A note on computing certain answers to queries over incomplete databases

Marcelo Arenas[1], Elena Botoeva[2], Egor V. Kostylev[3], and Vladislav Ryzhikov[2]

[1]Pontificia Universidad Católica de Chile, Chile
[2]Free University of Bozen-Bolzano, Italy
[3]University of Oxford, UK

1 Introduction

The ability to handle incomplete information is fundamental in many areas, including data integration, data exchange, inconsistent databases, and the Semantic Web. In such areas, it is often impossible to describe the domain of interest in a full, comprehensive way, so the aim of an incomplete database is to concisely represent a (potentially infinite) number of completions. Incomplete information is usually represented by allowing placeholders for unknown values, which are called "nulls". An incomplete database I with such nulls then represents a set of databases with complete information, called representations of I, each of which is obtained by replacing nulls by actual values.

One of the most important problems associated to databases is query answering. Since an incomplete database I can have several representations, the aim in this context is to find the "certain answers" to a query Q, that is, an incomplete database I' that precisely represents the answers to Q over all the representations of I. Unfortunately, this is possible only in restricted settings, which allow for the so called strong representation systems [5]. To overcome this limitation, it was recently proposed in [7] the idea of computing "certain answers as objects". It is argued in [7] that we should aim for finding an object (i.e., an incomplete database) representing the answers over the representations of an incomplete database in the most informative way. More precisely, informativeness in this context is formalised as the following preorder on incomplete databases: $I_1 \preceq I_2$ if and only if each representation of I_2 is a representation of I_1, that is, the more representations an incomplete database has, the less informative it is. Then given a query Q over an incomplete database I, the certain answers as objects to Q over I are defined as a greatest lower bounds under $\preceq$ of the set of answers to Q over all the representations of I.

In this paper, we make initial steps in the study of the complexity of the problem of computing certain answers as objects. In particular, we concentrate on the widely studied setting of union of conjunctive queries with inequalities and incomplete databases under the open-world assumption, and we provide positive and negative results. On the positive side, we show that in this case the certain answers to a query can be computed. On the negative side, we show that this computation can be costly, as the certain answers can be of exponential size even if we restrict to conjunctive queries with inequalities over binary relations.

2 Preliminaries

We assume countably infinite disjoint sets of constants, denoted by Const, and of nulls, denoted by Null. A *relational schema* (or just *schema*) Σ is a set of relation names with associated arities. An *incomplete database* I over Σ is an assignment of a k-ary relation R^I over $(\mathsf{Const} \cup \mathsf{Null})$ to each k-ary name R in Σ. Sets of constants and nulls that occur in I are denoted by $\mathsf{Const}(I)$ and $\mathsf{Null}(I)$, respectively, and their union, called an *active domain*, is denoted by $\mathsf{adom}(I)$. A *complete* database is an incomplete database without nulls. In what follows, we use I, I_1, I_2, I', ... to denote incomplete databases, and D, D_1, D_2, D', ... to denote complete databases.

The semantics of an incomplete database I is defined in terms of its representations, which are complete databases that are considered as possible interpretations of I [1, 5]. To define this semantics, we need the following notion: a *valuation* on an incomplete database I is a map $v : \mathsf{adom}(I) \to \mathsf{Const}$ that is the identity on $\mathsf{Const}(I)$. Such a mapping naturally extends to databases, so we write $v(I)$ as well. Then, under the *open-world assumption*, the set of *representations* of I, denoted by $[\![I]\!]$, is defined as $[\![I]\!] = \{D \mid D \text{ is a complete database and } v(I) \subseteq D \text{ for some valuation } v \text{ of } I\}$. Note that $[\![I]\!] \neq \emptyset$ for every incomplete database I. An incomplete database I_2 is *at least as informative as* an incomplete database I_1 [7], denoted by $I_1 \preceq I_2$, if $[\![I_2]\!] \subseteq [\![I_1]\!]$. Notice that $\preceq$ is a preorder, that is, it is reflexive and transitive. Moreover, given an incomplete database I and a set $\mathcal{I}$ of incomplete databases, I is a *lower bound* for $\mathcal{I}$ if $I \preceq I'$ for every $I' \in \mathcal{I}$, and I is a *greatest lower bound* for $\mathcal{I}$ if I is a lower bound for $\mathcal{I}$ and for every lower bound I' for $\mathcal{I}$, it holds that $I' \preceq I$. The set of all greatest lower bounds of $\mathcal{I}$ is denoted by $\mathsf{glb}(\mathcal{I})$.

