Corrigendum to *"Counting Database Repairs that Satisfy Conjunctive Queries with Self-Joins"*

Jef Wijsen

University of Mons, Belgium

Abstract

The helping Lemma 7 in [Maslowski and Wijsen, ICDT, 2014] is false. The lemma is used in (and only in) the proof of Theorem 3 of that same paper. In this corrigendum, we provide a new proof for the latter theorem.

1 The Flaw

The helping Lemma 7 in [MW14] is false. A counterexample is given next.

Example 1. For $\mathbf{S} = \{R, S\}$ and $q = \{R(\underline{x}, y),\ S(\underline{y})\}$, we have $\mathsf{enc}_{\mathbf{S}}(q) = \{N(\underline{R, x}, y),\ N(\underline{S, y}, 0)\}$. From [MW14, Lemma 8], it follows that $\sharp\mathsf{CERTAINTY}(\mathsf{enc}_{\mathbf{S}}(q))$ is $\sharp\mathbf{P}$-hard. From [MW13, Theorem 4], it follows that $\sharp\mathsf{CERTAINTY}(q)$ is in $\mathbf{FP}$. Consequently, assuming $\sharp\mathbf{P} \neq \mathbf{FP}$, there exists no polynomial-time many-one reduction from $\sharp\mathsf{CERTAINTY}(\mathsf{enc}_{\mathbf{S}}(q))$ to $\sharp\mathsf{CERTAINTY}(q)$. Lemma 7 in [MW14] is thus false. □

The first part in the proof of Lemma 7 in [MW14] is correct; it shows a polynomial-time many-one reduction from $\sharp\mathsf{CERTAINTY}(q)$ to $\sharp\mathsf{CERTAINTY}(\mathsf{enc}_{\mathbf{S}}(q))$. However, the second part in that proof is flawed when it claims *"We can compute in polynomial time the (unique) database $\mathbf{db}'_0$ with schema $\mathbf{S}$ such that $\mathsf{enc}_{\mathbf{S}}(\mathbf{db}'_0) = \mathbf{db}_0$."* The flaw is that the database $\mathbf{db}'_0$ does not generally exist, as shown next. Let $\mathbf{S} = \{R, S\}$ and $q = \{R(\underline{x}, y),\ S(\underline{y})\}$, as in Example 1. Then, $\mathsf{enc}_{\mathbf{S}}(q) = \{N(\underline{R, x}, y),\ N(\underline{S, y}, 0)\}$. A legal input to $\sharp\mathsf{CERTAINTY}(\mathsf{enc}_{\mathbf{S}}(q))$ is $\mathbf{db}_0 = \{N(\underline{R, b}, c),\ N(\underline{S, c}, 0),\ N(\underline{S, c}, 1)\}$. However, there exists no database $\mathbf{db}'_0$ such that $\mathsf{enc}_{\mathbf{S}}(\mathbf{db}'_0) = \mathbf{db}_0$. Indeed, for every database $\mathbf{db}'_0$ with schema $\mathbf{S}$, if $N(\underline{S, c}, s) \in \mathsf{enc}_{\mathbf{S}}(\mathbf{db}'_0)$, then $s = 0$.

2 The Solution

The following treatment is relative to a database schema $\mathbf{S}$. Let k, m be non-negative integers such that every relation name in $\mathbf{S}$ has at most k primary-key positions, and at most m non-primary-key positions. We define a new function $\mathsf{enc}^*_{\mathbf{S}}(q)$ which encodes Boolean conjunctive queries q into unirelational Boolean conjunctive queries. For $\mathsf{enc}^*_{\mathbf{S}}(q)$, we use a fresh relation name N with $k+1$ primary-key positions, and m non-primary-key positions. For every atom $R(\underline{\vec{x}}, \vec{y})$ in q, the query $\mathsf{enc}^*_{\mathbf{S}}(q)$ will contain some atom $N(\underline{R, \vec{x}, \vec{0}}, \vec{y}, \vec{z})$, where $\vec{0}$ is a sequence of padding zeros, and $\vec{z}$ is a sequence of padding fresh variables, all distinct and not occurring elsewhere. This encoding is different from [MW14, Definition 3] where a sequence of padding zeros was used instead of $\vec{z}$.

