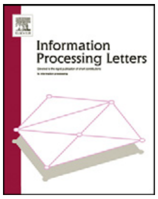
Information Processing Letters 110 (2010) 950–955

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl



A remark on the complexity of consistent conjunctive query answering under primary key violations

Jef Wijsen

Université de Mons (UMONS), Place du Parc 20, B-7000 Mons, Belgium

ARTICLE INFO

Article history:
Received 5 March 2010
Received in revised form 30 May 2010
Accepted 21 July 2010
Available online 3 August 2010
Communicated by J. Chomicki

Keywords:
Databases
Consistent query answering
Database repairing

ABSTRACT

A natural way for capturing uncertainty in the relational data model is by allowing relations that violate their primary key. A repair of such relation is obtained by selecting a maximal number of tuples without ever selecting two tuples that agree on their primary key. Given a Boolean query q, CERTAINTY(q) is the problem that takes as input a relational database and asks whether q evaluates to true on every repair of that database. In recent years, CERTAINTY(q) has been studied primarily for conjunctive queries. Conditions have been determined under which CERTAINTY(q) is **coNP**-complete, first-order expressible, or not first-order expressible. A remaining open question was whether there exist conjunctive queries q without self-join such that CERTAINTY(q) is in **PTIME** but not first-order expressible. We answer this question affirmatively.

1. Introduction

Uncertainty can be captured in the relational data model by allowing two or more tuples that agree on their primary key. In this way, we model that only one of the tuples can be true, but we do not know which one. If **db** is a database that violates primary key constraints, then a *repair* of **db** is obtained by selecting a maximal number of tuples without ever selecting two tuples that agree on their primary key. As observed in [1, p. 98], the number of repairs can be exponential in the size of **db**. Then, given a Boolean query q, rather than asking whether q evaluates to true on **db**, we ask whether q evaluates to true on *every repair* of **db**. This setting is a special case of the general framework introduced in [2].

It is well-known that the data complexity of evaluating a first-order query q on a single database **db** is in the low complexity class $\mathbf{AC}^0$. An interesting and important question is: How does this complexity change if we want to know whether q evaluates to true on every repair of **db**? To study this question, let CERTAINTY(q) be the set of databases (over some fixed schema) such that CERTAINTY(q) contains **db** if and only if q evaluates to true on every repair of **db**.

It was shown in [1, p. 103] that CERTAINTY(q_1) is **coNP**-complete for the conjunctive query $q_1 = \exists x \exists y \exists z (T(\underline{x}, y, a) \wedge T(\underline{z}, y, b))$, where the primary key of T is the first coordinate of T. Throughout this article, primary key coordinates will always be underlined.

On the other hand, there exist conjunctive queries q such that CERTAINTY(q) is first-order expressible (and hence in the low complexity class $\mathbf{AC}^0$). For example, it is easy to see that for the query $q_2 = \exists x R(\underline{x}, a)$, the set CERTAINTY($q_2$) coincides with the set of databases satisfying $\varphi_2 = \exists x (R(\underline{x}, a) \wedge \forall y (R(\underline{x}, y) \rightarrow y = a))$. Thus, in order to decide whether q_2 evaluates to true on every repair of some database **db**, it suffices to evaluate φ_2 on the (possibly inconsistent) database **db**, without the need for computing repairs. Since φ_2 is first-order, it can be encoded in SQL.

It is known that **coNP**-complete sets are not first-order expressible. But we do not have to mount up to intractability to get first-order inexpressibility: there exist conjunctive queries q such that CERTAINTY(q) is in **PTIME** but not first-order expressible. The first example of this was the conjunctive query with inequality $q_3 = \exists x \exists y \exists z (R(\underline{x}, z) \wedge R(\underline{y}, z) \wedge x \neq y)$ found in [3]. An example

E-mail address: jef.wijsen@umons.ac.be.

0020-0190/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2010.07.021

without inequality appeared in [4], where it was shown that for the query $q_4 = \exists x \exists y (R(\underline{x}, y) \wedge R(\underline{y}, a))$, the set $\mathsf{CERTAINTY}(q_4)$ is in **PTIME** but not first-order expressible. It should be noticed that both q_3 and q_4 contain a self-join, i.e. the same relation name occurs more than once in these queries.

Fuxman and Miller [5] claim a dichotomy result for a class of conjunctive queries without self-join: for every query q in that class, either $\mathsf{CERTAINTY}(q)$ is first-order expressible or **coNP**-complete. At the end of that article [5, p. 634], they state as an interesting open problem whether there are conjunctive queries without self-join such that $\mathsf{CERTAINTY}(q)$ is in **PTIME** but not first-order expressible. In this article, we answer this question affirmatively. For the conjunctive query $q_0 = \exists x \exists y (R(\underline{x}, y) \wedge S(\underline{y}, x))$, we show that $\mathsf{CERTAINTY}(q_0)$ is in **PTIME** but not first-order expressible.

We point out that Fuxman and Miller claim the **coNP**-completeness of $\mathsf{CERTAINTY}(q_0)$ [5, p. 629], but that result is based on a wrong argumentation and is refuted by our results.

