, y), T(\underline{y}, x)\}$. □

Typed databases. For every variable x, we assume an infinite set of constants, denoted $\text{type}(x)$, such that $x \neq y$ implies $\text{type}(x) \cap \text{type}(y) = \emptyset$. We say that a database **db** is *typed relative to* q, with $q \in$ sjfBCQ$^\neg$, if for every atom $R(x_1, \ldots, x_n)$ in $q^+ \cup q^-$, for every $i \in \{1, \ldots, n\}$, if x_i is a variable, then for every fact $R(a_1, \ldots, a_n)$ in **db**, $a_i \in \text{type}(x_i)$ and the constant a_i does not occur in q.

If $q \in$ sjfBCQ$^\neg$, then, because of the absence of self-joins, a database **db** can be trivially transformed into a database **db′** that is typed relative to q such that CERTAINTY(q) yields the same answer on problem instances **db** and **db′**.

Guarded and weakly-guarded negation. Let q be a query in sjfBCQ$^\neg$. We say that negation in q is *guarded* if for every $N \in q^-$, there exists $P \in q^+$ such that $\text{vars}(N) \subseteq \text{vars}(P)$. We say that negation in q is *weakly-guarded* if for every $N \in q^-$, for all $x, y \in \text{vars}(N)$, there exists $P \in q^+$ such that $x, y \in \text{vars}(P)$. Obviously, if negation is guarded, it is also weakly-guarded.

Example 3.2. Negation is not weakly-guarded in the query $\{X(x), Y(y), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\}$, because the variables x and y occur together in a negated atom, but do not occur together in a non-negated atom. Negation is weakly-guarded, but not guarded, in the query $\{R(\underline{x, y}, z, u), S(\underline{y, w}, z), T(\underline{x}, u, w), \neg N(\underline{x, y, z, u, w})\}$. □

Key-relevant facts. Let q be a query in sjfBCQ$^\neg$. Assume that F is an R-atom in $q^+ \cup q^-$ (thus, q contains either F or $\neg F$). Let **r** be a consistent database. We say that an R-fact A of **r** is *key-relevant for* q *in* **r** if there exists a valuation θ over $\text{vars}(q)$ such that $\mathbf{r} \models \theta(q)$ and $\theta(F) \sim A$.

Example 3.3. Let $q_1 = \{R(\underline{x}, y), \neg S(\underline{y}, x)\}$, and let $\mathbf{r} = \{R(\underline{b}, 1), S(\underline{1}, a), S(\underline{2}, a)\}$. The only valuation θ over $\{x, y\}$ such that $\mathbf{r} \models \theta(q)$ is $\theta = \{x \mapsto b, y \mapsto 1\}$. Then, $S(\underline{1}, a)$ is key-relevant, because it is key-equal to $S(\underline{\theta(y)}, \theta(x)) = S(\underline{1}, b)$. On the other hand, $S(\underline{2}, a)$ is not key-relevant. □

4 THE MAIN THEOREM

In this section we present our main theorem, which implies decidability of membership of CERTAINTY(q) in **FO**, when negation in q is weakly-guarded. In order to state the theorem, we first need to introduce the notion of *attack graph* for queries in sjfBCQ$^\neg$.

4.1 The Attack Graph

Attack graphs were introduced in [29] for α-acyclic self-join-free conjunctive queries, and were extended in [19] to all self-join-free conjunctive queries. We now define a further extension that deals with negated atoms. For a set p of non-negated atoms, we define $\mathcal{K}(p)$ as the following set of functional dependencies whose left- and right-hand sides are sets of variables:

$$\mathcal{K}(p) := \{\text{key}(F) \rightarrow \text{vars}(F) \mid F \in p\}.$$

Let q be a query in sjfBCQ$^\neg$. For every atom $F \in q^+ \cup q^-$, we define the set of variables

$$F^{\oplus, q} := \{x \in \text{vars}(q) \mid \mathcal{K}(q^+ \setminus \{F\}) \models \text{key}(F) \rightarrow x\}.$$

That is, $F^{\oplus, q}$ is the closure of $\text{key}(F)$ with respect to the functional dependencies arising in atoms that are not negated and distinct from F.

Attacks between variables. Let $F \in q^+ \cup q^-$ and $u, w \in \text{vars}(q)$ such that $u \in \text{vars}(F)$. We write $F|u \overset{q}{\rightsquigarrow} w$ if there exists a sequence $(u_0, u_1, \ldots, u_\ell)$ of variables in $\text{vars}(q)$ such that $\ell \geq 0$ and

- $u_0 = u$ and $u_\ell = w$;
- for all $i \in \{0, \ldots, \ell - 1\}$, there exists an atom $P \in q^+$ such that $u_i, u_{i+1} \in \text{vars}(P)$; and
- for all $i \in \{0, \ldots, \ell\}$, $u_i \notin F^{\oplus, q}$.

The sequence $(u_0, u_1, \ldots, u_\ell)$ will be called a *witness* for $F|u \overset{q}{\rightsquigarrow} w$. We write $F|u \overset{q}{\not\rightsquigarrow} w$ if it is not the case that $F|u \overset{q}{\rightsquigarrow} w$. We write $F \overset{q}{\rightsquigarrow} w$ (and we say that F *attacks* w) if $F|u \overset{q}{\rightsquigarrow} w$ for some $u \in \text{vars}(F)$. We write $F \overset{q}{\not\rightsquigarrow} w$ if it is not the case that $F \overset{q}{\rightsquigarrow} w$.

Attacks between atoms. The *attack graph* of q is a directed graph, without self-loops, whose vertex set is $q^+ \cup q^-$. There is a directed edge from F to G ($F \neq G$) if $F \overset{q}{\rightsquigarrow} y$ for some $y \in \text{key}(G)$. We write $F \overset{q}{\rightsquigarrow} G$ (and we say that F *attacks* G) if the attack graph of q contains a directed edge from F to G. We write $F \overset{q}{\not\rightsquigarrow} G$ if the attack graph of q contains no directed edge from F to G.

We should note here that, when q contains no negated atoms, then the notion of attack as defined above is identical to the notion of attack used in [19].

Example 4.1. Let $q_2 = \{P(\underline{x, y}), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\}$. As $\mathcal{K}(q_2{}^+) = \{xy \rightarrow xy\} \equiv \{\}$, we necessarily have $P^{\oplus, q_2} = \{x, y\}$, $R^{\oplus, q_2} = \{x\}$, and $S^{\oplus, q_2} = \{y\}$. We have $R|y \overset{q_2}{\rightsquigarrow} y$ and $S|x \overset{q_2}{\rightsquigarrow} x$. The attack graph of q_2 contains four edges: $R \overset{q_2}{\rightsquigarrow} S$, $S \overset{q_2}{\rightsquigarrow} R$, $R \overset{q_2}{\rightsquigarrow} P$, $S \overset{q_2}{\rightsquigarrow} P$. □

Example 4.2. Let $q_3 = \{P(\underline{x}, y), \neg N(\underline{c}, y)\}$. Then, $\mathcal{K}(q_3{}^+ \setminus \{P\}) = \{\}$ and $\mathcal{K}(q_3{}^+ \setminus \{N\}) = \{x \rightarrow y\}$. Consequently, $P^{\oplus, q_3} = \{x\}$ and $N^{\oplus, q_3} = \{\}$. We have $P|y \overset{q_3}{\rightsquigarrow} y$ and $N|y \overset{q_3}{\rightsquigarrow} y$ and $N|y \overset{q_3}{\rightsquigarrow} x$. A witness for $N|y \overset{q_3}{\rightsquigarrow} x$ is the sequence (y, x). The attack graph of q_3 contains one edge: $N \overset{q_3}{\rightsquigarrow} P$. Note that $P \overset{q_3}{\not\rightsquigarrow} N$ (because P attacks no variable belonging to N's primary key). □

4.2 Main Theorem Statement

We can now state our main theorem.

Theorem 4.3. *Let q be a self-join-free Boolean conjunctive query with negated atoms such that negation in q is weakly-guarded. Then,*

1. *if the attack graph of q is cyclic, then* CERTAINTY(q) *is* **L**-*hard (and thus not in* **FO***); and*

(2) *if the attack graph of q is acyclic, then* CERTAINTY(q) *is in* **FO**.

Moreover, if CERTAINTY(q) *is in* **FO**, *then a consistent first-order rewriting for q can be effectively constructed.*

It is not hard to figure out that it is decidable, given q, whether the attack graph of q is acyclic (in fact, this can be decided in polynomial time in the size of q). Therefore, Theorem 4.3 gives us a decision procedure to decide membership in **FO**.

Example 4.4. Consider $q_2 = \{P(\underline{x,y}), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\}$ from Example 4.1. Negation in q_2 is weakly-guarded. Since $R \overset{q_2}{\rightsquigarrow} S$ and $S \overset{q_2}{\rightsquigarrow} R$, the attack graph of q_2 is cyclic, and thus Theorem 4.3 tells us that CERTAINTY(q_2) is not in **FO**. □

Example 4.5. Consider $q_3 = \{P(\underline{x}, y), \neg N(\underline{c}, y)\}$ from Example 4.2. Since negation in q_3 is weakly-guarded and since the attack graph of q_3 is acyclic (because $P \overset{q}{\not\rightsquigarrow} N$), the query q_3 has a consistent first-order rewriting. The construction of such rewriting will be explained in Section 6, and will read as follows:

$$\exists x \exists y P(\underline{x}, y) \wedge \forall z \left(N(\underline{c}, z) \rightarrow \exists x \left(\begin{array}{l} \exists y P(\underline{x}, y) \wedge \\ \forall w (P(\underline{x}, w) \rightarrow w \neq z) \end{array} \right) \right).$$

The rewriting says that the set of P-facts is not empty, and that for every N-fact $N(\underline{c}, a)$, there exists a P-block in which a does not occur. □

Concerning the practical impact of Theorem 4.3, one may ask whether acyclicity of attack graphs is a strong requirement in practice. We therefore end this section with queries on a concrete database schema of four binary relations, admitting meaningful queries with and without consistent first-order rewriting.

Example 4.6. A poll has resulted in facts of the form Likes($\underline{p,t}$), indicating that person p has liked town t. The database schema contains three other binary relations between persons and towns: Born($\underline{p}, t$), Lives($\underline{p}, t$), and Mayor($\underline{t}, p$) mean, respectively, "p was born in t," "p currently lives in t," and "p is the mayor of t."

The following queries are canonical queries with a cyclic attack graph. The query q_1, for example, asks whether there exist towns whose mayor does not live in the town.

$$\begin{aligned} q_1 &= \{\text{Mayor}(\underline{t}, p), \neg\text{Lives}(\underline{p}, t)\} \\ q_2 &= \{\text{Likes}(\underline{p,t}), \neg\text{Lives}(\underline{p}, t), \neg\text{Mayor}(\underline{t}, p)\} \end{aligned}$$

The following queries have an acyclic attack graph, and thus have a consistent first-order rewriting. The query q_a, for example, asks whether people stay in a town which is not their birth town and which they do not like.

$$\begin{aligned} q_a &= \{\text{Lives}(\underline{p}, t), \neg\text{Born}(\underline{p}, t), \neg\text{Likes}(\underline{p,t})\} \\ q_b &= \{\text{Likes}(\underline{p,t}), \neg\text{Born}(\underline{p}, t), \neg\text{Lives}(\underline{p}, t)\} \end{aligned}$$

The attack graph of q_a contains one attack, which goes from Lives to Likes (we have Lives$|t \overset{q_a}{\rightsquigarrow} t$). The attack graph of q_b contains two attacks, both ending in Likes (we have Born$|t \overset{q_b}{\rightsquigarrow} t$ and Lives$|t \overset{q_b}{\rightsquigarrow} t$).

4.3 Properties of the Attack Relation

We show three properties of the attack relation $\overset{q}{\rightsquigarrow}$ that will be used later on. The following two lemmas express how attacks on variables carry over to atoms, and vice versa.

Lemma 4.7. *Let q be a query in* sjfBCQ$^\neg$, *and let $F \in q^+ \cup q^-$. If $F|w \overset{q}{\rightsquigarrow} u$ for some $w \in$* vars(F) *and $u \in$* vars(q), *then for every $P \in q^+ \setminus \{F\}$ such that $u \in$* vars(P), *there exists $x \in$* key(P) *such that $F|w \overset{q}{\rightsquigarrow} x$ (and thus $F \overset{q}{\rightsquigarrow} P$).*

Lemma 4.8. *Let q be a query in* sjfBCQ$^\neg$. *Let $F \in q^+ \cup q^-$ and $P \in q^+$ such that $F \neq P$. If $F \overset{q}{\rightsquigarrow} P$, then $F \overset{q}{\rightsquigarrow} u$ for every $u \in$* vars(P) $\setminus F^{\oplus,q}$.

The following lemma will allow us to conclude that under the restriction that negation is weakly-guarded, every cyclic attack graph has a cycle of length two.

Lemma 4.9. *Let q be a query in* sjfBCQ$^\neg$ *with weakly-guarded negation. For all $F, G, H \in q^+ \cup q^-$, if $F \overset{q}{\rightsquigarrow} G$ and $G \overset{q}{\rightsquigarrow} H$, then either $F \overset{q}{\rightsquigarrow} H$ or $G \overset{q}{\rightsquigarrow} F$.*

5 CYCLIC ATTACK GRAPHS

In this section we prove the first item of Theorem 4.3, which is expressed by the following lemma.

Lemma 5.1. *Let q be a query in* sjfBCQ$^\neg$ *with weakly-guarded negation. If the attack graph of q is cyclic, then* CERTAINTY(q) *is* **L**-*hard.*

In Section 5.1, we show that three simple canonical queries have no consistent first-order rewriting. These results are generalized in Section 5.2. Finally, the proof of Lemma 5.1 is given in Section 5.3.

5.1 Simple Queries Without Consistent First-Order Rewriting

To show that CERTAINTY(q) is not in **FO** for some query q in sjfBCQ$^\neg$, we will use first-order reductions from consistent query answering for three "canonical" queries, containing zero, one, and two negated atoms respectively.

$$\begin{aligned} q_0 &= \{R(\underline{x}, y), S(\underline{y}, x)\} \\ q_1 &= \{R(\underline{x}, y), \neg S(\underline{y}, x)\} \\ q_2 &= \{R(\underline{x,y}), \neg S(\underline{x}, y), \neg T(\underline{y}, x)\} \end{aligned}$$

L-hardness of CERTAINTY(q_0) is shown in [19, Lemma 4.2]. For query q_1, we show that CERTAINTY(q_1) is as hard as the Bipartite Perfect Matching (BPM) problem, which is known to be **NL**-hard [7].