Example 2. We illustrate the difference between the old encoding $\mathsf{enc}_{\mathbf{S}}(\cdot)$ of [MW14, Definition 3] and the newly proposed encoding $\mathsf{enc}^{*}_{\mathbf{S}}(\cdot)$. For $q_0 = \{R(\underline{x}, y),\ S(\underline{y})\}$, we have

$$\begin{aligned}\mathsf{enc}_{\mathbf{S}}(q_0) &= \{N(\underline{R, x}, y), N(\underline{S, y}, 0)\},\\ \mathsf{enc}^{*}_{\mathbf{S}}(q_0) &= \{N(\underline{R, x}, y), N(\underline{S, y}, z)\}.\end{aligned}$$

We recall from [MW14, p. 156] that the *complex part* of a Boolean conjunctive query contains every atom $F \in q$ such that some non-primary-key position in F contains either a variable with two or more occurrences in q or a constant. Note that $N(\underline{S, y}, 0)$ belongs to the complex part of $\mathsf{enc}_{\mathbf{S}}(q_0)$, while $N(\underline{S, y}, z)$ is not in the complex part of $\mathsf{enc}^{*}_{\mathbf{S}}(q_0)$. □

Definition 1. We define skBCQ as the class of Boolean conjunctive queries in which all relation names are simple-key. We say that a query $q \in \mathsf{skBCQ}$ is *minimal* if both

- q contains no two distinct atoms $R_1(\underline{x_1}, \vec{y}_1)$, $R_2(\underline{x_2}, \vec{y}_2)$ such that $R_1 = R_2$ and $x_1 = x_2$; and
- there exists no substitution θ over $\mathsf{vars}(q)$ such that $\theta(q) \subsetneq q$.

We define cxBCQ as the class of *unirelational* Boolean conjunctive queries q whose relation name has signature $[n, 2]$ (for some $n \geq 2$) such that for every $F \in q$, the first position of F is a constant.

Definition 2. The *intersection graph* of a Boolean conjunctive query is an undirected graph whose vertices are the atoms of q. There is an undirected edge between any two atoms that have a variable in common.

Lemma 1. *Assume* $\sharp\mathbf{P} \neq \mathbf{FP}$. *For every minimal query* q *in* skBCQ*, if* $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^{*}_{\mathbf{S}}(q))$ *is* $\sharp\mathbf{P}$*-hard, then so is* $\sharp\mathsf{CERTAINTY}(q)$*.*

Proof. Let q be a minimal query in skBCQ such that $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^{*}_{\mathbf{S}}(q))$ is $\sharp\mathbf{P}$-hard. Note that q does not need to be unirelational or self-join-free. The query $\mathsf{enc}^{*}_{\mathbf{S}}(q)$, which is unirelational, is a legal input to the function IsEasy of [MW14, p. 163].† Since $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^{*}_{\mathbf{S}}(q))$ is $\sharp\mathbf{P}$-hard, the function IsEasy will return **false** on input $\mathsf{enc}^{*}_{\mathbf{S}}(q)$. This function will repeat, as long as possible, the following step: pick some atom $N(\underline{R, c}, \vec{y})$ and some variable $y \in \mathsf{vars}(\vec{y})$, with R some relation name (treated as a constant) and c some constant, and replace all occurrences of y with an arbitrary constant. Let $\bar{q}$ be the query that results from these steps. Clearly, for every atom $N(\underline{R, s}, \vec{t})$ in $\bar{q}$, either s is a constant or $\vec{t}$ is variable-free. Since IsEasy returns **false** on input $\bar{q}$, it follows that $\bar{q}$ does not satisfy the premise of [MW14, Lemma 5]. Therefore, it must be the case that $\bar{q}$ contains two distinct atoms $N(\underline{R, x}, \vec{u})$ and $N(\underline{S, y}, \vec{w})$ that are connected in the intersection graph of $\bar{q}$ such that

- R and S are relation names (serving as constants), not necessarily distinct;
- x and y are distinct variables; and
- neither $\vec{u}$ nor $\vec{w}$ is exclusively composed of variables occurring only once in the query. That is, $N(\underline{R, x}, \vec{u})$ and $N(\underline{S, y}, \vec{w})$ belong to the complex part of $\bar{q}$.

†For uniformity of notation, we will assume that the unirelational query uses relation name N.