This article is organized as follows. Section 2 gives preliminary definitions. Section 3 shows that $\mathsf{CERTAINTY}(q_0)$ is in **PTIME**. Section 4 shows that $\mathsf{CERTAINTY}(q_0)$ is not first-order expressible. Section 5 concludes.

2. Preliminaries

We assume denumerably many *relation names*. Each relation name R has a unique *signature*, which is a pair $[n, k]$ where $1 \leqslant k \leqslant n$: the integer n denotes the arity of R, and the set $\{1, 2, \ldots, k\}$ of coordinates is the *primary key* of R. If R is a relation name with signature $[n, k]$ and $a_1, \ldots, a_n$ are constants, then $R(a_1, \ldots, a_n)$ is an R-*fact* (or simply fact). It is common to underline the coordinates of the primary key, as in $R(\underline{a_1, \ldots, a_k}, a_{k+1}, \ldots, a_n)$. Uppercase letters A, B will be used to denote facts. Two facts are *key-equal* if they share the same relation name and agree on all coordinates of the primary key of that relation name. Every fact is key-equal to itself.

A *schema* is a finite set of relation names. The following definitions are relative to a fixed schema. A *database* is a finite set of facts using only the relation names of the schema. A database is *consistent* if it contains no two distinct facts that are key-equal. If $A \in \mathbf{db}$, then $[\![A]\!]_{\mathbf{db}}$ denotes the set of all facts of $\mathbf{db}$ that are key-equal to A. Thus, a database $\mathbf{db}$ is consistent if and only if for each $A \in \mathbf{db}$, $[\![A]\!]_{\mathbf{db}}$ is the singleton $\{A\}$. A *repair* of a (not necessarily consistent) database $\mathbf{db}$ is a maximal consistent subset of $\mathbf{db}$.

A *Boolean conjunctive query* is a first-order sentence of the form:

$$\exists^* \big(R_1(\vec{x}_1) \wedge R_2(\vec{x}_2) \wedge \cdots \wedge R_m(\vec{x}_m) \big)$$

where each R_i is a relation name of the schema and $\vec{x}_i$ is a sequence of variables and constants whose length is the arity of R_i. This query is *without self-join* if $i \neq j$ implies $R_i \neq R_j$ $(1 \leqslant i, j \leqslant m)$. Satisfaction of Boolean queries is defined in the standard way (and omitted here).

Given a Boolean conjunctive query q, we define $\mathsf{CERTAINTY}(q)$ as the set of databases (over the fixed schema) containing database $\mathbf{db}$ if and only if every repair of $\mathbf{db}$ satisfies q. The set $\mathsf{CERTAINTY}(q)$ is said to be *first-order expressible* if there exists a first-order sentence φ such that for all databases $\mathbf{db}$, $\mathbf{db} \in \mathsf{CERTAINTY}(q)$ if and only if $\mathbf{db}$ satisfies φ. Such formula φ, if it exists, has also been called a consistent first-order rewriting for q.

We assume two relation names R and S, both of signature $[2, 1]$. The Boolean conjunctive query q_0 is defined by:

$$q_0 = \exists x \exists y \big(R(\underline{x}, y) \wedge S(\underline{y}, x) \big).$$

If A is the fact $R(\underline{a}, b)$, then $-A$ denotes the fact $S(\underline{b}, a)$. Conversely, if A is the fact $S(\underline{c}, d)$, then $-A$ denotes the fact $R(\underline{d}, c)$. Clearly, since A and $-A$ contain different relation names, they cannot be key-equal. Notice that $-(-A) = A$.

The following lemma is obvious.

Lemma 1. *A database* $\mathbf{db}$ *satisfies* q_0 *if and only if for some fact* $A \in \mathbf{db}$*, we also have* $-A \in \mathbf{db}$*.*

3. CERTAINTY(q_0) is in PTIME

3.1. Every fact conflicting

From now on, the fixed schema is $\{R, S\}$. In this section, we show that a database $\mathbf{db}$ has a repair that falsifies q_0 if for every fact $A \in \mathbf{db}$, $[\![A]\!]_{\mathbf{db}}$ contains at least two facts. The following definition introduces the construct of secondary repair sequence; the notion of primary repair sequence will be defined in Section 3.2.

Definition 1. Let $\mathbf{db}$ and $\mathbf{r}$ be databases (over schema $\{R, S\}$). We write $(\mathbf{db}, \mathbf{r}) \xrightarrow{A} (\mathbf{db}', \mathbf{r}')$ if the following conditions hold:

1. A is a fact of $\mathbf{db}$ satisfying $[\![A]\!]_{\mathbf{db}} = \{A\}$, if such fact exists; if no such fact exists, then A is an arbitrary fact of $\mathbf{db}$.
2. $\mathbf{db}' = (\mathbf{db} \setminus [\![A]\!]_{\mathbf{db}}) \setminus \{-A\}$ and $\mathbf{r}' = \mathbf{r} \cup \{A\}$.