A *homomorphism* from an incomplete database I_1 to an incomplete database I_2 is a function $h : \mathsf{adom}(I_1) \to \mathsf{adom}(I_2)$ that is the identity on $\mathsf{Const}(I_1)$ and such that $h(I_1) \subseteq I_2$ (we define $h(I)$ analogously to $v(I)$ before). We write $I_1 \mapsto I_2$ if there is a homomorphism from I_1 to I_2, and $I_1 \mapsto \mathcal{I}$, for a set $\mathcal{I}$ of incomplete databases, if $I_1 \mapsto I_2$ for each $I_2 \in \mathcal{I}$. Then a characterization of $\preceq$ is given by the following statement [7]:

$$I_1 \preceq I_2 \text{ if and only if } I_1 \mapsto I_2. \tag{1}$$

As customary, a *query* Q over a schema Σ is a function that assigns to each complete database D of Σ a complete database $Q(D)$. Moreover, given an incomplete database I, the *certain answers as object* (or *certain answers*, for short) to Q over I, denoted by $\mathsf{cert}(Q, I)$, is defined [7] as an incomplete database satisfying

$$\mathsf{cert}(Q, I) \in \mathsf{glb}(\mathsf{ans}(Q, I)), \quad \text{where} \quad \mathsf{ans}(Q, I) = \{Q(D) \mid D \in [\![I]\!]\}.$$

Notice that two greatest lower bounds I_1, I_2 of $\mathsf{ans}(Q, I)$ are equivalent in terms of the information ordering: $I_1 \preceq I_2$ and $I_2 \preceq I_1$; thus, we can choose any greatest lower bound of $\mathsf{ans}(Q, I)$ as the certain answer to Q over I.

3 Computing Certain Answers

In this section, we focus on the problem of computing $\mathsf{cert}(Q, I)$ for a union of conjunctive queries Q with inequalities ($\text{UCQ}^{\neq}$) and an incomplete database I. It is assumed

that a reader is familiar with the syntax of such queries as well as their semantics (i.e., the definition of $Q(D)$). Complete proofs of the results below can be found in [3].

The first issue we have to deal with when computing $\mathsf{cert}(Q, I)$ is the fact that $\mathsf{ans}(Q, I)$ may be infinite. The following proposition overcomes this limitation by showing that it is not necessary to consider the entire set $\mathsf{ans}(Q, I)$ when computing $\mathsf{cert}(Q, I)$, but instead one can consider just a finite subset of it.

Proposition 1. *Let Q be a UCQ$^{\neq}$ over a schema Σ and I an incomplete database over Σ. Then there exists $\mathcal{D} \subseteq \mathsf{ans}(Q, I)$ such that $\mathcal{D}$ is finite and $\mathsf{cert}(Q, I)$ is a greatest lower bound of $\mathcal{D}$.*

In fact, from the proof of Proposition 1, it is possible to obtain a simple exponential-time algorithm that, given a UCQ$^{\neq}$ Q and an incomplete database I, returns a set $\mathcal{D} \subseteq \mathsf{ans}(Q, I)$ such that $\mathsf{cert}(Q, I) \in \mathsf{glb}(\mathcal{D})$, where each $D \in \mathcal{D}$ is of linear size in the size of I. We denote such $\mathcal{D}$ by $\mathsf{ans}_{\mathsf{can}}(Q, I)$.

Now, given a finite set $\mathcal{D}$, it is known (see, e.g., [8], [6, Proposition 5], or [4]) that a greatest lower bound of $\mathcal{D}$ with respect to the relation $\mapsto$ and, by (1), to the relation $\preceq$ can be computed via the (*direct*) *product* of the databases in $\mathcal{D}$, which is defined as follows. Let I_1 and I_2 be incomplete databases over a schema Σ. The product $I_1 \times I_2$ is a database I such that, for each k-ry relation R in Σ,

$$R^I = \{\big(a_1 \times b_1, \ldots, a_k \times b_k\big) \mid (a_1, \ldots, a_k) \in R^{I_1},\ (b_1, \ldots, b_k) \in R^{I_2}\};$$

here $a \times a = a$ for all $a \in \mathsf{Const} \cup \mathsf{Null}$ and $a \times b$ is a fresh null $n_{a,b}$ for all different $a, b \in \mathsf{Const} \cup \mathsf{Null}$. It immediately follows that $\mathsf{Const}(I) \subseteq \mathsf{Const}(I_1) \cap \mathsf{Const}(I_2)$. Note that the product is associative and commutative (up to renaming of nulls). For $\mathcal{D} = \{D_1, \ldots, D_n\}$, we denote by $\prod \mathcal{D}$ the product $D_1 \times \cdots \times D_n$. The following picture illustrates this notion for a schema consisting of a single binary relation:

I_1 $\quad$ I_2 $\quad$ $I_1 \times I_2$

By combining Proposition 1 with the algorithm for computing $\mathsf{ans}_{\mathsf{can}}(Q, I)$, we obtain an algorithm for computing a certain answer.

Proposition 2. *Let Q be a UCQ$^{\neq}$ over a schema Σ and I and incomplete database over Σ. Then $\mathsf{cert}(Q, I)$ can be computed as $\prod \mathsf{ans}_{\mathsf{can}}(Q, I)$.*

Finally, in the following theorem, which is the main result of this note, we prove that the certain answers as objects can be of exponential size, even if we restrict to the case of conjunctive queries over binary relations.