For every relation name R that appears in q, we assume fresh relation names $R_1, R_2, R_3, \ldots$ with the same signature as R. Using these relation names, we can construct a self-join-free Boolean conjunctive query q' such that $|q'| = |q|$ and for every atom $R(\underline{x}, \vec{y})$ in q, the query q contains some atom $R_i(\underline{x}, \vec{y})$. For example, if $q = \{R(\underline{x}, y),\ R(\underline{y}, z),\ S(\underline{z}, x)\}$, then we can let $q' = \{R_1(\underline{x}, y),$ $R_2(\underline{y}, z),\ S_1(\underline{z}, x)\}$. It can now be shown that the function IsSafe in [MW14, p. 158] will return **false** on input q', and thus $\sharp\mathsf{CERTAINTY}(q')$ is $\sharp\mathbf{P}$-hard. Indeed, whenever IsEasy picked $N(\underline{R, c}, \vec{y})$ and some variable $y \in \mathsf{vars}(\vec{y}) \cap \mathsf{vars}(q)$, the function IsSafe can execute SE3 on the corresponding R_i-atom of q'. This eventually leads to a query whose complex part contains two atoms $R_i(\underline{x}, \vec{u}')$ and $S_j(\underline{y}, \vec{w}')$, $x \neq y$, that are connected in the intersection graph, at which point IsSafe will return **false**. In this reasoning, one needs that non-primary-key positions are padded with fresh variables occurring only once, as can be seen from Example 2.

In the remainder of this proof, we show the existence of a polynomial-time many-one reduction from $\sharp\mathsf{CERTAINTY}(q')$ to $\sharp\mathsf{CERTAINTY}(q)$. We incidentally note that the remaining reasoning, which generalizes the proof of [MW14, Lemma 2], does not require that relation names are simple-key.

Let f be a mapping from facts to facts such that for every atom $R_i(x_1, \ldots, x_n) \in q'$, for every R_i-fact $A := R_i(a_1, \ldots, a_n)$, $f(A) := R(\langle a_1, x_1\rangle, \ldots, \langle a_n, x_n\rangle)$. Notice that f maps R_i-facts to R-facts. Here, every couple $\langle a_i, x_i\rangle$ denotes a constant such that $\langle a_i, x_i\rangle = \langle a_j, x_j\rangle$ if and only if both $a_i = a_j$ and $x_i = x_j$. Moreover, if c is a constant, then $\langle c, c\rangle := c$. Since no two distinct atoms of q agree on both their relation name and primary key, it will be the case that for all facts A and B, $A \sim B$ if and only if $f(A) \sim f(B)$, where $\sim$ denotes "is key-equal-to."

We extend the function f in the natural way to databases $\mathbf{db}$ that use only relation names from q': $f(\mathbf{db}) := \{f(A) \mid A \in \mathbf{db}\}$. Clearly, $f(\mathbf{db})$ can be computed in polynomial time in the size of $\mathbf{db}$. Let $\mathbf{db}$ be a set of facts with relation names in q'. It can be easily seen that $|\mathsf{rset}(\mathbf{db})| = |\mathsf{rset}(f(\mathbf{db}))|$ and $\mathsf{rset}(f(\mathbf{db})) = \{f(\mathbf{r}) \mid \mathbf{r} \in \mathsf{rset}(\mathbf{db})\}$. Let $\mathbf{r}$ be an arbitrary repair of $\mathbf{db}$. It suffices to show that

$$\mathbf{r} \models q' \iff f(\mathbf{r}) \models q.$$

For the implication $\implies$, assume that $\mathbf{r} \models q'$. We can assume a valuation θ over $\mathsf{vars}(q')$ such that $\theta(q') \subseteq \mathbf{r}$. Let μ be the valuation such that for every variable $x \in \mathsf{vars}(q')$, $\mu(x) = \langle\theta(x), x\rangle$. By our construction of q' and f, it will be the case that $\mu(q) \subseteq f(\mathbf{r})$, thus $f(\mathbf{r}) \models q$.

For the implication $\impliedby$, assume that $f(\mathbf{r}) \models q$. We can assume a valuation θ over $\mathsf{vars}(q)$ such that $\theta(q) \subseteq f(\mathbf{r})$. Notice that if c is a constant in q, then it must be the case that $\theta(c) = \langle c, c\rangle := c$. We define θ_L as the substitution that maps every variable x in $\mathsf{vars}(q)$ to the first coordinate of $\theta(x)$; and θ_R maps every x to the second coordinate of $\theta(x)$. It is convenient to think of L and R as references to the Left and the Right coordinates, respectively. Thus, by definition, $\theta(x) = \langle\theta_L(x), \theta_R(x)\rangle$.