A *secondary repair sequence* for $\mathbf{db}$ is then a maximal sequence

$$(\mathbf{db}_0, \mathbf{r}_0) \xrightarrow{A_1} (\mathbf{db}_1, \mathbf{r}_1) \xrightarrow{A_2} (\mathbf{db}_2, \mathbf{r}_2) \cdots \xrightarrow{A_n} (\mathbf{db}_n, \mathbf{r}_n)$$

where $\mathbf{db}_0 = \mathbf{db}$ and $\mathbf{r}_0 = \{\}$. Clearly, the maximality condition implies $\mathbf{db}_n = \{\}$.

Example 1. Let $\mathbf{db} = \{R(\underline{a}, 1), R(\underline{a}, 2), R(\underline{b}, 2), R(\underline{b}, 3), S(\underline{1}, a), S(\underline{1}, b)\}$. Since every fact of $\mathbf{db}$ is key-equal to another fact of $\mathbf{db}$, the construction of a secondary repair sequence starts with selecting an arbitrary fact of $\mathbf{db}$; let $R(\underline{a}, 1)$ be the selected fact. We obtain $(\mathbf{db}, \{\}) \xrightarrow{R(\underline{a},1)} (\mathbf{db}_1, \mathbf{r}_1)$ with $\mathbf{db}_1 = \{R(\underline{b}, 2), R(\underline{b}, 3), S(\underline{1}, b)\}$ and $\mathbf{r}_1 = \{R(\underline{a}, 1)\}$.

Since $[\![S(\underline{1}, b)]\!]_{\mathbf{db}_1} = \{S(\underline{1}, b)\}$, the construction proceeds with $(\mathbf{db}_1, \mathbf{r}_1) \xrightarrow{S(\underline{1},b)} (\mathbf{db}_2, \mathbf{r}_2)$ with $\mathbf{db}_2 = \{R(\underline{b}, 2), R(\underline{b}, 3)\}$ and $\mathbf{r}_2 = \{R(\underline{a}, 1), S(\underline{1}, b)\}$.

Next, either fact of $\mathbf{db}_2$ can be selected. If $R(\underline{b}, 2)$ is selected, we obtain $(\mathbf{db}_2, \mathbf{r}_2) \xrightarrow{R(\underline{b},2)} (\mathbf{db}_3, \mathbf{r}_3)$ with $\mathbf{db}_3 = \{\}$ and $\mathbf{r}_3 = \{R(\underline{a}, 1), S(\underline{1}, b), R(\underline{b}, 2)\}$. This finishes the construction.

We now provide a couple of properties of secondary repair sequences.

Lemma 2. *Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \xrightarrow{A_1} (\mathbf{db}_1, \mathbf{r}_1) \cdots \xrightarrow{A_n} (\mathbf{db}_n, \mathbf{r}_n)$$

be a secondary repair sequence for database $\mathbf{db}$. *Let* $0 \leqslant i \leqslant n$. *Then,*

1. *for all* $A \in \mathbf{db}_i$ *and* $B \in \mathbf{r}_i$, *A and B are not key-equal*;
2. *for all* $A \in \mathbf{db}_i$ *and* $B \in \mathbf{r}_i$, $A \neq -B$;
3. $\mathbf{r}_i$ *is consistent*; *and*
4. $\mathbf{r}_i$ *falsifies* q_0.

Proof. The proof is straightforward using induction on increasing i. □

Lemma 3. *Let* $\mathbf{db}$ *be a database such that for each* $A \in \mathbf{db}$, $\llbracket A \rrbracket_{\mathbf{db}}$ *contains two or more facts. Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \xrightarrow{A_1} (\mathbf{db}_1, \mathbf{r}_1) \cdots \xrightarrow{A_n} (\mathbf{db}_n, \mathbf{r}_n)$$

be a secondary repair sequence for $\mathbf{db}$. *For* $0 \leqslant i \leqslant n$,

1. *there is at most one fact in* $\mathbf{db}_i$ *satisfying* $\llbracket A \rrbracket_{\mathbf{db}_i} = \{A\}$; *and*
2. *every fact of* $\mathbf{db}$ *is key-equal to some fact in* $\mathbf{db}_i \cup \mathbf{r}_i$.

Proof. Proof by induction on increasing i. The base case $i = 0$ is trivial for both (1) and (2). For the induction step $i - 1 \to i$, let

$$\mathbf{db}_i = (\mathbf{db}_{i-1} \setminus \llbracket A_i \rrbracket_{\mathbf{db}_{i-1}}) \setminus \{-A_i\} \quad \text{and}$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} \cup \{A_i\},$$

where A_i is a fact of $\mathbf{db}_{i-1}$ satisfying $\llbracket A_i \rrbracket_{\mathbf{db}_{i-1}} = \{A_i\}$, if such fact exists; otherwise A_i is any fact of $\mathbf{db}_{i-1}$. The proof of the induction step is straightforward for (1). We next show the induction step for (2).