Lemma 5.2. CERTAINTY(q_1) *is* **NL**-*hard for the Boolean query $q_1 = \{R(\underline{x}, y), \neg S(\underline{y}, x)\}$.*

Proof. We will show a first-order reduction from BPM to the complement of CERTAINTY(q_1). In BPM, we are given a bipartite graph G with m vertices on each side, and we ask whether G has a perfect matching (i.e., a matching of size m). BPM is known to be **NL**-hard [7]. Since **NL** is closed under complement, this implies that CERTAINTY(q_1) is **NL**-hard.

Given a bipartite graph $G = (A, B, E)$, we construct a database **db** as follows: for every edge $\{a, b\} \in E$, where $a \in A, b \in B$, the database **db** contains the facts $R(\underline{a}, b)$ and $S(\underline{b}, a)$. The database is obviously computable in **FO**. We will show that G has a perfect matching if and only if there exists a repair of **db** that falsifies q_1.

Assume that G has a perfect matching M. Then, we construct a repair **r** as follows. For every $a \in A$, if $\{a, b\}$ is the unique edge in M adjacent to a, we include in **r** the fact $R(\underline{a}, b)$. Similarly, for every $b \in B$, if $\{a, b\}$ is the unique edge in M adjacent to b, we include in **r** the fact $S(\underline{b}, a)$. It is easy to verify that **r** falsifies q_1, since if $R(\underline{a}, b) \in \mathbf{r}$, then $S(\underline{b}, a) \in \mathbf{r}$ as well.

For the opposite direction, assume that **r** is a repair of **db** that falsifies q_1. Observe that such a repair must satisfy the formula $\neg q_1 \equiv \forall x \forall y (R(\underline{x}, y) \rightarrow S(\underline{y}, x))$. We construct a matching M as follows: if $R(\underline{a}, b) \in \mathbf{r}$, we add to M the edge $\{a, b\}$. It is easy to see that M is a matching; indeed, it is not possible that $R(\underline{a}, b), R(\underline{a'}, b) \in \mathbf{r}$ and $a \neq a'$, since that would imply that $S(\underline{b}, a), S(\underline{b}, a') \in \mathbf{r}$, a contradiction. It is also a matching of size m, since every one of the m vertices in A is matched. □

The proof of the following lemma is in the Appendix.

Lemma 5.3. *CERTAINTY(q_2) is **L**-hard for the Boolean query* $q_2 = \{R(\underline{x, y}), \neg S(\underline{x}, y), \neg T(\underline{y}, x)\}$.

5.2 Attack Cycles of Length Two

The following helping lemma implies that if CERTAINTY(q) is hard for a complexity class under first-order reductions, then the problem remains hard if one or more negated atoms are added to q. The proof is straightforward and can be found in the Appendix.

Lemma 5.4. *Let q be a query in* sjfBCQ$^{\neg}$. *For every $q' \subseteq q$ such that $q^+ \subseteq q'$, there exists a first-order reduction from* CERTAINTY(q') *to* CERTAINTY(q).

We now aim to show that if the attack graph of a query $q \in$ sjfBCQ$^{\neg}$ contains a cycle of length two, then CERTAINTY(q) is not in **FO**. Three cases are distinguished, depending on whether the cycle contains zero, one, or two negated atoms. In the case of two negated atoms (and only in this case), the assumption of weakly-guarded negation is needed.

Lemma 5.5. *Let q be a query in* sjfBCQ$^{\neg}$. *Suppose that there exist two atoms $F, G \in q^+$ such that $F \overset{q}{\rightsquigarrow} G \overset{q}{\rightsquigarrow} F$. Then,* CERTAINTY($q$) *is **L**-hard (and thus not in **FO**).*

Proof. It can be seen that $F \overset{q^+}{\rightsquigarrow} G \overset{q^+}{\rightsquigarrow} F$. For sjfBCQ$^{\neg}$ queries without negated atoms, the attack graph defined in Section 3 is identical to the construct with the same name in [19]. **L**-hardness of CERTAINTY(q) then follows by [19, Lemma 4.3] and Lemma 5.4. □

Lemma 5.6. *Let q be a query in* sjfBCQ$^{\neg}$. *Suppose that there exist two atoms $F \in q^+$, $G \in q^-$ such that $F \overset{q}{\rightsquigarrow} G \overset{q}{\rightsquigarrow} F$. Then,* CERTAINTY($q$) *is **NL**-hard (and thus not in **FO**).*

Proof. We will show a first-order reduction from the problem CERTAINTY(q_1) with $q_1 = \{R(\underline{x}, y), \neg S(\underline{y}, x)\}$, which is **NL**-hard by Lemma 5.2.

Since $F \overset{q}{\rightsquigarrow} G$, there exists $v_F \in \mathrm{vars}(F), u \in \mathrm{key}(G)$ such that $F|v_F \overset{q}{\rightsquigarrow} u$. Similarly, there exists $v_G \in \mathrm{vars}(G), u' \in \mathrm{key}(F)$ such that $G|v_G \overset{q}{\rightsquigarrow} u'$. For all $a \in \mathrm{type}(x)$ and $b \in \mathrm{type}(y)$, define Θ_b^a as the following valuation over vars(q). For every $w \in \mathrm{vars}(q)$,

$$\Theta_b^a(w) = \begin{cases} a & \text{if } G|v_G \overset{q}{\rightsquigarrow} w \text{ and } F|v_F \overset{q}{\not\rightsquigarrow} w \\ b & \text{if } F|v_F \overset{q}{\rightsquigarrow} w \text{ and } G|v_G \overset{q}{\not\rightsquigarrow} w \\ \langle a, b\rangle & \text{if } F|v_F \overset{q}{\rightsquigarrow} w \text{ and } G|v_G \overset{q}{\rightsquigarrow} w \\ \bot & \text{otherwise} \end{cases}$$

Sublemma 5.1. *For every $H \in q^+ \setminus \{F\}$, for all $a, a' \in \mathrm{type}(x)$ and $b, b' \in \mathrm{type}(y)$, $\{\Theta_b^a(H), \Theta_{b'}^{a'}(H)\}$ is consistent.*

Proof. Assume $\Theta_b^a(H)$ and $\Theta_{b'}^{a'}(H)$ are key-equal. We need to show $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a = a'$ and $b = b'$. Trivial.

Case $a = a'$ and $b \neq b'$. Then, $F|v_F \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathrm{key}(H)$. By Lemma 4.7, $F|v_F \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathrm{vars}(H)$ and, consequently, $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a \neq a'$ and $b = b'$. Symmetric to the previous case.

Case $a \neq a'$ and $b \neq b'$. Then $F|v_F \overset{q}{\not\rightsquigarrow} w$ and $G|v_G \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathrm{key}(H)$. By Lemma 4.7, $F|v_F \overset{q}{\not\rightsquigarrow} w$ and $G|v_G \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathrm{vars}(H)$ and, consequently, $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

This concludes the proof of Sublemma 5.1. □

Sublemma 5.2. *For all $a, a' \in \mathrm{type}(x)$ and $b, b' \in \mathrm{type}(y)$,*

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

Consequently, the set $\{R(\underline{a}, b), R(\underline{a'}, b')\}$ is consistent if and only if $\{\Theta_b^a(F), \Theta_{b'}^{a'}(F)\}$ is consistent.

Proof. $\boxed{1 \Longrightarrow}$ $G|v_G \overset{q}{\rightsquigarrow} u'$ for some $u' \in \mathrm{key}(F)$. $\boxed{1 \Longleftarrow}$ If $w \in \mathrm{key}(F)$, then $F|v_F \overset{q}{\not\rightsquigarrow} w$. $\boxed{2 \Longrightarrow}$ Since equal facts are key-equal, the equality $a = a'$ follows from $1 \Longrightarrow$. From $F|v_F \overset{q}{\rightsquigarrow} v_F$ with $v_F \in \mathrm{vars}(F)$, it follows $b = b'$. $\boxed{2 \Longleftarrow}$ Trivial. This concludes the proof of Sublemma 5.2. □

Sublemma 5.3. *For all $a, a' \in \mathrm{type}(x)$ and $b, b' \in \mathrm{type}(y)$,*

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

Consequently, the set $\{S(\underline{b}, a), S(\underline{b'}, a')\}$ is consistent if and only if $\{\Theta_b^a(G), \Theta_{b'}^{a'}(G)\}$ is consistent.

Proof. This can be proved along the same lines as the proof of Sublemma 5.2. □

For every database **db** that is input to CERTAINTY(q_1), we define $f(\mathbf{db})$ as follows:

- if **db** contains $R(\underline{a}, b)$, then $f(\mathbf{db})$ includes $\Theta_b^a(q^+)$;
- if **db** contains $S(\underline{b}, a)$, then $f(\mathbf{db})$ contains $\Theta_b^a(G)$.

Clearly, f is computable in **FO**. Note incidentally that whenever T is a relation name in $q^- \setminus \{G\}$, then $f(\mathbf{db})$ contains no T-facts. Let

$$g(\mathbf{db}) = (f(\mathbf{db}) \setminus \{\Theta_b^a(F) \mid R(\underline{a}, b) \in \mathbf{db}\}) \setminus \{\Theta_b^a(G) \mid S(\underline{b}, a) \in \mathbf{db}\}.$$

By Sublemmas 5.1, 5.2, and 5.3,

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

Let $\mathbf{db}$ be a database that is input to CERTAINTY(q_1). Let $\mathbf{r}$ be a repair of $\mathbf{db}$. It suffices to show that the following are equivalent:

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

$\boxed{1 \Longrightarrow 2}$ We can assume $R(\underline{a}, b) \in \mathbf{r}$ such that $S(\underline{b}, a) \notin \mathbf{r}$. Then $\Theta_b^a(q^+) \subseteq f(\mathbf{r})$. Assume, towards a contradiction, $\Theta_b^a(G) \in f(\mathbf{r})$. Then there must exist $S(\underline{b'}, a') \in \mathbf{r}$ such that $\Theta_{b'}^{a'}(G) = \Theta_b^a(G)$. Then, from the second item in Sublemma 5.3, $a = a'$ and $b = b'$, hence $S(\underline{b}, a) \in \mathbf{r}$, a contradiction. We conclude by contradiction $\Theta_b^a(G) \notin f(\mathbf{r})$. It is now correct to conclude $f(\mathbf{r}) \cup g(\mathbf{db}) \models q$.

$\boxed{2 \Longrightarrow 1}$ Let θ be a valuation over vars(q) such that $\theta(q^+) \subseteq f(\mathbf{r}) \cup g(\mathbf{db})$ and $\theta(G) \notin f(\mathbf{r})$. By our construction, we can assume $R(\underline{a}, b) \in \mathbf{r}$ such that $\theta(F) = \Theta_b^a(F)$. By item 2 in Sublemma 5.2, $R(\underline{a}, b)$ is unique.

We will show that for every $x \in \mathsf{vars}(G)$, $\theta(x) = \Theta_b^a(x)$. It follows that $\theta(G) = \Theta_b^a(G)$. It is then correct to conclude that $S(\underline{b}, a) \notin \mathbf{r}$, or else $\Theta_b^a(G) = \theta(G) \in f(\mathbf{r})$, a contradiction.

We can assume a sequence

$$(v_0, P_1, v_1, P_2, v_2 \ldots, P_{\ell-1}, v_{\ell-1}, P_\ell, v_\ell)$$

such that $(v_0, v_1, v_2 \ldots, v_{\ell-1}, v_\ell)$ is a witness for $G|v_G \overset{q}{\rightsquigarrow} u'$ (thus $v_0 = v_G$ and $v_\ell = u'$), and for all $i \in \{1, \ldots, \ell\}$, $P_i \in q^+$ such that $v_{i-1}, v_i \in \mathsf{vars}(P_i)$. Obviously, for every $i \in \{0, \ldots, \ell\}$, we have $G|v_G \overset{q}{\rightsquigarrow} v_i$.

We show by induction on decreasing i that for all $i \in \{0, \ldots, \ell\}$, $\theta(v_i) \in \{a, \langle a, b \rangle\}$. For the basis of the induction (i.e., $i = \ell$), we have $v_\ell = u' \in \mathsf{key}(F)$ and, since $\theta(F) = \Theta_b^a(F)$, $\theta(u') = \Theta_b^a(u') \in \{a, \langle a, b \rangle\}$. Note incidentally that from $u' \in \mathsf{key}(F)$ and thus $F|v_F \overset{q}{\not\rightsquigarrow} u'$, it follows $\theta(u') = a$. For the induction step (i.e., $i \rightarrow i-1$), the induction hypothesis is $\theta(v_i) \in \{a, \langle a, b \rangle\}$. We can assume an R-fact $R(\underline{a'}, b') \in \mathbf{r}$ such that $\theta(P_i) = \Theta_{b'}^{a'}(P_i) \in \Theta_{b'}^{a'}(q^+)$, and thus $\theta(v_i) = \Theta_{b'}^{a'}(v_i) \in \{a', \langle a', b' \rangle\}$. Since $\theta(v_i) \in \{a, \langle a, b \rangle\}$, it must be the case that $a' = a$. From $R(\underline{a}, b), R(\underline{a}, b') \in \mathbf{r}$, it follows $b' = b$ (because $\mathbf{r}$ is consistent). From $\theta(P_i) = \Theta_b^a(P_i)$, it follows $\theta(v_{i-1}) \in \{a, \langle a, b \rangle\}$, which concludes the induction step. In particular, $\theta(v_G) \in \{a, \langle a, b \rangle\}$.

Likewise, we can assume a sequence

$$(v_0, P_1, v_1, P_2, v_2 \ldots, P_{m-1}, v_{m-1}, P_m, v_m)$$

such that $(v_0, v_1, v_2 \ldots, v_{m-1}, v_m)$ is a witness for $F|v_F \overset{q}{\rightsquigarrow} u$ (thus $v_0 = v_F$ and $v_m = u$), and for all $i \in \{1, \ldots, m\}$, $P_i \in q^+$ such that $v_{i-1}, v_i \in \mathsf{vars}(P_i)$. Obviously, for every $i \in \{0, \ldots, m\}$, we have $F|v_F \overset{q}{\rightsquigarrow} v_i$.

We show by induction on increasing i that for all $i \in \{0, \ldots, m\}$, there exists $a_i \in \mathsf{type}(x)$ such that $\theta(v_i) \in \{b, \langle a_i, b \rangle\}$. For the basis of the induction (i.e., $i = 0$), we have $v_0 = v_F \in \mathsf{vars}(F)$ and, since $\theta(F) = \Theta_b^a(F)$, $\theta(v_F) = \Theta_b^a(v_F) \in \{b, \langle a, b \rangle\}$. For the induction step (i.e., $i \rightarrow i+1$), the induction hypothesis is $\theta(v_i) \in \{b, \langle a_i, b \rangle\}$ for some $a_i \in \mathsf{type}(x)$. We can assume an R-fact $R(\underline{a'}, b') \in \mathbf{r}$ such that $\theta(P_{i+1}) = \Theta_{b'}^{a'}(P_{i+1}) \in \Theta_{b'}^{a'}(q^+)$, and thus $\theta(v_i) = \Theta_{b'}^{a'}(v_i) \in \{b', \langle a', b' \rangle\}$. Since $\theta(v_i) \in \{b, \langle a_i, b \rangle\}$, it must be the case that $b' = b$. From $\theta(P_{i+1}) = \Theta_b^{a'}(P_{i+1})$, it follows $\theta(v_{i+1}) \in \{b, \langle a', b \rangle\}$, which concludes the induction step. In particular, $\theta(u) \in \{b, \langle a_m, b \rangle\}$ for some $a_m \in \mathsf{type}(x)$. Further, from $u \in \mathsf{key}(G)$, it follows $G|v_G \overset{q}{\not\rightsquigarrow} u$, hence $\theta(u) = b$.