By inspecting the right-hand coordinates of couples $\langle a_i, x_i\rangle$ in $f(\mathbf{r})$, it can be easily seen that $\theta(q) \subseteq f(\mathbf{r})$ implies $\theta_R(q) \subseteq q$. Since the query q is minimal, it follows that $\theta_R(q) = q$, i.e., θ_R is an automorphism. Since the inverse of an automorphism is an automorphism, ${\theta_R}^{-1}$ is an automorphism as well. Note that θ_R will be the identity on constants that appear in q. We now define $\mu := \theta_L \circ {\theta_R}^{-1}$ (i.e., μ is the composed function θ_L after the inverse of θ_R), and show that $\mu(q') \subseteq \mathbf{r}$, which implies the desired result that $\mathbf{r} \models q'$. To this extent, let $R_i(x_1, \ldots, x_n)$ be an arbitrary atom of q'. It suffices to show $R_i(\mu(x_1), \ldots, \mu(x_n)) \in \mathbf{r}$, which can be proved as follows. From $R_i(x_1, \ldots, x_n) \in q'$, it follows $R(x_1, \ldots, x_n) \in q$. Thus, since ${\theta_R}^{-1}$ is an automorphism,

$$R\left(\ {\theta_R}^{-1}(x_1),\ \ldots,\ {\theta_R}^{-1}(x_n)\ \right) \in q.$$

Since $\theta(q) \subseteq f(\mathbf{r})$,

$$R\left(\ \theta\left(\theta_R{}^{-1}(x_1)\right),\ \ldots,\ \theta\left(\theta_R{}^{-1}(x_n)\right)\ \right) \in f(\mathbf{r}).$$

Since, for every symbol s, $\theta(s) = \langle \theta_L(s), \theta_R(s) \rangle$ and $\theta_R\left(\theta_R{}^{-1}(s)\right) = s$, we obtain

$$R\left(\ \langle \theta_L(\theta_R{}^{-1}(x_1)), x_1 \rangle,\ \ldots,\ \langle \theta_L(\theta_R{}^{-1}(x_n)), x_n \rangle\ \right) \in f(\mathbf{r}).$$

That is, by our definition of μ,

$$R\left(\ \langle \mu(x_1), x_1 \rangle,\ \ldots,\ \langle \mu(x_n), x_n \rangle\ \right) \in f(\mathbf{r}).$$

From this, it is correct to conclude that $R_i(\mu(x_1), \ldots, \mu(x_n)) \in \mathbf{r}$. This concludes the proof. $\square$

Lemma 2. *For every Boolean conjunctive query q, there exists a polynomial-time many-one reduction from $\sharp\mathsf{CERTAINTY}(q)$ to $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^*_{\mathbf{S}}(q))$.*

Proof. Let q be a Boolean conjunctive query. Let R be a relation name that occurs in q. Let $\{R(\underline{\vec{x}_i}, \vec{y}_i)\}_{i=1}^m$ be the set of R-atoms of q. Then, $\mathsf{enc}^*_{\mathbf{S}}(q)$ will contain, for every $i \in \{1, \ldots, m\}$, some atom $N(\underline{R, \vec{x}_i, \vec{0}}, \vec{y}_i, \vec{z}_i)$, where $\vec{z}_i$ is a (possibly empty) sequence of distinct fresh variables not occurring elsewhere. For every R-fact $A := R(\underline{\vec{a}}, \vec{b})$, we define $f(A) := N(\underline{R, \vec{a}, \vec{0}}, \vec{b}, \vec{0})$. Note here that $f(A)$ depends on the signatures of R and N, but not on the R-atoms of q. The mapping f is defined similarly for all relation names that appear in q. It can be easily seen that for all facts A and B whose relation names appear in q, $A \sim B$ if and only if $f(A) \sim f(B)$.

If $\mathbf{db}$ is an instance of $\sharp\mathsf{CERTAINTY}(q)$, we can assume without loss of generality that every relation name in $\mathbf{db}$ also appears in q. We extend the function f to such instances $\mathbf{db}$ of $\sharp\mathsf{CERTAINTY}(q)$: $f(\mathbf{db}) := \{f(A) \mid A \in \mathbf{db}\}$. Obviously, $f(\mathbf{db})$ can be computed in polynomial time in the size of $\mathbf{db}$. It is also obvious that $|\mathsf{rset}(\mathbf{db})| = |\mathsf{rset}(f(\mathbf{db})|$ and $\mathsf{rset}(f(\mathbf{db})) = \{f(\mathbf{r}) \mid \mathbf{r} \in \mathsf{rset}(\mathbf{db})\}$. It suffices to show that for every repair $\mathbf{r}$ of $\mathbf{db}$,

$$\mathbf{r} \models q \iff f(\mathbf{r}) \models \mathsf{enc}^*_{\mathbf{S}}(q).$$