(2) By the induction hypothesis, every fact of $\mathbf{db}$ is key-equal to some fact of $\mathbf{db}_{i-1} \cup \mathbf{r}_{i-1}$. It suffices to show that every fact of $\mathbf{db}_{i-1} \cup \mathbf{r}_{i-1}$ is key-equal to some fact of $\mathbf{db}_i \cup \mathbf{r}_i$. Let $A \in \mathbf{db}_{i-1} \cup \mathbf{r}_{i-1}$. We distinguish three cases:

- $A \in \llbracket A_i \rrbracket_{\mathbf{db}_{i-1}}$. Then A is key-equal to A_i. Hence, A is key-equal to some fact in $\mathbf{r}_i$.
- $A = -A_i$. From item (2) in Lemma 2, it follows $-A_i \in \mathbf{db}_{i-1}$. By item (1) of the current proof, the set $\{B \in \mathbf{db}_{i-1} \mid \llbracket B \rrbracket_{\mathbf{db}_{i-1}} = \{B\}\}$ is either the empty set or the singleton $\{A_i\}$. From $A_i \neq -A_i$, it follows that $\llbracket -A_i \rrbracket_{\mathbf{db}_{i-1}}$ must contain at least two distinct facts. Hence, we can assume $B \in \llbracket -A_i \rrbracket_{\mathbf{db}_{i-1}}$ with $B \neq -A_i$. Since $B \in \mathbf{db}_i$, A is key-equal to some fact in $\mathbf{db}_i$.
- $A \notin \llbracket A_i \rrbracket_{\mathbf{db}_{i-1}}$ and $A \neq -A_i$. Then, $A \in \mathbf{db}_i$. □

Corollary 1. *Let* $\mathbf{db}$ *be a database such that for each* $A \in \mathbf{db}$, $\llbracket A \rrbracket_{\mathbf{db}}$ *contains two or more facts. Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \xrightarrow{A_1} (\mathbf{db}_1, \mathbf{r}_1) \cdots \xrightarrow{A_n} (\mathbf{db}_n, \mathbf{r}_n)$$

be a secondary repair sequence for $\mathbf{db}$. *Then,* $\mathbf{r}_n$ *is a repair of* $\mathbf{db}$ *that falsifies* q_0.

Proof. By Lemma 2, $\mathbf{r}_n$ is consistent and falsifies q_0. By Lemma 3 and since $\mathbf{db}_n = \{\}$, every fact of $\mathbf{db}$ is key-equal to some fact of $\mathbf{r}_n$. It follows that $\mathbf{r}_n$ is a repair of $\mathbf{db}$. □

3.2. Polynomial time algorithm

The construct of primary repair sequence is defined next; it differs in two respects from secondary repair sequences defined earlier. First, if some database $\mathbf{db}_i$ has been constructed so far and every fact of $\mathbf{db}_i$ is key-equal to another fact in $\mathbf{db}_i$, then the primary repair sequence halts; in this situation, a secondary repair sequence can continue with selecting an arbitrary fact of $\mathbf{db}_i$. Second, in a primary repair sequence, a fact $A \in \mathbf{db}_i$ cannot be removed if both $\llbracket A \rrbracket_{\mathbf{db}_i}$ and $\llbracket -A \rrbracket_{\mathbf{db}_i}$ are singletons.

Definition 2. Let $\mathbf{db}$ and $\mathbf{r}$ be databases (over schema $\{R, S\}$). We write $(\mathbf{db}, \mathbf{r}) \overset{A}{\rightsquigarrow} (\mathbf{db}', \mathbf{r}')$ if the following conditions hold:

1. $A \in \mathbf{db}$ such that $\llbracket A \rrbracket_{\mathbf{db}} = \{A\}$;
2. if $-A \in \mathbf{db}$, then $\llbracket -A \rrbracket_{\mathbf{db}}$ contains two or more facts; and
3. $\mathbf{db}' = \mathbf{db} \setminus \{A, -A\}$ and $\mathbf{r}' = \mathbf{r} \cup \{A\}$.

A *primary repair sequence* for $\mathbf{db}$ is a maximal sequence

$$(\mathbf{db}_0, \mathbf{r}_0) \overset{A_1}{\rightsquigarrow} (\mathbf{db}_1, \mathbf{r}_1) \overset{A_2}{\rightsquigarrow} (\mathbf{db}_2, \mathbf{r}_2) \cdots \overset{A_n}{\rightsquigarrow} (\mathbf{db}_n, \mathbf{r}_n)$$

where $\mathbf{db}_0 = \mathbf{db}$ and $\mathbf{r}_0 = \{\}$.

Example 2. Let $\mathbf{db} = \{R(\underline{a}, b), R(\underline{c}, b), S(\underline{b}, a), S(\underline{b}, c)\}$. The construction of a primary repair sequence can start with selecting $R(\underline{a}, b) \in \mathbf{db}$ because $\llbracket R(\underline{a}, b) \rrbracket_{\mathbf{db}}$ is a singleton and $\llbracket S(\underline{b}, a) \rrbracket_{\mathbf{db}}$ is not a singleton. We obtain $(\mathbf{db}, \{\}) \overset{R(\underline{a},b)}{\rightsquigarrow} (\mathbf{db}_1, \{R(\underline{a}, b)\})$ where $\mathbf{db}_1 = \{R(\underline{c}, b), S(\underline{b}, c)\}$. Since $R(\underline{c}, b) = -S(\underline{b}, c)$ and since both $\llbracket R(\underline{c}, b) \rrbracket_{\mathbf{db}_1}$ and $\llbracket S(\underline{b}, c) \rrbracket_{\mathbf{db}_1}$ are singletons, the construction halts.