Let x be an arbitrary variable in vars(G). We can assume an atom $P_0 \in q^+$ such that $x, v_G \in \mathsf{vars}(P_0)$. We can assume $R(\underline{a'}, b')$ such that $\theta(P_0) = \Theta_{b'}^{a'}(P_0)$, hence $\theta(v_G) = \Theta_{b'}^{a'}(v_G)$ and $\theta(x) = \Theta_{b'}^{a'}(x)$. From $\theta(v_G) \in \{a, \langle a, b \rangle\}$, it follows $a = a'$. Likewise, we can assume an atom $P_1 \in q^+$ such that $x, u \in \mathsf{vars}(P_1)$. We can assume $R(\underline{a''}, b'')$ such that $\theta(P_1) = \Theta_{b''}^{a''}(P_1)$, hence $\theta(u) = \Theta_{b''}^{a''}(u)$ and $\theta(x) = \Theta_{b''}^{a''}(x)$. From $\theta(u) = b$, it follows $b = b''$. Form $\theta(x) = \Theta_{b'}^{a}(x) = \Theta_b^{a''}(x)$, it is correct to conclude that $\theta(x) = \Theta_b^a(x)$. This concludes the proof. □

The proof of the following lemma, which can be found in the Appendix, shows **L**-hardness by a first-order reduction from the problem CERTAINTY(q_2), which is **L**-hard by Lemma 5.3. Note that, as announced in the beginning of this section, the assumption of weakly-guarded negation is made in the lemma's hypothesis. In fact, Example 7.1 in Section 7 shows that without this assumption, the lemma fails to hold true.

Lemma 5.7. *Let q be a query in* sjfBCQ$^\neg$ *with weakly-guarded negation. Suppose that there exist two atoms $F, G \in q^-$ such that $F \overset{q}{\rightsquigarrow} G \overset{q}{\rightsquigarrow} F$. Then,* CERTAINTY$(q)$ *is* **L***-hard (and thus not in* **FO***).*

5.3 Proof of Lemma 5.1

The main lemma of this section now has a short proof.

Proof of Lemma 5.1. Assume that the attack graph of q is cyclic. By Lemma 4.9, the attack graph of q contains a cycle of length two. Depending on the number of atoms of q^- in the cycle, we use Lemma 5.5, Lemma 5.6, or Lemma 5.7 to conclude that the problem CERTAINTY(q) is not in **FO**. □

6 ACYCLIC ATTACK GRAPHS

In this section, we show the second item of Theorem 4.3, which is expressed by the following lemma.

Lemma 6.1. *Let q be a query in* sjfBCQ$^\neg$ *with weakly-guarded negation such that the attack graph of q is acyclic. Then, the problem* CERTAINTY(q) *is in* **FO** *and a consistent first-order rewriting for q can be effectively constructed.*

Algorithm 1 solves the problem CERTAINTY(q) under the conditions of Lemma 6.1. In Section 6.1, we show that if a query q in sjfBCQ$^\neg$ contains a negated atom $\neg N$ such that key$(N) = \emptyset$, then the problem CERTAINTY(q) can be simplified by eliminating $\neg N$ from q. Then, in Section 6.2, we show that unattacked variables can be *reified*, that is, treated as constants in CERTAINTY(q). Reification and elimination of atoms with variable-free primary keys are the essential cruxes in the proof of Lemma 6.1, which is given in Section 6.3.

Algorithm 1: Outline of an algorithm for CERTAINTY(q) with data complexity in **FO**.

FUNCTION **IsCertain**(q, **db**)
INPUT : $q \in$ sjfBCQ$^\neg$ with weakly-guarded negation and acyclic attack graph; database **db**
OUTPUT : Is q true in every repair of **db**?

if every atom of q is all-key
then (return **true** if **db** satisfies q; otherwise return **false**)
else pick an unattacked, non-all-key atom $F \in q^+ \cup q^-$
  case key(F) $\neq \emptyset$
  return **true** if
    there exists a valuation θ over key(F)
    such that **IsCertain**($\theta(q)$, **db**);
  otherwise return **false**

  case key(F) $= \emptyset$; let $F = R(\underline{\vec{a}}, \vec{y})$, vars($\vec{a}$) $= \emptyset$
    subcase $F \in q^-$; let $q' = q \setminus \{\neg F\}$
    return **true** if
      IsCertain(q', **db**) and for all $\vec{b}$ such
      that $R(\underline{\vec{a}}, \vec{b}) \in$ **db**, it is the case that
      IsCertain($q' \cup \{\neg E(\underline{\vec{y}})\}$, **db** $\cup \{E(\underline{\vec{b}})\}$);
      /* E is a fresh, all-key relation name */
    otherwise return **false**

    subcase $F \in q^+$; let $q' = q \setminus \{F\}$
    return **true** if
      there exists $\vec{b}$ such that $R(\underline{\vec{a}}, \vec{b}) \in$ **db** and
      for all $\vec{b}'$ such that $R(\underline{\vec{a}}, \vec{b}') \in$ **db**, there
      exists a valuation θ over vars($\vec{y}$) such
      that $\theta(\vec{y}) = \vec{b}'$ and **IsCertain**($\theta(q')$, **db**);
    otherwise return **false**

6.1 Variable-Free Primary Keys

We give two lemmas that explain how to deal with negated atoms $\neg N$ such that key(N) $= \emptyset$. Notice that such N can never have an incoming attack. We first treat the case where N contains no variables.

Lemma 6.2. *Let q be a query in* sjfBCQ$^\neg$. *Let $N \in q^-$ such that* vars(N) $= \emptyset$. *For every database* **db**, *q is true in every repair of* **db** *if and only if both $N \notin$* **db** *and $q \setminus \{\neg N\}$ is true in very repair of* **db**.

We next focus on eliminating negated atoms of the form $\neg R(\underline{a}, \vec{y})$ where $\vec{a}$ is variable-free, but $\vec{y}$ is not. This elimination requires disequalities between variables and constants. We therefore extend queries q with disequalities of the form $\vec{v} \neq \vec{c}$, where $\vec{v}$ is a sequence of distinct variables and $\vec{c}$ a sequence of constants. For the following definition, note that queries are modeled as sets containing atoms and disequalities (thus, the union acts as a logical AND).

Definition 6.3. We denote by sjfBCQ$^{\neg\neq}$ the set of queries that can be written as a disjoint union $q \cup C$ with $q \in$ sjfBCQ$^\neg$ and C is a set of disequalities of the form $\vec{v} \neq \vec{c}$, where $\vec{v}$ is a sequence of distinct variables in vars(q) and $\vec{c}$ is a sequence of constants. The semantics is standard: $q \cup C$ is satisfied by a database **db** if there exists a valuation θ over vars(q) such for every $P \in q^+$, $\theta(P) \in$ **db**, for every $N \in q^-$, $\theta(N) \notin$ **db**, and for every $\langle v_1, \dots, v_\ell \rangle \neq \langle c_1, \dots, c_\ell \rangle$ in C, there exists $i \in \{1, \dots, \ell\}$ such that $\theta(v_i) \neq c_i$.

We say that negation in $q \cup C$ is weakly-guarded if negation in q is weakly-guarded and for every disequality $\vec{v} \neq \vec{c}$ in C, for every $v_1, v_2 \in$ vars($\vec{v}$), there exists an atom $P \in q^+$ such that $v_1, v_2 \in$ vars(P). □

Example 6.4. $\{R(\underline{x}, y, z), \neg N(\underline{y})\} \cup \{xz \neq ab\}$ denotes the query $\exists x \exists y \exists z (R(\underline{x}, y, z) \wedge \neg N(\underline{y}) \wedge \neg(x = a \wedge z = b))$.

Whenever we write $q \cup C$, it is understood that q contains no disequalities, and that C contains only disequalities.

Lemma 6.5. *Let q be a query in* sjfBCQ$^\neg$. *Let $N \in q^-$ such that* key(N) $= \emptyset$ *and* vars(N) $\neq \emptyset$. *Let $\vec{y}$ be a sequence of distinct variables such that* vars($\vec{y}$) $=$ vars(N). *For every database* **db**, *q is true in every repair of* **db** *if and only if the following two conditions are satisfied:*

(1) *$q \setminus \{\neg N\}$ is true in every repair of* **db**; *and*
(2) *for every valuation θ over* vars(N) *such that $\theta(N) \in$* **db**, *the query $(q \setminus \{\neg N\}) \cup \{\vec{y} \neq \theta(\vec{y})\}$ is true in every repair of* **db**.

Proof. Easy. Note that if a database **db** falsifies the singleton conjunctive query $\{N\}$, then the condition 2 is necessarily satisfied; in that case (and only in that case), condition 1 is relevant. □

Notice that we require vars($\vec{y}$) $\neq \emptyset$ in the statement of Lemma 6.5. If we allow vars($\vec{y}$) $= \emptyset$ and admit that the empty valuation is the only valuation over the empty set of variables, then, under the restriction vars($\vec{y}$) $= \emptyset$, condition 2 is equivalent to requiring $N \notin$ **db**. Thus, if we allow vars($\vec{y}$) $= \emptyset$ in Lemma 6.5, then we obtain a generalization of Lemma 6.2.

Lemma 6.5 reduces consistent query answering from a query in sjfBCQ$^\neg$ to a query in sjfBCQ$^{\neg\neq}$. Since attack graphs have not been designed for queries in sjfBCQ$^{\neg\neq}$, we have to be able to translate back from sjfBCQ$^{\neg\neq}$ to sjfBCQ$^\neg$. The following lemma states that this translation is indeed possible.

Lemma 6.6. *Let $q \cup C \in$* sjfBCQ$^{\neg\neq}$ *such that C contains $\vec{v} \neq \vec{c}$. Let $C' = C \setminus \{\vec{v} \neq \vec{c}\}$. Then, there exists a first-order reduction from* CERTAINTY($q \cup C$) *to* CERTAINTY($q \cup \{\neg E(\underline{\vec{v}})\} \cup C'$), *where E is a fresh relation name that is all-key.*

Proof. Let **db** be a database that is input to CERTAINTY($q \cup C$). Let g(**db**) $=$ **db** $\cup \{E(\underline{\vec{c}})\}$. Clearly, g(**db**) is computable in **FO**. It is easy to see that every repair of **db** satisfies $q \cup C$ if and only if every repair of g(**db**) satisfies $q \cup \{\neg E(\underline{\vec{v}})\} \cup C'$. □

6.2 Reifiable Variables

The following definition introduces the notion of reifiable variables. We then show a lemma, Lemma 6.8, whose corollary will be that unattacked variables are reifiable, provided that negation is weakly-guarded.

Definition 6.7. Let q be a query in sjfBCQ$^\neg$. Let $X = \{x_1, \dots, x_\ell\}$ be a non-empty subset of vars(q). We say that X is *reifiable* in q if for every database **db** such that q is true in every repair of **db**, there exist constants $c_1, \dots, c_\ell$ (which depend on **db**) such that $q_{[\langle x_1, \dots, x_\ell \rangle \mapsto \langle c_1, \dots, c_\ell \rangle]}$ is true in every repair of **db**. □

Reifiable variables will turn out to be essential in the proof of Lemma 6.1, in the following manner. Let $\vec{x}$ be a sequence of distinct variables occurring in some query q of sjfBCQ$^{\neg}$. We aim to treat the variables of $\vec{x}$ as constants: we denote by $q(\vec{x})$ the query q in which $\vec{x}$ is seen as a sequence of constants, which behave like Skolem constants. Now assume that CERTAINTY($q(\vec{x})$) is in **FO** and that φ is a consistent first-order rewriting for $q(\vec{x})$. Since $\vec{x}$ was seen as a sequence of constants, these constants will also occur in φ. Then, if vars($\vec{x}$) is reifiable in q, it will be the case that $\exists\vec{x}\varphi$ is a consistent first-order rewriting for q.

Lemma 6.8. *Let q be a query in* sjfBCQ$^{\neg}$ *with weakly-guarded negation and let $X \subseteq$ vars(q). Let $G \in q^+ \cup q^-$ such that for every $x \in X$, $G \overset{q}{\not\rightsquigarrow} x$. Let $\mathbf{r}$ be a consistent database. Let A, B be key-equal G-facts such that $A \in \mathbf{r}$ is key-relevant for q in $\mathbf{r}$. Let $\mathbf{r}_B = (\mathbf{r} \setminus \{A\}) \cup \{B\}$. For every valuation ζ over X, if $\mathbf{r}_B \models \zeta(q)$, then $\mathbf{r} \models \zeta(q)$.*

Proof. The proof is trivial if $A = B$. We next consider the case $A \neq B$.

Assume a valuation ζ over X such that $\mathbf{r}_B \models \zeta(q)$. We can assume a valuation ζ^+ over vars(q) that extends ζ such that $\mathbf{r}_B \models \zeta^+(q)$. Since A is key-relevant for q in $\mathbf{r}$, we can assume a valuation μ over vars(q) such that $\mathbf{r} \models \mu(q)$ and $\mu(G) \sim A$. We distinguish two cases: $G \in q^-$ and $G \in q^+$.

Case that $G \in q^-$. Since $\mathbf{r} \models \mu(q)$, we have $\mu(G) \notin \mathbf{r}$ and, since $A \in \mathbf{r}$, $\mu(G) \neq A$. If $\zeta^+(G) \neq A$, then $\mathbf{r} \models \zeta^+(q)$ and the desired result holds. In what follows, we consider the case $\zeta^+(G) = A$. From $\zeta^+(G) = A \sim \mu(G)$, it follows that ζ^+ and μ agree on all variables of key(G). Since $\mathbf{r} \setminus \{A\} = \mathbf{r}_B \setminus \{B\}$ and $\mu(q^+) \subseteq \mathbf{r} \setminus \{A\}$ and $\zeta^+(q^+) \subseteq \mathbf{r}_B \setminus \{B\}$, it follows from [29, Lemma 4.3] that ζ^+ and μ agree on all variables of $G^{\oplus,q}$. Then, vars(G) $\nsubseteq G^{\oplus,q}$, or else $\mu(G) = \zeta^+(G) = A$, a contradiction.