Theorem 1. *There exists a family of incomplete databases I_n, for $n \in \mathbb{N}$, and a conjunctive query Q with inequality such that the size of the smallest $\mathsf{cert}(Q, I_n)$ grows exponentially in the size of I_n.*

Proof. (Sketch) For a natural number m, consider the complete database

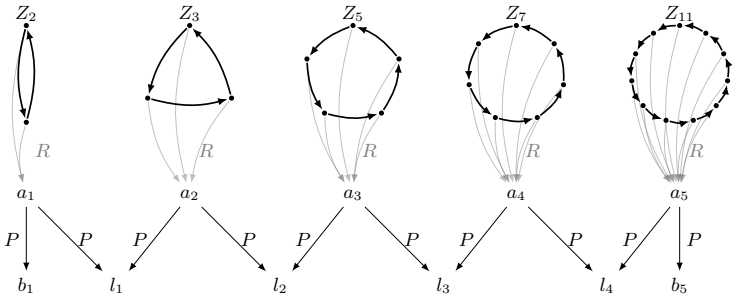
$$Z_m = \{Z(d_1, d_2), \ldots, Z(d_{m-1}, d_m), Z(d_m, d_1)\},$$

where $d_1, d_2, \ldots, d_m$ are (distinct) constants, encoding a directed cycle of length m.

Let n be a natural number, and let $p_1, \ldots, p_n$ be the first n prime numbers. We define I_n to be the incomplete database containing

- disjoint cycles $Z_{p_1}, \ldots, Z_{p_n}$,
- for a constant a_i, $i = 1, .., n$, the pairs $R(c_i, a_i)$ for each constant c_i in Z_{p_i},
- for constants b_1 and b_n, the pairs $P(a_1, b_1)$ and $P(a_n, b_n)$, and
- for nulls $l_1, \ldots, l_{n-1}$, the pairs $P(a_i, l_i)$ and $P(a_{i+1}, l_i)$ for $1 \leq i \leq n-1$.

Below we depict I_5:



Consider the query

$$Q(x, y) = \exists z(Z(x, y) \wedge R(x, z) \wedge R(y, z) \wedge Q^{\neq}(z)),$$
$$\text{where} \quad Q^{\neq}(z) = \exists u \exists v(P(z, u) \wedge P(z, v) \wedge (u \neq v)).$$

We can show the following property:

(minimal) each binary relation $C_{p_i} = \{(c, d) \mid Z(c, d) \in Z_{p_i}\}$, for $1 \leq i \leq n$, is in $\mathsf{ans}(Q, I_n)$, and any other relation in $\mathsf{ans}(Q, I_n)$ subsumes one of them.

Let $\mathcal{Z}_n = \{C_{p_1}, \ldots, C_{p_n}\}$. We define a directed cycle of nulls of size $p_1 \times \cdots \times p_n$

$$Z^* = \{Z(n_1, n_2), \ldots, Z(n_{p_1 \times \cdots \times p_n - 1}, n_{p_1 \times \cdots \times p_n}), Z(n_{p_1 \times \cdots \times p_n}, n_1)\}.$$

We can prove two claims.

Claim. Any of certain answers $\mathsf{cert}(Q, I_n)$ is in $\mathsf{glb}(\mathcal{Z}_n)$.

Claim. Set Z^* is in $\mathsf{glb}(\mathcal{Z}_n)$.

The proof of the claims relies on **(minimal)**, while the construction of Z^* and the proof of the second claim rely on the fact that C_{p_i} are cycles of prime length. Indeed, Z^* is (isomorphic to) the product of all C_{p_i}, as the product of a pair of directed cycles of sizes k_1 and k_1, when k_1 and k_2 are co-prime numbers, is a directed cycle of the size $k_1 \times k_2$.

From the above claims, it follows that Z^* and $\mathsf{cert}(Q, I_n)$ are homomorphically equivalent. It is well-known from graph theory that every graph that is homomorphically equivalent to a directed cycle has to contain that cycle. Hence, the minimal certain answer $\mathsf{cert}(Q, I_n)$ is of exponential size in the size of I_n. (We observe that the size of I_n is polynomially bounded in n, as $p_n < cn^2$, for a constant c [2].) □

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. T. M. Apostol. *Introduction to Analytic Number Theory*. Springer-Verlag, New York, 1976. Undergraduate Texts in Mathematics.
3. M. Arenas, E. Botoeva, E. V. Kostylev, and V. Ryzhikov. A note on computing certain answers to queries over incomplete databases (extended version). Technical report, 2017. Available at `http://marenas.sitios.ing.uc.cl/publications/amw17-ext.pdf`.
4. C. Chang and H. Keisler. *Model Theory*. North-Holland, Amsterdam, 1990.
5. T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
6. L. Libkin. Incomplete information and certain answers in general data models. In *Proc. of the 30th ACM Symp. on Principles of Database Systems (PODS)*, pages 59–70, 2011.
7. L. Libkin. Certain answers as objects and knowledge. *Artif. Intell.*, 232:1–19, 2016.
8. B. ten Cate and V. Dalmau. The product homomorphism problem and applications. In *Proc. of the 18th International Conference on Database Theory (ICDT 2015)*, pages 161–176, 2015.