We now show a number of properties of primary repair sequences. The ultimate result is that to decide $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$, we construct a primary repair sequence for $\mathbf{db}$. Let $(\mathbf{db}_n, \mathbf{r}_n)$ be the last element in this sequence. Then, $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$ if and only if for some $A \in \mathbf{db}_n$, $\llbracket A \rrbracket_{\mathbf{db}_n}$ is a singleton.

Lemma 4. *Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \overset{A_1}{\rightsquigarrow} (\mathbf{db}_1, \mathbf{r}_1) \cdots \overset{A_n}{\rightsquigarrow} (\mathbf{db}_n, \mathbf{r}_n)$$

be a primary repair sequence for database $\mathbf{db}$. *For* $0 \leqslant i \leqslant n$,

1. *for all* $A \in \mathbf{db}_i$ *and* $B \in \mathbf{r}_i$, *A and B are not key-equal*;

2. *for all* $A \in \mathbf{db}_i$ *and* $B \in \mathbf{r}_i$, $A \neq -B$;
3. $\mathbf{r}_i$ *is consistent*;
4. $\mathbf{r}_i$ *falsifies* q_0;
5. *every fact of* $\mathbf{db}$ *is key-equal to some fact in* $\mathbf{db}_i \cup \mathbf{r}_i$; *and*
6. *for every repair* $\mathbf{rep}$ *of* $\mathbf{db}$ *such that* $\mathbf{rep}$ *falsifies* q_0,
   6.1. $\mathbf{r}_i \subseteq \mathbf{rep}$; *and*
   6.2. $\mathbf{rep}$ *contains a repair of* $\mathbf{db}_i$.

Proof. Proof by induction on increasing i. The base case $i = 0$ is trivial for all items (1)–(6). For the induction step $i-1 \to i$, let

$$\mathbf{db}_i = \mathbf{db}_{i-1} \setminus \{A_i, -A_i\} \quad \text{and}$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} \cup \{A_i\},$$

where A_i is a fact of $\mathbf{db}_{i-1}$ satisfying $\llbracket A_i \rrbracket_{\mathbf{db}_{i-1}} = \{A_i\}$ and either $-A_i \notin \mathbf{db}_{i-1}$ or $\llbracket -A_i \rrbracket_{\mathbf{db}_{i-1}}$ contains at least two facts. The induction steps for (1)–(4) are straightforward and similar to corresponding items in Lemma 2. The induction step for (5) follows from the fact that if $-A_i \in \mathbf{db}_{i-1}$, then $\llbracket -A_i \rrbracket_{\mathbf{db}_{i-1}}$ contains at least two facts, hence $\mathbf{db}_i$ contains a fact that is key-equal to $-A_i$. Finally, we show the induction step for (6).

(6) Let $\mathbf{rep}$ be a repair of $\mathbf{db}$ such that $\mathbf{rep}$ falsifies q_0. By the induction hypothesis, $\mathbf{r}_{i-1} \subseteq \mathbf{rep}$ and $\mathbf{rep}$ contains a repair of $\mathbf{db}_{i-1}$. Since $A_i \in \mathbf{db}_{i-1}$ and $\llbracket A_i \rrbracket_{\mathbf{db}_{i-1}} = \{A_i\}$, every repair of $\mathbf{db}_{i-1}$ must contain A_i. It follows $A_i \in \mathbf{rep}$. Consequently, $\mathbf{r}_i = \mathbf{r}_{i-1} \cup \{A_i\} \subseteq \mathbf{rep}$.

Since $\mathbf{rep}$ contains a repair of $\mathbf{db}_{i-1}$ and $\llbracket A_i \rrbracket_{\mathbf{db}_{i-1}} = \{A_i\}$, it follows that $\mathbf{rep}$ must contain a repair of $\mathbf{db}_{i-1} \setminus \{A_i\}$. Moreover, since $A_i \in \mathbf{rep}$ (preceding paragraph) and $\mathbf{rep}$ falsifies q_0, it must be the case that $-A_i \notin \mathbf{rep}$. Consequently, $\mathbf{rep}$ contains a repair of $\mathbf{db}_{i-1} \setminus \{A_i, -A_i\} = \mathbf{db}_i$. □

Lemma 5. *Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \overset{A_1}{\rightsquigarrow} (\mathbf{db}_1, \mathbf{r}_1) \cdots \overset{A_n}{\rightsquigarrow} (\mathbf{db}_n, \mathbf{r}_n)$$

be a primary repair sequence for database $\mathbf{db}$. *One of the following conditions is true*:

1. $\mathbf{db}_n = \{\}$;
2. $\mathbf{db}_n \neq \{\}$ *and for all* $A \in \mathbf{db}_n$, $\llbracket A \rrbracket_{\mathbf{db}_n}$ *contains two or more facts*; *or*
3. *there exists* $A \in \mathbf{db}_n$ *such that* $\llbracket A \rrbracket_{\mathbf{db}_n} = \{A\}$, $-A \in \mathbf{db}_n$, *and* $\llbracket -A \rrbracket_{\mathbf{db}_n} = \{-A\}$.