We define κ as the valuation over vars(q) such that for every $P \in q^+$,

$$\kappa(P) = \begin{cases} \mu(P) & \text{if } G \overset{q}{\rightsquigarrow} P \\ \zeta^+(P) & \text{otherwise} \end{cases}$$

We show below that κ is well-defined, that $\mathbf{r} \models \kappa(q)$, and that $\kappa[X] = \zeta$. It is then correct to conclude $\mathbf{r} \models \zeta(q)$.

Proof that κ is well-defined. To show that κ is well-defined, assume $P_1, P_2 \in q^+$ and $v \in$ vars(P_1) $\cap$ vars(P_2) such that $G \overset{q}{\rightsquigarrow} P_1$ but $G \overset{q}{\not\rightsquigarrow} P_2$. From $G \in q^-$, it follows $P_1 \neq G \neq P_2$. By Lemma 4.7, from $G \overset{q}{\not\rightsquigarrow} P_2$ and $v \in$ vars(P_2), it follows $G \overset{q}{\not\rightsquigarrow} v$. By Lemma 4.8, from $G \overset{q}{\rightsquigarrow} P_1$, $v \in$ vars(P_1), and $G \overset{q}{\not\rightsquigarrow} v$, it follows $v \in G^{\oplus,q}$. Consequently, $\zeta^+(v) = \mu(v)$.

Proof that $\mathbf{r} \models \kappa(q)$. We have $\mathbf{r} \setminus \{A\} = \mathbf{r}_B \setminus \{B\}$, where A, B are G-facts. For every $P \in q^+$, we have $\kappa(P) \in \mathbf{r}$, because either $\kappa(P) = \mu(P) \in \mathbf{r} \setminus \{A\}$ or $\kappa(P) = \zeta^+(P) \in \mathbf{r}_B \setminus \{B\}$.

We show next that for every $N \in q^-$ (possibly $N = G$), either $\kappa(N) = \mu(N)$ or $\kappa(N) = \zeta^+(N)$. To this extent, let $N \in q^-$ and let w be an arbitrary variable in vars(N). We distinguish two cases.

- Case $G \overset{q}{\rightsquigarrow} u$ for some $u \in$ vars(N). Since negation is weakly-guarded, we can assume $P \in q^+$ such that $u, w \in$ vars(P). It follows by Lemma 4.7 that $G \overset{q}{\rightsquigarrow} P$, hence $\kappa(P) = \mu(P)$. It follows $\kappa(w) = \mu(w)$.
- Case $G \overset{q}{\not\rightsquigarrow} u$ for all $u \in$ vars(N). Since negation is safe, we can assume $P \in q^+$ such that $w \in$ vars(P). If $G \overset{q}{\not\rightsquigarrow} P$, then $\kappa(P) = \zeta^+(P)$, and thus $\kappa(w) = \zeta^+(w)$. Assume next that $G \overset{q}{\rightsquigarrow} P$. Since $G \overset{q}{\not\rightsquigarrow} w$, it follows $w \in G^{\oplus,q}$ by Lemma 4.8. Since μ and ζ^+ agree on all variables of $G^{\oplus,q}$, it follows $\kappa(w) = \mu(w) = \zeta^+(w)$.

It is correct to conclude that for all $N \in q^-$, either $\kappa(N) = \mu(N)$ or $\kappa(N) = \zeta^+(N)$. Let $N \in q^- \setminus \{G\}$. Since $\mathbf{r}$ and $\mathbf{r}_B$ contain exactly the same set of N-facts, it follows that $\mu(N) \notin \mathbf{r}$ and $\zeta^+(N) \notin \mathbf{r}$, thus $\kappa(N) \notin \mathbf{r}$. So it remains to be shown that $\kappa(G) \notin \mathbf{r}$. Since vars($G$) $\nsubseteq G^{\oplus,q}$, we have $G \overset{q}{\rightsquigarrow} u$ for some $u \in$ vars(G) (the first case above), and thus $\kappa(G) = \mu(G) \notin \mathbf{r}$.

Proof that $\kappa(x) = \zeta(x)$ for every $x \in X$. This is obvious if $x \in G^{\oplus,q}$. Assume next that $x \notin G^{\oplus,q}$. We can assume an atom $P \in q^+$ such that $x \in$ vars(P). We have $G \overset{q}{\not\rightsquigarrow} P$, or else $G \overset{q}{\rightsquigarrow} x$ by Lemma 4.8 and thus $x \notin X$, a contradiction. It follows $\kappa(P) = \zeta^+(P)$, hence $\kappa(x) = \zeta(x)$.

Case that $G \in q^+$. This part of the proof can be found in the Appendix. The proof outline is similar to the previous case. □

Corollary 6.9. *Let q be a query in* sjfBCQ$^{\neg}$ *with weakly-guarded negation. Let $X \subseteq$ vars(q) such that for every $x \in X$, there exists no $G \in q^+ \cup q^-$ such that $G \overset{q}{\rightsquigarrow} x$. Then, X is reifiable in q.*

Proof. Let **db** be a database that is given as input to the problem CERTAINTY(q). For every repair $\mathbf{r}$ of **db**, define Reify($\mathbf{r}$) as the set of valuations over X such that Reify($\mathbf{r}$) contains ζ if $\mathbf{r} \models \zeta(q)$. The hypothesis is that every repair of **db** satisfies q, thus for every repair $\mathbf{r}$ of **db**, Reify($\mathbf{r}$) $\neq \emptyset$. We can assume the existence of a repair $\mathbf{r}$ of **db** such that Reify($\mathbf{r}$) is minimal (i.e., there exists no repair $\mathbf{s}$ of **db** such that Reify($\mathbf{s}$) $\subsetneq$ Reify($\mathbf{r}$)).

Let $\mathbf{s}$ be any repair of **db**. Construct a maximal sequence

$$\mathbf{r}_0,\quad A_1, B_1, \mathbf{r}_1,\quad A_2, B_2, \mathbf{r}_2,\quad \ldots,\quad A_\ell, B_\ell, \mathbf{r}_\ell,$$

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

(1) $A_i \in \mathbf{r}$ and A_i is key-relevant for $\mathbf{r}_{i-1}$ in q;
(2) $B_i \in \mathbf{s}$ such that $B_i \sim A_i$ and $B_i \neq A_i$;
(3) $\mathbf{r}_i = (\mathbf{r}_{i-1} \setminus \{A_i\}) \cup \{B_i\}$.

Then, for every $A \in \mathbf{r}_\ell$ such that A is key-relevant for q in $\mathbf{r}_\ell$, we have $A \in \mathbf{s}$. Consequently, Reify($\mathbf{r}_\ell$) $\subseteq$ Reify($\mathbf{s}$). By Lemma 6.8, Reify($\mathbf{r}_\ell$) $\subseteq$ Reify($\mathbf{r}$). Since Reify($\mathbf{r}$) is minimal, Reify($\mathbf{r}_\ell$) = Reify($\mathbf{r}$). It follows Reify($\mathbf{r}$) $\subseteq$ Reify($\mathbf{s}$). The desired result follows, since Reify($\mathbf{r}$) $\neq \emptyset$ and $\mathbf{s}$ is an arbitrary repair of **db**. □

6.3 Proof of Lemma 6.1

We first present a lemma which tells us that acyclicity of attack graphs and weakly-guardedness of negation are preserved when we replace a variable with a constant in a query.

Lemma 6.10. *Let q be a query in* sjfBCQ$^{\neg}$. *Let $x \in$ vars(q) and let c be a constant. Then,*

$$\begin{array}{l}
\exists x(S(\underline{x}) \\
\wedge \forall z(N_3(\underline{c},z) \rightarrow \exists x(S(\underline{x}) \wedge x \neq z)) \\
\wedge \forall z \left(N_2(\underline{c},z) \rightarrow \left(\begin{array}{l} \exists x(S(\underline{x}) \wedge x \neq z) \\ \wedge \forall z'(N_3(\underline{c},z') \rightarrow \exists x(S(\underline{x}) \wedge x \neq z \wedge x \neq z')) \end{array} \right) \right) \\
\wedge \forall z \left(N_1(\underline{c},z) \rightarrow \left(\begin{array}{l} \exists x(S(\underline{x}) \wedge x \neq z) \\ \wedge \forall z'(N_3(\underline{c},z') \rightarrow \exists x(S(\underline{x}) \wedge x \neq z \wedge x \neq z')) \\ \wedge \forall z' \left(N_2(\underline{c},z') \rightarrow \left(\begin{array}{l} \exists x(S(\underline{x}) \wedge x \neq z \wedge x \neq z') \\ \wedge \forall z''(N_3(\underline{c},z'') \rightarrow \exists x(S(\underline{x}) \wedge x \neq z \wedge x \neq z' \wedge x \neq z'')) \end{array} \right) \right) \end{array} \right) \right) \Bigg)
\end{array}$$

Figure 2: Consistent first-order rewriting for $\{S(\underline{x}), \neg N_1(\underline{c},x), \neg N_2(\underline{c},x), \neg N_3(\underline{c},x)\}$**.**

(1) *for every* $F, G \in q^+ \cup q^-$, *if* $F_{[x\mapsto c]} \overset{q_{[x\mapsto c]}}{\rightsquigarrow} G_{[x\mapsto c]}$, *then* $F \overset{q}{\rightsquigarrow} G$; *and*
(2) *if negation in* q *is weakly-guarded, then negation in* $q_{[x\mapsto c]}$ *is weakly-guarded.*

We now give the proof of Lemma 6.1, which shows membership of $\mathsf{CERTAINTY}(q)$ in **FO** through the construction of a consistent first-order rewriting for q. This construction is also captured by Algorithm 1.

Proof of Lemma 6.1. For every $q \in \mathsf{sjfBCQ}^{\neg}$, define $\alpha(q)$ as the number of atoms in $q^+ \cup q^-$ that are not all-key. Assume that the attack graph of q is acyclic. We will show by induction on increasing $\alpha(q)$ that $\mathsf{CERTAINTY}(q)$ is in **FO**. For the induction basis, let $\alpha(q) = 0$. Then all atoms in q are all-key. If **db** is an input to $\mathsf{CERTAINTY}(q)$, then **db** is consistent and is its own unique repair. Trivially, every repair of **db** satisfies q if and only if **db** satisfies q. $\mathsf{CERTAINTY}(q)$ is thus in **FO**.

For the induction step, let $\alpha(q) \geq 1$. It can be easily seen that every all-key atom has zero outdegree in the attack graph of q. Since the attack graph of q is acyclic, there exists an atom $F \in q^+ \cup q^-$ such that F is not all-key and for every $G \in q^+ \cup q^-$, for every $x \in \mathsf{key}(F)$, $G \overset{q}{\not\rightsquigarrow} x$.

Let $\vec{x} = \langle x_1, \ldots, x_\ell \rangle$ be a sequence of distinct variables such that $\mathsf{vars}(\vec{x}) = \mathsf{key}(F)$. By Corollary 6.9, for every database **db**, the following are equivalent:

(1) every repair of **db** satisfies q; and
(2) there exists a sequence $\vec{c}$ of constants such that every repair of **db** satisfies $q_{[\vec{x}\mapsto\vec{c}]}$.

Let $\vec{c} = \langle c_1, \ldots, c_\ell \rangle$ be a sequence of constants such that for $i \in \{1, \ldots, \ell\}$, $c_i \in \mathsf{type}(x_i)$ and c_i does not occur in q. In the sequel, we show that $q_{[\vec{x}\mapsto\vec{c}]}$ has a consistent first-order rewriting φ. It is then correct to conclude that $\exists\vec{x}\varphi'$ is a first-order rewriting for q, where φ' is obtained from φ by replacing each occurrence of each c_i by x_i. This is tantamount to constructing a consistent first-order rewriting for the query $q(\vec{x})$ in which the variables of $\vec{x}$ are free and treated as constants.

Let $\vec{y}$ be a sequence of distinct variables such that $\mathsf{vars}(\vec{y}) = \mathsf{vars}(F) \setminus \mathsf{key}(F)$. Let $q' = q \setminus \{F, \neg F\}$. We distinguish two cases.

Case $F \in q^+$. Obviously, the following are equivalent:

(1) $q_{[\vec{x}\mapsto\vec{c}]}$ is true in every repair of **db**;
(2) there exists a sequence $\vec{d}$ of constants such that $F_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]}$ belongs to **db**, and for every F-fact A of **db** that is key-equal to $F_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]}$, there exists a sequence $\vec{e}$ of constants such that $F_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{e}]} = A$ and $q'_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{e}]}$ is true in every repair of **db**.

We have that $\alpha(q'_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]}) < \alpha(q)$. By Lemma 6.10, the attack graph of $q'_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]}$ remains acyclic, and negation in $q'_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]}$ is weakly-guarded. We have that $\mathsf{CERTAINTY}(q'_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]})$ is in **FO** by the induction hypothesis. Since the condition in item 2 can be tested in first-order logic, $\mathsf{CERTAINTY}(q_{[\vec{x}\mapsto\vec{c}]})$ is in **FO**. In case that $F_{[\vec{x}\mapsto\vec{c}]} = R(\underline{\vec{c}}, \vec{y})$, a consistent first-order rewriting for $q_{[\vec{x}\mapsto\vec{c}]}$ is $\exists\vec{y}R(\underline{\vec{c}}, \vec{y}) \wedge \forall\vec{y}(R(\underline{c}, \vec{y}) \rightarrow \psi)$, where ψ is a consistent first-order rewriting for $q'_{[\vec{x}\mapsto\vec{c}]}(\vec{y})$. The rewriting is slightly more complicated if the non-primary-key part of $F_{[\vec{x}\mapsto\vec{c}]}$ contains constants or double occurrences of the same variable.

Case $F \in q^-$. The following are equivalent:

(1) $q_{[\vec{x}\mapsto\vec{c}]}$ is true in every repair of **db**;
(2) $q'_{[\vec{x}\mapsto\vec{c}]}$ is true in every repair of **db**; and
  (a) *If* $\vec{y}$ *is the empty sequence:* $F_{[\vec{x}\mapsto\vec{c}]} \notin$ **db**.
  (b) *If* $\vec{y}$ *is not the empty sequence:* for every sequence $\vec{d}$ of constants such that $F_{[\vec{x}\vec{y}\mapsto\vec{c}\vec{d}]} \in$ **db**, it is the case that $q'_{[\vec{x}\mapsto\vec{c}]} \wedge \vec{y} \neq \vec{d}$ is true in every repair of **db**.

The correctness follows from Lemmas 6.2 and 6.5. By Lemma 6.10, the attack graph of $q'_{[\vec{x}\mapsto\vec{c}]}$ is acyclic, and negation in $q'_{[\vec{x}\mapsto\vec{c}]}$ is weakly-guarded.