For the implication $\Longrightarrow$, assume $\mathbf{r} \models q$. We can assume a valuation θ over $\mathsf{vars}(q)$ such that $\theta(q) \subseteq \mathbf{r}$. Let θ' be the valuation that extends θ from $\mathsf{vars}(q)$ to $\mathsf{vars}(\mathsf{enc}^*_{\mathbf{S}}(q))$ such that $\theta'(z) = 0$ for every variable z that appears in $\mathsf{enc}^*_{\mathbf{S}}(q)$ but not in q. By the construction of f, it will be the case that $\theta'(\mathsf{enc}^*_{\mathbf{S}}(q)) \subseteq f(\mathbf{r})$. Indeed, if $\mathsf{enc}^*_{\mathbf{S}}(q)$ contains $N(\underline{R, \vec{x}_i, \vec{0}}, \vec{y}_i, \vec{z}_i)$, then $\mathbf{r}$ will contain $R(\underline{\theta(\vec{x}_i)}, \theta(\vec{y}_i))$, hence $f(\mathbf{r})$ will contain $N(\underline{R, \theta'(\vec{x}_i), \vec{0}}, \theta'(\vec{y}_i), \vec{0})$ and $\theta'(\vec{z}_i) = \vec{0}$.

For the implication $\Longleftarrow$, assume $f(\mathbf{r}) \models \mathsf{enc}^*_{\mathbf{S}}(q)$. We can assume a valuation θ over $\mathsf{vars}(\mathsf{enc}^*_{\mathbf{S}}(q))$ such that $\theta(\mathsf{enc}^*_{\mathbf{S}}(q)) \subseteq f(\mathbf{r})$. It is straightforward to see that $\theta(q) \subseteq \mathbf{r}$. $\square$

We now give the new proof for Theorem 3 in [MW14].

Theorem 1 ([MW14, Theorem 3])**.** *The set $\{\sharp\mathsf{CERTAINTY}(q) \mid q \in \mathsf{skBCQ}\}$ exhibits an effective* **FP**-$\sharp$**P**-*dichotomy.*

New proof. Let $q \in \mathsf{skBCQ}$. It can be decided whether q can be satisfied by a consistent database. If q cannot be satisfied by a consistent database, then for every database $\mathbf{db}$, the number of repairs of $\mathbf{db}$ satisfying q is 0. An example is $q = \{R(\underline{x}, 0),\ R(\underline{x}, 1)\}$. Assume next that q can be satisfied by a consistent database. Then, we can compute a minimal query q_m such that for every database, the number of repairs satisfying q_m is equal to the number of repairs satisfying q. That is, the problems $\sharp\mathsf{CERTAINTY}(q_m)$ and $\sharp\mathsf{CERTAINTY}(q)$ are identical.

Then, $\mathsf{enc}^*_{\mathbf{S}}(q_m)$ belongs to cxBCQ. By [MW14, Lemma8], the set $\{\sharp\mathsf{CERTAINTY}(q) \mid q \in \mathsf{cxBCQ}\}$ exhibits an effective **FP**-$\sharp$**P**-hard dichotomy. If the problem $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^*_{\mathbf{S}}(q_m))$ is in **FP**, then $\sharp\mathsf{CERTAINTY}(q)$ is in **FP** by Lemma 2; and if $\sharp\mathsf{CERTAINTY}(\mathsf{enc}^*_{\mathbf{S}}(q_m))$ is $\sharp$**P**-hard, then $\sharp\mathsf{CERTAINTY}(q)$ is $\sharp$**P**-hard by Lemma 1. Consequently, $\sharp\mathsf{CERTAINTY}(q)$ is in **FP** or $\sharp$**P**-hard, and it is is decidable which of the two cases applies. □

References

[MW13] Dany Maslowski and Jef Wijsen. A dichotomy in the complexity of counting database repairs. *J. Comput. Syst. Sci.*, 79(6):958–983, 2013.

[MW14] Dany Maslowski and Jef Wijsen. Counting database repairs that satisfy conjunctive queries with self-joins. In Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy, editors, *Proc. 17th International Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014.*, pages 155–164. OpenProceedings.org, 2014.