Proof. Assume (1) and (2) are false. Then, there exists $A \in \mathbf{db}_n$ such that $\llbracket A \rrbracket_{\mathbf{db}_n} = \{A\}$. Since the primary repair sequence is maximal, condition (2) in Definition 2 must be false. Consequently, $-A \in \mathbf{db}_n$ and $\llbracket -A \rrbracket_{\mathbf{db}_n} = \{-A\}$. □

Theorem 1. *Let*

$$\big(\mathbf{db}, \{\}\big) = (\mathbf{db}_0, \mathbf{r}_0) \overset{A_1}{\rightsquigarrow} (\mathbf{db}_1, \mathbf{r}_1) \cdots \overset{A_n}{\rightsquigarrow} (\mathbf{db}_n, \mathbf{r}_n)$$

be a primary repair sequence for database $\mathbf{db}$. *Then*, $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$ *if and only if* $\llbracket A \rrbracket_{\mathbf{db}_n} = \{A\}$ *for some* $A \in \mathbf{db}_n$.

Proof. ($\Rightarrow$) By contraposition. Assume that for all $A \in \mathbf{db}_n$, $\llbracket A \rrbracket_{\mathbf{db}_n}$ contains at least two facts. By Corollary 1, we can assume a repair $\mathbf{rep}'$ of $\mathbf{db}_n$ such that $\mathbf{rep}'$ falsifies q_0. By items (3) and (1) in Lemma 4, $\mathbf{r}_n$ is consistent, and for all $A \in \mathbf{r}_n$ and $B \in \mathbf{db}_n$, A and B are not key-equal. Since $\mathbf{rep}' \subseteq \mathbf{db}_n$, it follows that no fact of $\mathbf{r}_n$ is key-equal to some fact of $\mathbf{rep}'$. Since $\mathbf{rep}'$ is consistent, it follows that $\mathbf{r}_n \cup \mathbf{rep}'$ is consistent. By item (5) in Lemma 4, every fact of $\mathbf{db}$ is key-equal to some fact of $\mathbf{r}_n \cup \mathbf{db}_n$. Since $\mathbf{rep}'$ is a repair of $\mathbf{db}_n$, every fact of $\mathbf{db}_n$ is key-equal to some fact in $\mathbf{rep}'$. It follows that every fact of $\mathbf{db}$ is key-equal to some fact of $\mathbf{r}_n \cup \mathbf{rep}'$. Consequently, $\mathbf{r}_n \cup \mathbf{rep}'$ is a repair of $\mathbf{db}$. By items (4) and (2) of Lemma 4, $\mathbf{r}_n$ falsifies q_0, and for all $A \in \mathbf{r}_n$ and $B \in \mathbf{rep}'$, $A \neq -B$. It follows $\mathbf{r}_n \cup \mathbf{rep}'$ falsifies q_0. Consequently, $\mathbf{db} \notin \mathsf{CERTAINTY}(q_0)$.

($\Leftarrow$) Assume that for some $A \in \mathbf{db}_n$, we have $\llbracket A \rrbracket_{\mathbf{db}_n} = \{A\}$. By Lemma 5, we can assume $A \in \mathbf{db}_n$ such that $\llbracket A \rrbracket_{\mathbf{db}_n} = \{A\}$, $-A \in \mathbf{db}_n$, and $\llbracket -A \rrbracket_{\mathbf{db}_n} = \{-A\}$. Assume $\mathbf{db} \notin \mathsf{CERTAINTY}(q_0)$. We can assume a repair $\mathbf{rep}$ of $\mathbf{db}$ such that $\mathbf{rep}$ falsifies q_0. By item (6) of Lemma 4, $\mathbf{rep}$ contains a repair of $\mathbf{db}_n$. Consequently, $\{A, -A\} \subseteq \mathbf{rep}$. But then $\mathbf{rep}$ satisfies q_0, a contradiction. We conclude by contradiction $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$. □

Theorem 1 immediately leads to the following quadratic time algorithm for deciding $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$. Let N be the number of facts in $\mathbf{db}$. First, database $\mathbf{db}$ is sorted in time $\mathcal{O}(N \log N)$ such that key-equal facts are grouped together. The algorithm then starts the construction of a primary repair sequence for $\mathbf{db}$. Every transition $(\mathbf{db}_{i-1}, \mathbf{r}_{i-1}) \overset{A_i}{\rightsquigarrow} (\mathbf{db}_i, \mathbf{r}_i)$ goes as follows. Database $\mathbf{db}_{i-1}$ is scanned once to find a fact A_i satisfying $\llbracket A_i \rrbracket_{\mathbf{db}_{i-1}} = \{A_i\}$.