Define $p := q'_{[\vec{x}\mapsto\vec{c}]} \cup \{\neg E(\underline{\vec{y}})\}$, where E is a fresh relation name that is all-key. It is easy to see that the attack graph of p is acyclic and that negation in p is weakly-guarded (because $\mathsf{vars}(\vec{y}) \subseteq \mathsf{vars}(F)$ and $\neg F$ is weakly-guarded in q). Since $\alpha(p) < \alpha(q)$, $\mathsf{CERTAINTY}(p)$ is in **FO** by the induction hypothesis. By Lemma 6.6, the problem $\mathsf{CERTAINTY}(q'_{[\vec{x}\mapsto\vec{c}]} \wedge \vec{y} \neq \vec{d})$ can be first-order reduced to $\mathsf{CERTAINTY}(p)$, and can thus be solved in **FO**.

Since the conditions in item 2 can be tested in first-order logic, $\mathsf{CERTAINTY}(q_{[\vec{x}\mapsto\vec{c}]})$ is in **FO**. The construction of a consistent first-order rewriting for $\mathsf{CERTAINTY}(q_{[\vec{x}\mapsto\vec{c}]})$ goes as follows. Assume $F_{[\vec{x}\mapsto\vec{c}]} = R(\underline{\vec{c}}, \vec{s})$. Notice that $\mathsf{vars}(\vec{s}) = \mathsf{vars}(\vec{y})$, but unlike $\vec{y}$, the sequence $\vec{s}$ can contain constants and double occurrences of the same variable. Note that every variable of $\mathsf{vars}(\vec{y})$ will occur in some non-negated atom of $q'_{[\vec{x}\mapsto\vec{c}]}$. If $\mathsf{vars}(\vec{y}) = \emptyset$, a consistent first-order rewriting for $q_{[\vec{x}\mapsto\vec{c}]}$ is $\psi \wedge \neg R(\underline{\vec{c}}, \vec{s})$, where ψ is a consistent first-order rewriting for $q'_{[\vec{x}\mapsto\vec{c}]}$. If $\mathsf{vars}(\vec{y}) \neq \emptyset$, a consistent first-order rewriting for $q_{[\vec{x}\mapsto\vec{c}]}$ is $\psi \wedge \forall\vec{z}(R(\underline{\vec{c}}, \vec{z}) \rightarrow \phi)$, where $\vec{z}$ is a sequence of fresh variables, of the same length as $\vec{s}$, and ϕ is a consistent first-order rewriting for $q'_{[\vec{x}\mapsto\vec{c}]} \wedge \vec{z} \neq \vec{s}$. □

Example 6.11. Consider $q = \{P(\underline{y}), \neg N(\underline{c}, a, y, y)\}$. The attack graph of q contains a single edge from N to P. The following query is a consistent first-order rewriting for q:

$$\exists y P(\underline{y}) \wedge \forall z_1 \forall z_2 \forall z_3$$
$$(N(\underline{c}, z_1, z_2, z_3) \rightarrow \exists y(P(\underline{y}) \wedge \neg(z_1 = a \wedge z_2 = y \wedge z_3 = y))),$$

which can be simplified into:
$$\exists y P(\underline{y}) \wedge \forall z(N(\underline{c}, a, z, z) \rightarrow \exists y(P(\underline{y}) \wedge y \neq z)). \quad \square$$

Example 6.12. Example 1.2 argued that the query $q_{Hall} = \{S(\underline{x}), \neg N_1(\underline{c}, x), \neg N_2(\underline{c}, x), \ldots, \neg N_\ell(\underline{c}, x)\}$ captures the complement of S-Covering. The attack graph of q_{Hall} is acyclic, and thus q_{Hall} has a consistent first-order rewriting. Figure 2 shows a consistent first-order rewriting for the case $\ell = 3$ that is constructed as outlined in the proof of Lemma 6.1.

Let **db** be an input database to CERTAINTY(q_{Hall}). What the rewriting says is that for every (possibly empty) subset $\mathbf{s} \subseteq \mathbf{db}$ containing only N_i-facts and not containing two facts with the same relation name (i.e., if $N_i(\underline{c}, u), N_j(\underline{c}, w) \in \mathbf{s}$, then $i = j$ implies $u = w$), there exists an S-fact $S(\underline{a})$ in **db** such that a does not occur in any of the N_i-facts of **s**.

Consider the instance of S-Covering defined in Example 1.2, and let **db** be the database obtained by the reduction described in that example. It can be easily verified that the S-Covering problem has no solution if and only if **db** satisfies the consistent first-order rewriting for q_{Hall}. Note incidentally that the length of the rewriting is exponential in the size of the rewritten query. $\square$

7 BEYOND WEAKLY-GUARDED NEGATION

Theorem 4.3 covers all queries of sjfBCQ$^\neg$ with weakly-guarded negation. However, it does not extend to all sjfBCQ$^\neg$ queries. In fact, one can show that sjfBCQ$^\neg$ contains queries q with an acyclic attack graph such that CERTAINTY(q) is not in **FO**; and queries q' with a cyclic attack graph such that CERTAINTY(q') is in **FO** (an example is the query q_4 of Example 7.1). Thus, for sjfBCQ$^\neg$ queries in which negation is not weakly-guarded, acyclicity of the attack graph is neither necessary nor sufficient for the existence of consistent first-order rewritings. Note nevertheless that a cycle of length two with at least one non-negated atom always implies that no consistent first-order rewriting exists, even if negation is not weakly-guarded, because Lemmas 5.5 and 5.6 do not make the hypothesis that negation is weakly-guarded.

We now argue that allowing non-weakly-guarded negation requires some fundamental, but currently not well understood, extension in the construction of consistent first-order rewritings.

Our construction of consistent-first-order rewritings in the proof of Lemma 6.1 repeatedly finds a literal F or $\neg F$ such that key(F) is reifiable, and then rewrites F by means of an existential quantification over the variables in key(F) (and a universal quantification over the remaining variables of F). Intuitively, the existential quantification fixes a block, and the universal quantification ranges over all facts of that block. Our results tell us that if negation is weakly-guarded, then a consistent first-order rewriting (if it exists) can always be constructed in this way. The following Example 7.1 shows that this construction is no longer sufficient if we allow non-weakly-guarded negation. Indeed, for the query q_4 in that example, no primary key is reifiable, and yet q_4 has a consistent first-order rewriting.

X: $\underline{x}$
1
2
3

Y: $\underline{y}$
a
b

R: $\underline{x}$	y
1	a
1	b
2	a
2	b
3	a
3	b

S: $\underline{y}$	x
a	1
a	2
a	3
b	1
b	2
b	3

Figure 3: All repairs satisfy the Boolean query $q_4 = \{X(\underline{x}), Y(\underline{y}), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\}$.

Example 7.1. Consider the query
$$q_4 = \{X(\underline{x}), Y(\underline{y}), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\},$$
in which negation is not weakly-guarded. We can show that the problem CERTAINTY(q_4) is in **FO** by means of combinatorial arguments. Let **db** be a set containing X-facts $X(\underline{a_1}), X(\underline{a_2}), \ldots, X(\underline{a_m})$, and Y-facts $Y(\underline{b_1}), Y(\underline{b_2}), \ldots, Y(\underline{b_n})$. If $m = 0$ or $n = 0$, then q_4 is obviously false in every repair of **db**. So assume next that $m, n \geq 1$, and consider the negation of q_4:
$$\neg q_4 \equiv \forall x \forall y((X(x) \wedge Y(y)) \rightarrow (R(\underline{x}, y) \vee S(\underline{y}, x))).$$
For every repair, there are $m \times n$ different valuations that make the premise $X(x) \wedge Y(y)$ in $\neg q_4$ true, but a repair can contain at most m R-facts of the form $R(\underline{a_i}, b_j)$ and at most n S-facts of the form $S(\underline{b_j}, a_i)$. It follows that if $m \times n > m + n$ (i.e., if $m \neq 1$, $n \neq 1$, and either $m \neq 2$ or $n \neq 2$, which can be tested in **FO**), then no repair can satisfy $\neg q_4$, and thus every repair will satisfy q_4, no matter what is the content of R and S. For example, in the database of Fig. 3, we have $m = 3$ and $n = 2$; since $3 \times 2 > 3 + 2$, every repair satisfies q_4. The only remaining cases to consider are $m = 1$, $n = 1$, or $m = n = 2$. These degenerated cases can be easily dealt with in **FO**. For example, if $m = n = 2$, then there exists a repair satisfying $\neg q_4$ if and only if **db** includes $\{R(\underline{a_1}, b_{j_1}), R(\underline{a_2}, b_{j_2}), S(\underline{b_{j_1}}, a_2), S(\underline{b_{j_2}}, a_1)\}$, with $1 \leq j_1 \neq j_2 \leq 2$; indeed, the latter set is consistent and covers the four pairs in $\{a_1, a_2\} \times \{b_1, b_2\}$. So we can conclude that CERTAINTY(q_4) is in **FO**.

Significantly, neither $\{x\}$ nor $\{y\}$ is reifiable in q_4. Indeed, as argued above, all repairs of the database of Fig. 3 satisfy q_4, but for every $i \in \{1, 2, 3\}$, there exists a repair falsifying $q_{[x \mapsto i]}$; likewise, there exists a repair falsifying $q_{[y \mapsto a]}$, and a repair falsifying $q_{[y \mapsto b]}$. To conclude, CERTAINTY(q_4) has a consistent first-order rewriting which is not based on "reification," and which differs in a fundamental way from the consistent first-order rewritings that apply for weakly-guarded negation. $\square$

Even in the case of non-weakly-guarded negation, reifiable variables can be eliminated in **FO** complexity. An important task is therefore to determine decidability of the following problem: given a query q in sjfBCQ$^\neg$ and a variable $x \in \text{vars}(q)$, is $\{x\}$ reifiable in q? The following lemma states that attacked variables are not reifiable (independent of whether negation is weakly-guarded or not). Together with Corollary 6.9, this implies that in the case of weakly-guarded negation, the reifiable variables are exactly the unattacked variables. Such a characterization remains open for non-weakly-guarded negation.

Proposition 7.2. *Let q be a query in* sjfBCQ$^{\neg}$*, and let $x \in$* vars(q)*. If $F \overset{q}{\rightsquigarrow} x$ for some $F \in q^{+} \cup q^{-}$, then $\{x\}$ is not reifiable in q.*

8 CONCLUSION

We have studied the complexity of the problem CERTAINTY(q) for queries q in sjfBCQ$^{\neg}$, the class of self-join-free Boolean conjunctive queries with negated atoms. It was shown that for queries q in sjfBCQ$^{\neg}$ with weakly-guarded negation, membership of the problem CERTAINTY(q) in **FO** is decidable. Moreover, if CERTAINTY(q) is in **FO**, then a consistent first-order rewriting for q can be effectively constructed. It remains an open problem to extend this decidability result to the entire class sjfBCQ$^{\neg}$. Weakly-guarded negation relaxes guarded negation. Remarkably, within **FO**, the same "rewriting-by-reification" method applies to all queries with weakly-guarded negation, but falls short when negation is not weakly-guarded.

Another open problem is to determine, for queries q in sjfBCQ$^{\neg}$, the exact complexity of CERTAINTY(q) if it is not in **FO**. In this respect, notice that for $q_1 = \{R(\underline{x}, y), \neg S(\underline{y}, x)\}$, CERTAINTY$(q_1)$ is intimately related to Bipartite Perfect Matching whose exact complexity is an important open problem. Further, we pose the conjecture that for every query q in sjfBCQ$^{\neg}$, CERTAINTY(q) is either in **P** or **coNP**-complete.

ACKNOWLEDGMENTS

Paraschos Koutris was supported by Google and by the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

A PROOFS OF SECTION 4

Proof of Lemma 4.7. Assume $F|w \overset{q}{\rightsquigarrow} u$ for some $w \in \mathsf{vars}(F)$ and $u \in \mathsf{vars}(q)$. We can assume a sequence $(w_0, w_1, \dots, w_\ell)$ that is a witness for $F|w \overset{q}{\rightsquigarrow} u$ (thus $w_0 = w$ and $w_\ell = u$).

Let $P \in q^{+} \setminus \{F\}$ such that $u \in \mathsf{vars}(P)$. Assume towards a contradiction that for every $x \in \mathsf{key}(P)$, we have $F|w \overset{q}{\not\rightsquigarrow} x$. Then, for every $x \in \mathsf{key}(P)$, $(w_0, w_1, \dots, w_\ell, x)$ is not a witness for $F|w \overset{q}{\rightsquigarrow} x$, and thus, since $w_\ell, x \in \mathsf{vars}(P)$, we have $x \in F^{\oplus,q}$. Consequently, $\mathsf{key}(P) \subseteq F^{\oplus,q}$, and thus $\mathsf{vars}(P) \subseteq F^{\oplus,q}$. Then $u \in F^{\oplus,q}$, and thus $F \overset{q}{\not\rightsquigarrow} u$, a contradiction. We conclude by contradiction that $F|w \overset{q}{\rightsquigarrow} x$ for some $x \in \mathsf{key}(F)$. □

Proof of Lemma 4.8. Assume $F \overset{q}{\rightsquigarrow} P$. We can assume variables $w \in \mathsf{vars}(F)$ and $x \in \mathsf{key}(P)$ such that $(w_0, w_1, \dots, w_\ell)$ is a witness for $F|w \overset{q}{\rightsquigarrow} x$ (thus $w_0 = w$ and $w_\ell = x$). Then, for every $u \in \mathsf{vars}(P) \setminus F^{\oplus,q}$, the sequence $(w_0, w_1, \dots, w_\ell, u)$ is a witness for $F|w \overset{q}{\rightsquigarrow} u$, hence $F \overset{q}{\rightsquigarrow} u$. □

The following lemma will be used in the proof of Lemma 4.9; it does not assume that negation is weakly-guarded.

Lemma A.1. *Let q be a query in* sjfBCQ$^{\neg}$*. Let $F, G \in q^{+} \cup q^{-}$ such that $F \neq G$. If*

- *$F|v_F \overset{q}{\rightsquigarrow} w_G$ for some $v_F \in \mathsf{vars}(F)$ and $w_G \in \mathsf{key}(G)$*
- *and $G|x_G \overset{q}{\rightsquigarrow} y$ for $x_G \in \mathsf{vars}(G)$ and $y \in \mathsf{vars}(q)$ such that*
- *there exists $P \in q^{+}$ such that $w_G, x_G \in \mathsf{vars}(P)$*

then either $F|v_F \overset{q}{\rightsquigarrow} y$ or $G|x_G \overset{q}{\rightsquigarrow} v$ for some $v \in \mathsf{key}(F)$.