- If no such A_i is found, then $(\mathbf{db}_{i-1}, \mathbf{r}_{i-1})$ is the last element in the primary repair sequence; the algorithm outputs $\mathbf{db} \notin \mathsf{CERTAINTY}(q_0)$ and halts.
- If such A_i is found, $\mathbf{db}_{i-1}$ is scanned a second time to determine whether $-A_i \in \mathbf{db}_{i-1}$ and $\llbracket -A_i \rrbracket_{\mathbf{db}_{i-1}} = \{-A_i\}$. If this is the case, then the algorithm outputs $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$ and halts, because A_i and $-A_i$ cannot be deleted by later transitions. Otherwise, i.e. if $-A_i \notin \mathbf{db}_{i-1}$ or if $-A_i \in \mathbf{db}_{i-1}$ and $\llbracket -A_i \rrbracket_{\mathbf{db}_i}$ contains two or more facts, then $\mathbf{db}_i$ and $\mathbf{r}_i$ are constructed as $\mathbf{db}_i = \mathbf{db}_{i-1} \setminus \{A_i, -A_i\}$ and $\mathbf{r}_i = \mathbf{r}_{i-1} \cup \{A_i\}$.

Thus, each transition can be done in linear time, by two database scans. Since the number of $\overset{A_i}{\rightsquigarrow}$-transitions in a primary repair sequence is at most the cardinality of $\mathbf{db}$, it follows that deciding membership of $\mathsf{CERTAINTY}(q_0)$ can be done with quadratic time data complexity. Thus we have the following result.

Corollary 2. $\mathsf{CERTAINTY}(q_0)$ *is in* **PTIME**.

4. Inexpressibility result

The proof of the following theorem is based on Hanf-locality of first-order logic. It relies on constructs defined in Chapter 4 of [6]. We prefer not to copy–paste these definitions here, because they are rather lengthy, and the treatment in [6] is excellent.

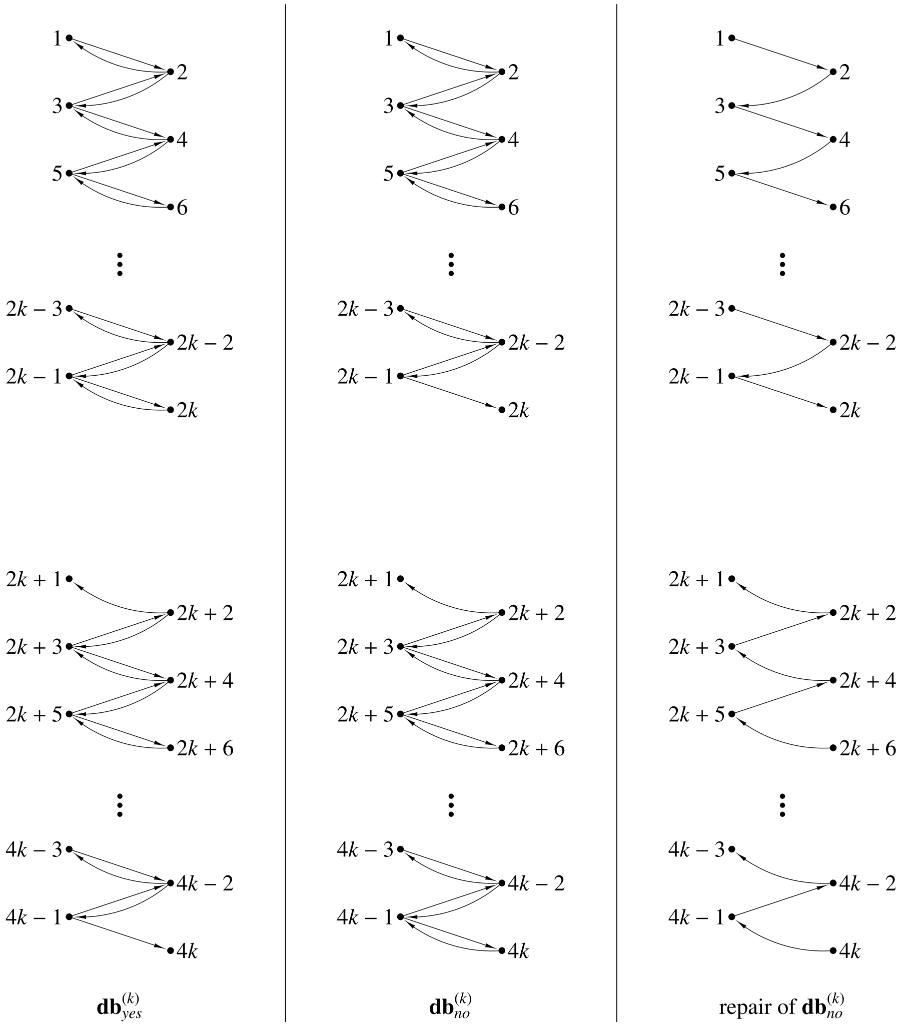


Fig. 1. Graphical representation of the databases $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$. Straight edges pointing to the right represent R-facts. Curved edges pointing to the left represent S-facts. The rightmost diagram shows a repair of $\mathbf{db}^{(k)}_{no}$ that falsifies q_0.