Proof. Assume $F|v_F \overset{q}{\rightsquigarrow} w_G$ and $G|x_G \overset{q}{\rightsquigarrow} y$. Let $P \in q^{+}$ such that $w_G, x_G \in \mathsf{vars}(P)$. Assume $F|v_F \overset{q}{\not\rightsquigarrow} y$. Let $(v_0, v_1, \dots, v_\ell)$ be a witness for $F|v_F \overset{q}{\rightsquigarrow} w_G$ (thus $v_0 = v_F$ and $v_\ell = w_G$). Let $(x_0, x_1, \dots, x_m)$ be a witness for $G|x_G \overset{q}{\rightsquigarrow} y$ (thus $x_0 = x_G$ and $x_m = y$). Since $w_G, x_G \in \mathsf{vars}(P)$ and $(v_0, v_1, \dots, v_\ell, x_0, x_1, \dots, x_m)$ is not a witness for $F|v_F \overset{q}{\rightsquigarrow} y$, it follows that for some $j \in \{0, \dots, m\}$, we have $x_j \in F^{\oplus,q}$. Then, there exists a sequence

$$S_0, P_1, S_1, P_2, S_2, \dots, P_{n-1}, S_{n-1}, P_n, S_n$$

such that

- $\mathsf{key}(F) = S_0 \subsetneq S_1 \subsetneq \dots \subsetneq S_n \subseteq \mathsf{vars}(q)$;
- $P_1, P_2, \dots, P_n \in q^{+} \setminus \{F\}$;
- for all $i \in \{1, \dots, n\}$, $\mathsf{key}(P_i) \subseteq S_{i-1}$ and $S_i = S_{i-1} \cup \mathsf{vars}(P_i)$; and
- $x_j \in S_n$.

We show that $G \notin \{P_1, P_2, \dots, P_n\}$. Assume, towards a contradiction, that for $k \in \{1, \dots, n\}$, $G = P_k$. Then, $G \in q^{+}$ and $\mathsf{key}(G) \subseteq F^{\oplus,q}$, hence $\mathsf{vars}(G) \subseteq F^{\oplus,q}$. Then $w_G \in F^{\oplus,q}$, which contradicts $F|v_F \overset{q}{\rightsquigarrow} w_G$.

We show the following:

> ***Countdown Property:*** for all $i \in \{1, \dots, n\}$, if $G|x_G \overset{q}{\rightsquigarrow} u$ for some $u \in S_i$, then for some $w \in S_{i-1}$, we have $G|x_G \overset{q}{\rightsquigarrow} w$.

To this extent, assume $i \in \{1, \dots, n\}$ and $u \in S_i$ such that $G|x_G \overset{q}{\rightsquigarrow} u$. The desired result is obvious if $u \in S_{i-1}$. Assume next that $u \notin S_{i-1}$. Then, $u \in \mathsf{vars}(P_i) \setminus \mathsf{key}(P_i)$. By Lemma 4.7, $G|x_G \overset{q}{\rightsquigarrow} w$ for some $w \in \mathsf{key}(P_i)$. From $\mathsf{key}(P_i) \subseteq S_{i-1}$, it follows $w \in S_{i-1}$.

Since $G|x_G \overset{q}{\rightsquigarrow} x_j$ with $x_j \in S_n$, by repeated application of the *Countdown Property*, we obtain that $G|x_G \overset{q}{\rightsquigarrow} v$ for some $v \in S_0 = \mathsf{key}(F)$. □

We can now give the proof of Lemma 4.9.

Proof of Lemma 4.9. The proof is obvious if $F = H$. We next treat the case $F \neq H$.

Assume $F \overset{q}{\rightsquigarrow} G$ and $G \overset{q}{\rightsquigarrow} H$. We can assume $v_F \in \mathsf{vars}(F)$ and $w_G \in \mathsf{key}(G)$ such that $F|v_F \overset{q}{\rightsquigarrow} w_G$. Likewise, we can assume $x_G \in \mathsf{vars}(G)$ and $y_H \in \mathsf{key}(H)$ such that $G|x_G \overset{q}{\rightsquigarrow} y_H$. Since negation is weakly-guarded, we can assume $P \in q^{+}$ such that $w_G, x_G \in \mathsf{vars}(P)$ (if $G \in q^{+}$, then P can be taken to be G). By Lemma A.1, either $F|v_F \overset{q}{\rightsquigarrow} y_H$ or $G|x_G \overset{q}{\rightsquigarrow} v$ for some $v \in \mathsf{key}(F)$ (or both). Thus, either $F \overset{q}{\rightsquigarrow} H$ or $G \overset{q}{\rightsquigarrow} F$ (or both). □

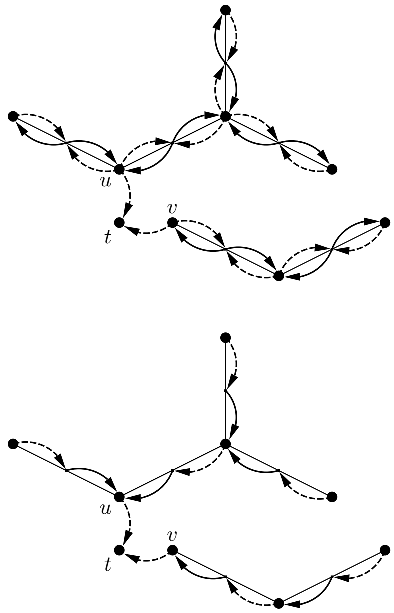

Figure 4: *Top:* Database that results from the reduction in the proof of Lemma 5.3. The round nodes and straight lines represent the undirected graph. Dashed and full arrows represent S-facts and T-facts respectively. *Bottom:* A repair that falsifies the query.

B PROOFS OF SECTION 5

Proof of Lemma 5.3. We show a first-order reduction from the problem UFA (Undirected Forest Accessibility) [8] to the problem CERTAINTY(q_2). 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 **L**-complete, and remains **L**-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 a database **db** as follows:

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

Clearly, the computation of **db** from G is in **FO**. As illustrated in Fig. 4, the database **db** can be represented as a digraph with dashed and full edges representing S-facts and T-facts respectively. Notice that every vertex in this digraph corresponds to either a vertex or an (undirected) edge of G.

We next show that there exists a path between u and v in G if and only if every repair of **db** satisfies q_2.

Assume first that u, v are not connected in G. A repair **r** of **db** that falsifies q_2 can be constructed as illustrated in Fig. 4. This repair **r** contains $S(\underline{u}, t)$ and $S(\underline{v}, t)$, and for every $s \notin \{u, v\}$, the repair contains a directed path from s to either u or v. In this way, for every $R(\underline{c, d}) \in \mathbf{r}$, either $S(\underline{c}, d) \in \mathbf{r}$ or $T(\underline{d}, c) \in \mathbf{r}$, hence $\mathbf{r} \not\models q_2$.

For the opposite implication, assume that u and v are connected in G. Let $u_0, u_1, \ldots, u_n$ with $u_0 = u$ and $u_n = v$ be the (unique) path between u and v in G. Assume towards a contradiction that **r** is a repair of **db** such that $\mathbf{r} \not\models q_2$. We show by induction on increasing i that for all $i \in \{0, \ldots, n-1\}$, the facts $S(\underline{u_{i+1}}, \{u_i, u_{i+1}\})$ and $T(\underline{\{u_i, u_{i+1}\}}, u_i)$ both belong to **r**.

For the induction basis, let $i = 0$. Since **r** contains $R(\underline{u_0, t})$ and $\mathbf{r} \not\models q_2$, it must be the case that $S(\underline{u_0}, t) \in \mathbf{r}$. Since $R(\underline{u_0, \{u_0, u_1\}}) \in \mathbf{r}$ and $S(\underline{u_0}, \{u_0, u_1\}) \notin \mathbf{r}$ and $\mathbf{r} \not\models q_2$, it must be the case that $T(\underline{\{u_0, u_1\}}, u_0) \in \mathbf{r}$. Since $R(\underline{u_1, \{u_0, u_1\}}) \in \mathbf{r}$ and $T(\underline{\{u_0, u_1\}}, u_1) \notin \mathbf{r}$ and $\mathbf{r} \not\models q_2$, it must be the case that $S(\underline{u_1}, \{u_0, u_1\}) \in \mathbf{r}$.

For the induction step, $i \longrightarrow i+1$, the induction hypothesis is that $S(\underline{u_{i+1}}, \{u_i, u_{i+1}\})$ and $T(\underline{\{u_i, u_{i+1}\}}, u_i)$ both belong to **r**. Since $R(\underline{u_{i+1}, \{u_{i+1}, u_{i+2}\}}) \in \mathbf{r}$ and $S(\underline{u_{i+1}}, \{u_{i+1}, u_{i+2}\}) \notin \mathbf{r}$ and $\mathbf{r} \not\models q_2$, it must be the case that $T(\underline{\{u_{i+1}, u_{i+2}\}}, u_{i+1}) \in \mathbf{r}$. Since $R(\underline{u_{i+2}, \{u_{i+1}, u_{i+2}\}}) \in \mathbf{r}$ and $T(\underline{\{u_{i+1}, u_{i+2}\}}, u_{i+2}) \notin \mathbf{r}$ and $\mathbf{r} \not\models q_2$, it must be the case that $S(\underline{u_{i+2}}, \{u_{i+1}, u_{i+2}\}) \in \mathbf{r}$.

It follows that $S(\underline{v}, \{u_{n-1}, v\}) \in \mathbf{r}$ (recall $v = u_n$). But then $S(\underline{v}, t) \notin \mathbf{r}$. Since **r** contains $R(\underline{v, t})$ but neither $S(\underline{v}, t)$ nor $T(\underline{t}, v)$ (the latter fact does not belong to **db**), it follows that $\mathbf{r} \models q_2$, a contradiction. □

Proof of Lemma 5.4. Let $q^+ \subseteq q' \subseteq q$. We show a first-order reduction from CERTAINTY(q') to CERTAINTY(q). Let **db** be a database that is input to CERTAINTY(q'). Let $\mathbf{db}_0$ be the database obtained from **db** by deleting, for all $\neg N \in q \setminus q'$, all N-facts. Obviously, $\mathbf{db}_0$ can be computed in **FO**. It is straightforward to see that every repair of **db** satisfies q' if and only if every repair of $\mathbf{db}_0$ satisfies q. □

Proof of Lemma 5.7. We will show a first-order reduction from CERTAINTY(q_2) with $q_2 = \{T(\underline{x}, \underline{y}), \neg R(\underline{x}, y), \neg S(\underline{y}, x)\}$.

Since $F \overset{q}{\rightsquigarrow} G$, there exists $v_F \in \mathsf{vars}(F), u \in \mathsf{key}(G)$ such that $F|v_F \overset{q}{\rightsquigarrow} u$. Similarly, there exists $v_G \in \mathsf{vars}(G), u' \in \mathsf{key}(F)$ such that $G|v_G \overset{q}{\rightsquigarrow} u'$. For all $a \in \mathsf{type}(x)$ and $b \in \mathsf{type}(y)$, define Θ_b^a as the following valuation over $\mathsf{vars}(q)$. For every $w \in \mathsf{vars}(q)$,

$$\Theta_b^a(w) = \begin{cases} a & \text{if } G|v_G \overset{q}{\rightsquigarrow} w \text{ and } F|v_F \overset{q}{\not\rightsquigarrow} w \\ b & \text{if } F|v_F \overset{q}{\rightsquigarrow} w \text{ and } G|v_G \overset{q}{\not\rightsquigarrow} w \\ \langle a, b \rangle & \text{if } F|v_F \overset{q}{\rightsquigarrow} w \text{ and } G|v_G \overset{q}{\rightsquigarrow} w \\ \perp & \text{otherwise} \end{cases}$$

Sublemma B.1. *For every $H \in q^+$, for all $a, a' \in \mathsf{type}(x)$ and $b, b' \in \mathsf{type}(y)$, $\{\Theta_b^a(H), \Theta_{b'}^{a'}(H)\}$ is consistent.*

Proof. Assume $\Theta_b^a(H)$ and $\Theta_{b'}^{a'}(H)$ are key-equal. We need to show $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a = a'$ and $b = b'$. Trivial.

Case $a = a'$ and $b \neq b'$. Then, $F|v_F \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathsf{key}(H)$. By Lemma 4.7, $F|v_F \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathsf{vars}(H)$ and, consequently, $\Theta_b^a(H) = \Theta_{b'}^{a'}(H)$.

Case $a \neq a'$ and $b = b'$. Symmetric to the previous case.

Case $a \neq a'$ **and** $b \neq b'$**.** Then $F|v_F \overset{q}{\not\rightsquigarrow} w$ and $G|v_G \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathsf{key}(H)$. By Lemma 4.7, $F|v_F \overset{q}{\not\rightsquigarrow} w$ and $G|v_G \overset{q}{\not\rightsquigarrow} w$ for all $w \in \mathsf{vars}(H)$ and, consequently, $\Theta^a_b(H) = \Theta^{a'}_{b'}(H)$.

This concludes the proof of Sublemma B.1. $\square$

Sublemma B.2. *For all $a, a' \in \mathsf{type}(x)$ and $b, b' \in \mathsf{type}(y)$,*

(1) *$\Theta^a_b(F)$ and $\Theta^{a'}_{b'}(F)$ are key-equal if and only if $a = a'$; and*
(2) *$\Theta^a_b(F)$ and $\Theta^{a'}_{b'}(F)$ are equal if and only if $a = a'$ and $b = b'$.*

Consequently, the set $\{R(\underline{a}, b), R(\underline{a'}, b')\}$ is consistent if and only if $\{\Theta^a_b(F), \Theta^{a'}_{b'}(F)\}$ is consistent.

Proof. $\boxed{1 \Longrightarrow}$ $G|v_G \overset{q}{\rightsquigarrow} u'$ for some $u' \in \mathsf{key}(F)$. $\boxed{1 \Longleftarrow}$ If $w \in \mathsf{key}(F)$, then $F|v_F \overset{q}{\not\rightsquigarrow} w$. $\boxed{2 \Longrightarrow}$ $F|v_F \overset{q}{\rightsquigarrow} v_F$ with $v_F \in \mathsf{vars}(F)$. $\boxed{2 \Longleftarrow}$ Trivial. This concludes the proof of Sublemma B.2. $\square$

Sublemma B.3. *For all $a, a' \in \mathsf{type}(x)$ and $b, b' \in \mathsf{type}(y)$,*

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

Consequently, the set $\{S(\underline{b}, a), S(\underline{b'}, a')\}$ is consistent if and only if $\{\Theta^a_b(G), \Theta^{a'}_{b'}(G)\}$ is consistent.

Proof. Symmetric to the proof of Sublemma B.2. $\square$

For every database $\mathbf{db}$ that is input to $\mathsf{CERTAINTY}(q_2)$, we define $f(\mathbf{db})$ as follows:

- if $\mathbf{db}$ contains $T(\underline{a, b})$, then $f(\mathbf{db})$ includes $\Theta^a_b(q^+)$;
- if $\mathbf{db}$ contains $R(\underline{a}, b)$, then $f(\mathbf{db})$ contains $\Theta^a_b(F)$; and
- if $\mathbf{db}$ contains $S(\underline{b}, a)$, then $f(\mathbf{db})$ contains $\Theta^a_b(G)$.

Clearly, f is computable in **FO**. Note that whenever H is a relation name in $q^- \setminus \{F, G\}$, then $f(\mathbf{db})$ contains no H-facts. Let

$$\begin{aligned} g(\mathbf{db}) = (f(\mathbf{db}) &\setminus \{\Theta^a_b(F) \mid R(\underline{a}, b) \in \mathbf{db}\}) \\ &\setminus \{\Theta^a_b(G) \mid S(\underline{b}, a) \in \mathbf{db}\}. \end{aligned}$$

By Sublemmas B.1, B.2, and B.3,

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

Let $\mathbf{db}$ be a database that is input to $\mathsf{CERTAINTY}(q_2)$. Let $\mathbf{r}$ be a repair of $\mathbf{db}$. It suffices to show that the following are equivalent:

(1) $\mathbf{r} \models q_2$;
(2) $f(\mathbf{r}) \cup g(\mathbf{db}) \models q$.

$\boxed{1 \Longrightarrow 2}$ We can assume $T(\underline{a, b}) \in \mathbf{r}$ such that $R(\underline{a}, b) \notin \mathbf{r}$ and $S(\underline{b}, a) \notin \mathbf{r}$. We have $\Theta^a_b(q^+) \subseteq f(\mathbf{r})$. Assume, towards a contradiction, $\Theta^a_b(F) \in f(\mathbf{r})$. Then there must exist $R(\underline{a'}, b') \in \mathbf{r}$ such that $\Theta^{a'}_{b'}(F) = \Theta^a_b(F)$. From the second item in Sublemma B.2, $a = a'$ and $b = b'$, hence $R(\underline{a}, b) \in \mathbf{r}$, a contradiction. Symmetrically, we can show that $\Theta^a_b(G) \notin f(\mathbf{r})$. Consequently, $f(\mathbf{r}) \cup g(\mathbf{db}) \models q$.

$\boxed{2 \Longrightarrow 1}$ Note that from $u' \in \mathsf{key}(F) \subseteq F^{\oplus,q}$, it follows $F|v_F \overset{q}{\not\rightsquigarrow} u'$. Likewise, from $u \in \mathsf{key}(G) \subseteq G^{\oplus,q}$, it follows $G|v_G \overset{q}{\not\rightsquigarrow} u$. Consequently, for all $a \in \mathsf{type}(x)$ and $b \in \mathsf{type}(y)$, $\Theta^a_b(u') = a$ and $\Theta^a_b(u) = b$.

Let θ be a valuation over $\mathsf{vars}(q)$ such that $\theta(q^+) \subseteq f(\mathbf{r}) \cup g(\mathbf{db})$ and $\theta(F), \theta(G) \notin f(\mathbf{r})$. Since q is weakly-guarded, there exists an atom $F' \in q^+$ such that $v_F, u' \in \mathsf{vars}(F')$ and an atom $G' \in q^+$ such that $v_G, u \in \mathsf{vars}(G')$. By our construction, we can assume atoms $T(\underline{a, b}), T(\underline{a', b'}) \in \mathbf{r}$ such that $\theta(F') = \Theta^a_b(F')$ and $\theta(G') = \Theta^{a'}_{b'}(G')$. Consequently, $\theta(v_F) = \Theta^a_b(v_F) \in \{b, \langle a, b\rangle\}$ and $\theta(v_G) = \Theta^{a'}_{b'}(v_G) \in \{a', \langle a', b'\rangle\}$. Furthermore, $\theta(u') = \Theta^a_b(u') = a$ and $\theta(u) = \Theta^{a'}_{b'}(u) = b'$.

It will be the case that $a = a'$ and $b = b'$. The proof of $b = b'$ is given in the next paragraph; the proof of $a = a'$ is symmetrical.

We can assume a sequence

$$(v_0, P_1, v_1, P_2, v_2 \ldots, P_{\ell-1}, v_{\ell-1}, P_\ell, v_\ell)$$

such that $(v_0, v_1, v_2 \ldots, v_{\ell-1}, v_\ell)$ is a witness for $F|v_F \overset{q}{\rightsquigarrow} u$ (thus $v_0 = v_F$ and $v_\ell = u$) and for all $i \in \{1, \ldots, \ell\}$, $P_i \in q^+$ such that $v_{i-1}, v_i \in \mathsf{vars}(P_i)$. Obviously, for every $i \in \{0, \ldots, \ell\}$, we have $F|v_F \overset{q}{\rightsquigarrow} v_i$. We show by induction on increasing i that for all $i \in \{0, \ldots, \ell\}$, there exists $a_i \in \mathsf{type}(x)$ such that $\theta(v_i) \in \{b, \langle a_i, b\rangle\}$. For the basis of the induction, $i = 0$, we have already shown $\theta(v_0) \in \{b, \langle a, b\rangle\}$. For the induction step, $i \longrightarrow i+1$, the induction hypothesis is $\theta(v_i) \in \{b, \langle a_i, b\rangle\}$. We can assume $T(\underline{a'', b''}) \in \mathbf{r}$ such that $\theta(P_{i+1}) = \Theta^{a''}_{b''}(P_{i+1}) \in \theta(q^+)$. Consequently, $\theta(v_i) = \Theta^{a''}_{b''}(v_i) \in \{b'', \langle a'', b''\rangle\}$ and $\theta(v_{i+1}) = \Theta^{a''}_{b''}(v_{i+1}) \in \{b'', \langle a'', b''\rangle\}$. Since $\theta(v_i) \in \{b, \langle a_i, b\rangle\}$ by the induction hypothesis, $b = b''$, which concludes the induction step (take a_{i+1} equal to a''). Consequently, for $i = \ell$, we obtain $\theta(u) \in \{b, \langle a_\ell, b\rangle\}$ for some $a_\ell \in \mathsf{type}(x)$. Since we also have $\theta(u) = b'$, it is correct to conclude $b' = b$.

We show that for every variable $x \in \mathsf{vars}(F) \cup \mathsf{vars}(G)$, $\theta(x) = \Theta^a_b(x)$. To this extent, let $x \in \mathsf{vars}(F)$ (the case $x \in \mathsf{vars}(G)$ is symmetrical). We can assume the existence of an atom P_1 such that $x, v_F \in \mathsf{vars}(P_1)$. We can assume $T(\underline{a_1, b_1}) \in \mathbf{r}$ such that $\theta(P_1) = \Theta^{a_1}_{b_1}(P_1) \in \theta(q^+)$. Hence, $\theta(v_F) = \Theta^{a_1}_{b_1}(v_F) \in \{b_1, \langle a_1, b_1\rangle\}$. Since also $\theta(v_F) \in \{b, \langle a, b\rangle\}$, it follows $b = b_1$. Thus, $\theta(x) = \Theta^{a_1}_b(x)$. Likewise, we can assume the existence of an atom P_2 such that $x, u' \in \mathsf{vars}(P_2)$. We can assume $a_2 \in \mathsf{type}(x)$ and $b_2 \in \mathsf{type}(y)$ such that $\theta(P_2) = \Theta^{a_2}_{b_2}(P_2) \in \theta(q^+)$. Hence, $\theta(u') = \Theta^{a_2}_{b_2}(u') = a_2$. From $\theta(u') = a$, it follows $a = a_2$. Thus, $\theta(x) = \Theta^a_{b_2}(x)$. From $\theta(x) = \Theta^{a_1}_b(x)$ and $\theta(x) = \Theta^a_{b_2}(x)$, it is correct to conclude $\theta(x) = \Theta^a_b(x)$.

Assume now towards a contradiction that $R(\underline{a}, b) \in \mathbf{r}$. Then, $\Theta^a_b(F) = \theta(F) \in f(\mathbf{r})$, a contradiction. Hence, $R(\underline{a}, b) \notin \mathbf{r}$. By a symmetric argument, $S(\underline{b}, a) \notin \mathbf{r}$. It follows $\mathbf{r} \models q_2$. This concludes the proof $\square$

C PROOFS OF SECTION 6

Full proof of Lemma 6.8. The proof is trivial if $A = B$. We next consider the case $A \neq B$.

Assume a valuation ζ over X such that $\mathbf{r}_B \models \zeta(q)$. We can assume a valuation ζ^+ over $\mathsf{vars}(q)$ that extends ζ such that $\mathbf{r}_B \models \zeta^+(q)$. Since A is key-relevant for q in $\mathbf{r}$, we can assume a valuation μ over $\mathsf{vars}(q)$ such that $\mathbf{r} \models \mu(q)$ and $\mu(G) \sim A$. We distinguish two cases: $G \in q^-$ and $G \in q^+$.

Case that $G \in q^-$. This part of the proof was given in the main body of the article.

Case that $G \in q^+$. The desired result is obvious if $\zeta^+(G) \neq B$. In the remainder of the proof, we assume $\zeta^+(G) = B$. From $\mu(G) \in \mathbf{r}$ and $\mu(G) \sim A$, it follows $\mu(G) = A$.

Since A and B are key-equal, ζ^+ and μ agree on all variables of key(G). Since $\mathbf{r} \setminus \{A\} = \mathbf{r}_B \setminus \{B\}$ and $\mu(q^+ \setminus \{G\}) \subseteq \mathbf{r} \setminus \{A\}$ and $\zeta^+(q^+ \setminus \{G\}) \subseteq \mathbf{r}_B \setminus \{B\}$, it follows from [29, Lemma 4.3] that ζ^+ and μ agree on all variables of $G^{\oplus,q}$. Then, vars$(G) \nsubseteq G^{\oplus,q}$, or else $A = B$, a contradiction.

We define κ as the valuation over vars(q) such that for every $P \in q^+$,

$$\kappa(P) = \begin{cases} \mu(P) & \text{if } P = G \text{ or } G \overset{q}{\rightsquigarrow} P \\ \zeta^+(P) & \text{otherwise} \end{cases}$$

We show below that κ is well-defined, that $\mathbf{r} \models \kappa(q)$, and that $\kappa[X] = \zeta$. It is then correct to conclude $\mathbf{r} \models \zeta(q)$.

Proof that κ is well-defined. To show that κ is well-defined, assume $P_1, P_2 \in q^+$ and $v \in \text{vars}(P_1) \cap \text{vars}(P_2)$ such that $P_1 \neq G$ and $G \overset{q}{\not\rightsquigarrow} P_1$ (the "otherwise" case in the definition of κ) and either $P_2 = G$ or $G \overset{q}{\rightsquigarrow} P_2$. From $G \overset{q}{\not\rightsquigarrow} P_1$, it follows $G \overset{q}{\not\rightsquigarrow} v$ by Lemma 4.7. If $P_2 = G$, then from $P_2 \overset{q}{\not\rightsquigarrow} v$ and $v \in \text{vars}(P_2)$, it follows $v \in G^{\oplus,q}$, hence $\zeta^+(v) = \mu(v)$. If $G \overset{q}{\rightsquigarrow} P_2$ with $P_2 \neq G$, then, since $G \overset{q}{\not\rightsquigarrow} v$, it follows $v \in G^{\oplus,q}$ by Lemma 4.8, hence $\zeta^+(v) = \mu(v)$.

Proof that $\mathbf{r} \models \kappa(q)$. We first show that $\kappa(q^+) \subseteq \mathbf{r}$. From $\kappa(G) = \mu(G)$ and $\mu(G) = A \in \mathbf{r}$, it follows $\kappa(G) \in \mathbf{r}$. For every $P \in q^+ \setminus \{G\}$, either $\kappa(P) = \mu(P) \in \mathbf{r}$ or $\kappa(P) = \zeta^+(P) \in \mathbf{r}_B \setminus \{B\} = \mathbf{r} \setminus \{A\}$. It is correct to conclude $\kappa(q^+) \subseteq \mathbf{r}$.

We show next that for every $N \in q^-$, $\kappa(N) \notin \mathbf{r}$. To this extent, let $N \in q^-$. We show below that either $\kappa(N) = \mu(N)$ or $\kappa(N) = \zeta^+(N)$ (or both). We have that $\mathbf{r}$ and $\mathbf{r}_B$ contain the same set of N-facts. Since $\mu(N) \notin \mathbf{r}$ and $\zeta^+(N) \notin \mathbf{r}_B$, it follows $\kappa(N) \notin \mathbf{r}$.

To show that $\kappa(N) = \mu(N)$ or $\kappa(N) = \zeta^+(N)$, let w be an arbitrary variable of vars(N). We distinguish two cases:

- Case $G \overset{q}{\rightsquigarrow} u$ for some $u \in \text{vars}(N)$. Since negation is weakly-guarded, we can assume $P \in q^+$ such that $u, w \in \text{vars}(P)$. We distinguish two cases: $P \neq G$ or $P = G$. If $P \neq G$, then, by Lemma 4.7, $G \overset{q}{\rightsquigarrow} P$, hence $\kappa(P) = \mu(P)$, thus $\kappa(w) = \mu(w)$. If $P = G$, then $\kappa(P) = \mu(P)$, hence $\kappa(w) = \mu(w)$.
- Case $G \overset{q}{\not\rightsquigarrow} u$ for all $u \in \text{vars}(N)$. Since negation is safe, we can assume $P \in q^+$ such that $w \in \text{vars}(P)$. If $G \neq P$ and $G \overset{q}{\not\rightsquigarrow} P$, then $\kappa(P) = \zeta^+(P)$, and thus $\kappa(w) = \zeta^+(w)$.

  Assume next that $G = P$. From $w \in \text{vars}(G)$ and $G \overset{q}{\not\rightsquigarrow} w$, it follows $w \in G^{\oplus,q}$. Since μ and ζ^+ agree on all variables of $G^{\oplus,q}$, it follows $\kappa(w) = \mu(w) = \zeta^+(w)$.

  Finally, assume that $G \neq P$ and $G \overset{q}{\rightsquigarrow} P$. From $G \overset{q}{\not\rightsquigarrow} w$, it follows $w \in G^{\oplus,q}$ by Lemma 4.8, hence $\kappa(w) = \mu(w) = \zeta^+(w)$.

Proof that $\kappa(x) = \zeta(x)$ for every $x \in X$. Let $x \in X$, and thus $G \overset{q}{\not\rightsquigarrow} x$. We can assume an atom $P \in q^+$ such that $x \in \text{vars}(P)$. If $P = G$, then it must be the case that $x \in G^{\oplus,q}$, hence $\kappa(x) = \zeta(x) = \mu(x)$. Assume next $G \neq P$. If $G \overset{q}{\not\rightsquigarrow} P$, then $\kappa(P) = \zeta^+(P)$, hence $\kappa(x) = \zeta(x)$. If $G \overset{q}{\rightsquigarrow} P$, then $x \in G^{\oplus,q}$ by Lemma 4.8, hence $\kappa(x) = \zeta(x) = \mu(x)$. This concludes the proof. □

Proof of Lemma 6.10. The proof of the second item is straightforward. For the first item, assume $F_{[x\mapsto c]} \overset{q_{[x\mapsto c]}}{\rightsquigarrow} G_{[x\mapsto c]}$. Then, we can assume $v_G \in \text{vars}(F) \setminus \{x\}$ and $w_G \in \text{key}(G) \setminus \{x\}$ and a sequence $(v_0, v_1, \ldots, v_\ell)$ that is a witness for $F_{[x\mapsto c]}|v_F \overset{q_{[x\mapsto c]}}{\rightsquigarrow} w_G$ (thus $v_0 = v_F$ and $v_\ell = w_G$). Thus, for every $i \in \{0, 1, \ldots, \ell\}$, $v_i \neq x$ and $v_i \notin {F_{[x\mapsto c]}}^{\oplus,q_{[x\mapsto c]}}$.