Theorem 2. $\mathsf{CERTAINTY}(q_0)$ *is not first-order expressible.*

Proof. Assume to the contrary that φ is a first-order sentence such that for every database **db**, $\mathbf{db} \in \mathsf{CERTAINTY}(q_0)$ if and only if $\mathbf{db} \models \varphi$. From Theorem 4.12 in [6], it follows that φ is Hanf-local. Let $d \geqslant 0$ be the Hanf-locality rank of φ. Choose integer k such that $k > d$ and $k \geqslant 2$, and take two databases $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$ as shown in Fig. 1, where straight edges represent R-facts, and curved edges represent S-facts. Notice that all straight edges point from left to right, and all curved edges point from right to left. Databases $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$ contain the same R-facts. The S-fact $S(\underline{2k}, 2k-1)$ is in $\mathbf{db}^{(k)}_{yes}$ but not in $\mathbf{db}^{(k)}_{no}$, while $S(\underline{4k}, 4k-1)$ is in $\mathbf{db}^{(k)}_{no}$ but not in $\mathbf{db}^{(k)}_{yes}$.

The property $\mathbf{db}^{(k)}_{yes} \in \mathsf{CERTAINTY}(q_0)$ can be directly verified on the graph of $\mathbf{db}^{(k)}_{yes}$ in Fig. 1 (left): in the upper connected component (i.e. the component with vertices $1, 2, \ldots, 2k$), it is impossible to choose one outgoing edge from each vertex without creating a cycle of size 2. On the other hand, Fig. 1 (right) shows a repair of $\mathbf{db}^{(k)}_{no}$ that falsifies q_0, hence $\mathbf{db}^{(k)}_{no} \notin \mathsf{CERTAINTY}(q_0)$.

Let $dom^{(k)} = \{1, 2, \ldots, 4k\}$. Clearly, the active domains of $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$ are both equal to $dom^{(k)}$. The definitions of radius d ball and d-neighborhood can be found in [6]. For every $i \in dom^{(k)}$, the radius d ball around i in both $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$ is given by:

$$B_d(i) = \begin{cases} [i-d, i+d] \cap [1, 2k] & \text{if } i \in [1, 2k], \\ [i-d, i+d] \cap [2k+1, 4k] & \text{if } i \in [2k+1, 4k]. \end{cases}$$

Since $k > d$, there is no $i \in dom^{(k)}$ such that $B_d(i)$ contains both 1 and $2k$. Likewise, no $B_d(i)$ contains both $2k+1$ and $4k$.

Let $f : dom^{(k)} \to dom^{(k)}$ be the bijection defined by Table 1; for every $i \in dom^{(k)}$ that does not occur in the first column, let $f(i) = i$. It can be easily verified that for each $i \in dom^{(k)}$, the d-neighborhood of i in $\mathbf{db}^{(k)}_{yes}$ is isomorphic to the d-neighborhood of $f(i)$ in $\mathbf{db}^{(k)}_{no}$. Since d is the Hanf-locality rank of φ, $\mathbf{db}^{(k)}_{yes}$ and $\mathbf{db}^{(k)}_{no}$ must agree on φ, but

Table 1
Elements where f is not the identity.

i	$f(i)$
$2k-d$	$4k-d$
$2k-d+1$	$4k-d+1$
$\vdots$	$\vdots$
$2k-1$	$4k-1$
$2k$	$4k$
$4k-d$	$2k-d$
$4k-d+1$	$2k-d+1$
$\vdots$	$\vdots$
$4k-1$	$2k-1$
$4k$	$2k$

$\mathbf{db}_{yes}^{(k)} \in \mathsf{CERTAINTY}(q_0)$, and $\mathbf{db}_{no}^{(k)} \notin \mathsf{CERTAINTY}(q_0)$. We conclude by contradiction that $\mathsf{CERTAINTY}(q_0)$ is not first-order expressible. □

5. Conclusion

We showed that for the query $q_0 = \exists x \exists y (R(\underline{x}, y) \wedge S(\underline{y}, x))$, the set $\mathsf{CERTAINTY}(q_0)$ is in **PTIME** but not first-order expressible. In particular, deciding membership of $\mathsf{CERTAINTY}(q_0)$ can be done with quadratic time data complexity. In this way, we answered an interesting open question raised in [5].

Our result provides a useful new insight in the practice of consistent query answering under primary keys. It implies that first-order logic is not sufficiently expressive to capture all tractable instantiations of $\mathsf{CERTAINTY}(q)$ with q conjunctive and without self-join.

References

[1] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Inform. Comput. 197 (1–2) (2005) 90–121.

[2] M. Arenas, L.E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: PODS, ACM Press, 1999, pp. 68–79.

[3] A. Fuxman, R.J. Miller, Towards inconsistency management in data integration systems, in: S. Kambhampati, C.A. Knoblock (eds.), IIWeb, 2003, pp. 143–148.

[4] J. Wijsen, On the consistent rewriting of conjunctive queries under primary key constraints, Inf. Syst. 34 (7) (2009) 578–601.

[5] A. Fuxman, R.J. Miller, First-order query rewriting for inconsistent databases, J. Comput. System Sci. 73 (4) (2007) 610–635.

[6] L. Libkin, Elements of Finite Model Theory, Springer, 2004.