It is easy to show that $F^{\oplus,q} \setminus \{x\} \subseteq {F_{[x\mapsto c]}}^{\oplus,q_{[x\mapsto c]}}$. It follows that the above sequence is also a witness for $F|v_F \overset{q}{\rightsquigarrow} w_G$, hence $F \overset{q}{\rightsquigarrow} G$. □

D PROOF OF PROPOSITION 7.2

Proof of Proposition 7.2. Since $F \overset{q}{\rightsquigarrow} x$, there exists $v_F \in \text{vars}(F)$ such that $F|v_F \overset{q}{\rightsquigarrow} x$. We also have $F|v_F \overset{q}{\rightsquigarrow} v_F$. For every constant c, define Θ_c as the valuation over vars(q) such that for every $w \in \text{vars}(q)$,

$$\Theta_c(w) = \begin{cases} c & \text{if } F|v_F \overset{q}{\rightsquigarrow} w \\ \bot & \text{otherwise} \end{cases}$$

Let a, b be distinct constants. Let **db** be the database that includes both $\Theta_a(q^+)$ and $\Theta_b(q^+)$, and that contains both $\Theta_a(F)$ and $\Theta_b(F)$. Note that if T is a relation name in $q^- \setminus \{F\}$, then **db** contains no T-facts. We show that

(1) for every $H \in q^+ \setminus \{F\}$, $\{\Theta_a(H), \Theta_b(H)\}$ is consistent; and
(2) $\Theta_a(F)$ and $\Theta_b(F)$ are key-equal but distinct.

For the first item, by Lemma 4.7, either $F|v_F \overset{q}{\rightsquigarrow} u$ for some $u \in \text{key}(H)$, or $F|v_F \overset{q}{\not\rightsquigarrow} v$ for all $v \in \text{vars}(H)$. In the former case, $\Theta_a(H)$ and $\Theta_b(H)$ are not key-equal; in the latter case, $\Theta_a(H)$ and $\Theta_b(H)$ are identical. For the second item, from key$(F) \subseteq F^{\oplus,q}$, it follows that $\Theta_a(F)$ and $\Theta_b(F)$ are key-equal; from $\Theta_a(v_F) = a$ and $\Theta_b(v_F) = b$, it follows that $\Theta_a(F)$ and $\Theta_b(F)$ are distinct. It is correct to conclude that **db** has two repairs, which we denote as $\mathbf{r}_a \coloneqq \mathbf{db} \setminus \{\Theta_b(F)\}$ and $\mathbf{r}_b \coloneqq \mathbf{db} \setminus \{\Theta_a(F)\}$.

We show that whenever μ is a valuation over vars(q) such that $\mu(q^+) \subseteq \mathbf{db}$, then either $\mu = \Theta_a$ or $\mu = \Theta_b$. To this extent, assume $\mu(q^+) \subseteq \mathbf{db}$. Obviously, either $\mu(v_F) = a$ or $\mu(v_F) = b$. Assume $\mu(v_F) = a$ (the case $\mu(v_F) = b$ is symmetrical); it suffices to show $\mu = \Theta_a$. Let $v \in \text{vars}(q)$. If $F|v_F \overset{q}{\not\rightsquigarrow} v$, then $\mu(v) = \bot = \Theta_a(v)$. Assume next that $F|v_F \overset{q}{\rightsquigarrow} v$. Then, we can assume a sequence

$$(v_0, P_1, v_1, P_2, v_2 \ldots, P_{\ell-1}, v_{\ell-1}, P_\ell, v_\ell)$$

such that $(v_0, v_1, v_2 \ldots, v_{\ell-1}, v_\ell)$ is a witness for $F|v_F \overset{q}{\rightsquigarrow} v$ (thus $v_0 = v_F$ and $v_\ell = v$) and for all $i \in \{1, \ldots, \ell\}$, $P_i \in q^+$ such that $v_{i-1}, v_i \in \text{vars}(P_i)$. From $\mu(v_0) = a$ and $v_0 \in \text{vars}(P_1)$, it follows $\mu(P_1) = \Theta_a(P_1)$. It can now be easily shown, by induction on increasing i, that for all $i \in \{1, \ldots, \ell\}$, $\mu(P_i) = \Theta_a(P_i)$, and thus $\mu(v) = \Theta_a(v)$.

We show in the remainder that both repairs, $\mathbf{r}_a$ and $\mathbf{r}_b$, satisfy q, but that there exists no constant $c \in \{a, b\}$ such that both repairs satisfy $q_{[x\mapsto c]}$. We distinguish two cases.

Case $F \in q^-$. By symmetry, it suffices to show $\mathbf{r}_a \models q_{[x \mapsto b]}$ and $\mathbf{r}_a \not\models q_{[x \mapsto a]}$. The result $\mathbf{r}_a \models q_{[x \mapsto b]}$ follows from $\Theta_b(q^+) \subseteq \mathbf{r}_a$ and $\Theta_b(F) \notin \mathbf{r}_a$.

We next show $\mathbf{r}_a \not\models q_{[x \mapsto a]}$. Assume, towards a contradiction, a valuation μ over $\text{vars}(q) \setminus \{x\}$ such that $\mathbf{r}_a \models \mu(q_{[x \mapsto a]})$, thus $\mu(F_{[x \mapsto a]}) \notin \mathbf{r}_a$. Define $\mu_{[x \mapsto a]} := \mu \cup \{x \mapsto a\}$, a valuation over $\text{vars}(q)$ which is obviously distinct from Θ_b. Since $\mu_{[x \mapsto a]}(q^+) \subseteq \mathbf{r}_a$, it follows, by our previous reasoning, that either $\mu_{[x \mapsto a]} = \Theta_a$ or $\mu_{[x \mapsto a]} = \Theta_b$. So it must be the case that $\mu_{[x \mapsto a]} = \Theta_a$. We obtain $\mu(F_{[x \mapsto a]}) = \mu_{[x \mapsto a]}(F) = \Theta_a(F) \in \mathbf{r}_a$, a contradiction.

Case $F \in q^+$. By symmetry, it suffices to show $\mathbf{r}_a \models q_{[x \mapsto a]}$ and $\mathbf{r}_a \not\models q_{[x \mapsto b]}$. It is straightforward that $\mathbf{r}_a \models q_{[x \mapsto a]}$.

We next show $\mathbf{r}_a \not\models q_{[x \mapsto b]}$. Assume, towards a contradiction, a valuation μ over $\text{vars}(q) \setminus \{x\}$ such that $\mathbf{r}_a \models \mu(q_{[x \mapsto b]})$, thus $\mu(F_{[x \mapsto b]}) \in \mathbf{r}_a$. Define $\mu_{[x \mapsto b]} := \mu \cup \{x \mapsto b\}$, a valuation over $\text{vars}(q)$ which is obviously distinct from Θ_a. Since $\mu_{[x \mapsto b]}(q^+) \subseteq \mathbf{r}_a$, it follows, by our previous reasoning, that either $\mu_{[x \mapsto b]} = \Theta_a$ or $\mu_{[x \mapsto b]} = \Theta_b$. So it must be the case that $\mu_{[x \mapsto b]} = \Theta_b$. We obtain $\mu(F_{[x \mapsto b]}) = \mu_{[x \mapsto b]}(F) = \Theta_b(F) \in \mathbf{r}_a$, a contradiction. □

REFERENCES

[1] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. 1999. Consistent Query Answers in Inconsistent Databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania, USA*, Victor Vianu and Christos H. Papadimitriou (Eds.). ACM Press, 68–79. https://doi.org/10.1145/303976.303983

[2] Marcelo Arenas, Leopoldo E. Bertossi, Jan Chomicki, Xin He, Vijay Raghavan, and Jeremy P. Spinrad. 2003. Scalar aggregation in inconsistent databases. *Theor. Comput. Sci.* 296, 3 (2003), 405–434. https://doi.org/10.1016/S0304-3975(02)00737-5

[3] Vince Bárány, Balder ten Cate, and Luc Segoufin. 2015. Guarded Negation. *J. ACM* 62, 3 (2015), 22:1–22:26. https://doi.org/10.1145/2701414

[4] Leopoldo E. Bertossi. 2011. *Database Repairing and Consistent Query Answering*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00379ED1V01Y201108DTM020

[5] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. 2016. Explaining Inconsistency-Tolerant Query Answering over Description Logic Knowledge Bases. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 900–906. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12025

[6] Andrei A. Bulatov. 2011. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.* 12, 4 (2011), 24:1–24:66. https://doi.org/10.1145/1970398.1970400

[7] Ashok K. Chandra, Larry J. Stockmeyer, and Uzi Vishkin. 1984. Constant Depth Reducibility. *SIAM J. Comput.* 13, 2 (1984), 423–439. https://doi.org/10.1137/0213028

[8] Stephen A. Cook and Pierre McKenzie. 1987. Problems Complete for Deterministic Logarithmic Space. *J. Algorithms* 8, 3 (1987), 385–394. https://doi.org/10.1016/0196-6774(87)90018-6

[9] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. 2016. Bipartite perfect matching is in quasi-NC. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 754–763. https://doi.org/10.1145/2897518.2897564

[10] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. 2017. Guest Column: Parallel Algorithms for Perfect Matching. *SIGACT News* 48, 1 (2017), 102–109. https://doi.org/10.1145/3061640.3061655

[11] Gaëlle Fontaine. 2013. Why is it Hard to Obtain a Dichotomy for Consistent Query Answering?. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 550–559. https://doi.org/10.1109/LICS.2013.62

[12] Ariel Fuxman and Renée J. Miller. 2007. First-order query rewriting for inconsistent databases. *J. Comput. Syst. Sci.* 73, 4 (2007), 610–635. https://doi.org/10.1016/j.jcss.2006.10.013

[13] Víctor Gutiérrez-Basulto, Yazmin Angélica Ibáñez-García, and Roman Kontchakov. 2012. An Update on Query Answering with Restricted Forms of Negation. In *Web Reasoning and Rule Systems - 6th International Conference, RR 2012, Vienna, Austria, September 10-12, 2012. Proceedings (Lecture Notes in Computer Science)*, Markus Krötzsch and Umberto Straccia (Eds.), Vol. 7497. Springer, 75–89. https://doi.org/10.1007/978-3-642-33203-6_7

[14] Philip Hall. 1935. On Representatives of Subsets. *J. London Math. Soc.* 10, 1 (1935), 26–30.

[15] Phokion G. Kolaitis and Enela Pema. 2012. A dichotomy in the complexity of consistent query answering for queries with two atoms. *Inf. Process. Lett.* 112, 3 (2012), 77–85. https://doi.org/10.1016/j.ipl.2011.10.018

[16] Phokion G. Kolaitis, Enela Pema, and Wang-Chiew Tan. 2013. Efficient Querying of Inconsistent Databases with Binary Integer Programming. *PVLDB* 6, 6 (2013), 397–408. http://www.vldb.org/pvldb/vol6/p397-tan.pdf

[17] Paraschos Koutris and Dan Suciu. 2014. A Dichotomy on the Complexity of Consistent Query Answering for Atoms with Simple Keys, See [27], 165–176. https://doi.org/10.5441/002/icdt.2014.19

[18] Paraschos Koutris and Jef Wijsen. 2015. The Data Complexity of Consistent Query Answering for Self-Join-Free Conjunctive Queries Under Primary Key Constraints. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, Tova Milo and Diego Calvanese (Eds.). ACM, 17–29. https://doi.org/10.1145/2745754.2745769

[19] Paraschos Koutris and Jef Wijsen. 2017. Consistent Query Answering for Self-Join-Free Conjunctive Queries Under Primary Key Constraints. *ACM Trans. Database Syst.* 42, 2 (2017), 9:1–9:45. https://doi.org/10.1145/3068334

[20] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. 2015. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.* 33 (2015), 3–29. https://doi.org/10.1016/j.websem.2015.04.002

[21] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris, and Gerardo I. Simari. 2015. From Classical to Consistent Query Answering under Existential Rules. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, Blai Bonet and Sven Koenig (Eds.). AAAI Press, 1546–1552. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9817

[22] Carsten Lutz and Frank Wolter. 2015. On the Relationship between Consistent Query Answering and Constraint Satisfaction Problems. In *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium (LIPIcs)*, Marcelo Arenas and Martín Ugarte (Eds.), Vol. 31. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 363–379. https://doi.org/10.4230/LIPIcs.ICDT.2015.363

[23] Marco Manna, Francesco Ricca, and Giorgio Terracina. 2015. Taming primary key violations to query large inconsistent data via ASP. *TPLP* 15, 4-5 (2015), 696–710. https://doi.org/10.1017/S1471068415000320

[24] Mónica Caniupán Marileo and Leopoldo E. Bertossi. 2007. The Consistency Extractor System: Querying Inconsistent Databases Using Answer Set Programs. In *Scalable Uncertainty Management, First International Conference, SUM 2007, Washington, DC, USA, October 10-12, 2007, Proceedings (Lecture Notes in Computer Science)*, Henri Prade and V. S. Subrahmanian (Eds.), Vol. 4772. Springer, 74–88. https://doi.org/10.1007/978-3-540-75410-7_6

[25] Dany Maslowski and Jef Wijsen. 2013. A dichotomy in the complexity of counting database repairs. *J. Comput. Syst. Sci.* 79, 6 (2013), 958–983. https://doi.org/10.1016/j.jcss.2013.01.011

[26] Dany Maslowski and Jef Wijsen. 2014. Counting Database Repairs that Satisfy Conjunctive Queries with Self-Joins, See [27], 155–164. https://doi.org/10.5441/002/icdt.2014.18

[27] Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy (Eds.). 2014. *Proc. 17th International Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014*. OpenProceedings.org. http://openproceedings.org/edbticdt2014/ICDT_toc.html

[28] Jef Wijsen. 2010. On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*, Jan Paredaens and Dirk Van Gucht (Eds.). ACM, 179–190. https://doi.org/10.1145/1807085.1807111

[29] Jef Wijsen. 2012. Certain conjunctive query answering in first-order logic. *ACM Trans. Database Syst.* 37, 2 (2012), 9:1–9:35. https://doi.org/10.1145/2188349.2188351