Journal of Computer and System Sciences 79 (2013) 958–983

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

A dichotomy in the complexity of counting database repairs ☆

Dany Maslowski, Jef Wijsen *

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

ARTICLE INFO

Article history:
Received 23 September 2011
Received in revised form 27 March 2012
Accepted 16 January 2013
Available online 21 January 2013

Keywords:
Conjunctive queries
Consistent query answering
Primary key
Probabilistic databases

ABSTRACT

An uncertain database **db** is defined as a database in which distinct tuples of the same relation can agree on their primary key. A repair is obtained by selecting a maximal number of tuples without ever selecting two distinct tuples of the same relation that agree on their primary key. Obviously, the number of possible repairs can be exponential in the size of the database. Given a Boolean query q, certain (or consistent) query answering concerns the problem to decide whether q evaluates to true on every repair. In this article, we study a counting variant of consistent query answering. For a fixed Boolean query q, we define $\sharp\mathsf{CERTAINTY}(q)$ as the following counting problem: Given an uncertain database **db**, how many repairs of **db** satisfy q? Our main result is that conjunctive queries q without self-join exhibit a complexity dichotomy: $\sharp\mathsf{CERTAINTY}(q)$ is in **FP** or $\sharp$**P**-complete.

1. Introduction

Uncertainty arises in many database applications. It can be modeled in a clean and elegant way by relations that violate their primary key constraint. We use the term *uncertain database* for databases that allow primary key violations. Such uncertainty is not necessarily a bad thing. In planning databases, for example, primary key violations can represent different alternatives. In the conference planning database of Fig. 1, where primary keys are underlined, the exact town of VLDB 2016 is still uncertain (it can be Mons, Gent, or Paris). Uncertainty also arises as an inconvenient but inescapable consequence of data integration and data exchange. The relation **S** in Fig. 1 combines data from two different sources, each providing a different country for the city of Tunis.

Uncertainty by primary key violations gives rise to (exponentially many) "possible worlds", which we will call *repairs*: every repair is obtained by selecting a maximal number of tuples from each relation without ever selecting two distinct tuples that agree on their primary key.

In previous works [2–5], we have studied the decision problem $\mathsf{CERTAINTY}(q)$ for a fixed Boolean conjunctive query q. This problem takes as input an uncertain database and asks whether q evaluates to true on every repair. For example, our example database has six repairs (there are three choices for the city of VLDB 2016, and two choices for the country of Tunis), each satisfying the Boolean conjunctive query:

$$\exists x \exists y \exists z \big(R(\text{'VLDB'}, \underline{x}, y) \wedge S(\underline{y}, z) \wedge T(\underline{z}, \text{'Europe'})\big),$$

stating that VLDB will be organized in Europe in some year. On the other hand, not all repairs satisfy the query:

$$q_1 = \exists x \exists y \exists z \big(R(\text{'VLDB'}, \underline{x}, y) \wedge S(\underline{y}, \text{'Belgium'})\big),$$

☆ An extended abstract of this article was published in the workshop proceedings of LID 2011 (Maslowski and Wijsen, 2011 [1]).

* Corresponding author.
E-mail addresses: dany.maslowski@umons.ac.be (D. Maslowski), jef.wijsen@umons.ac.be (J. Wijsen).

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2013.01.011

R

Conf	Year	Town
VLDB	2016	Mons
VLDB	2016	Gent
VLDB	2016	Paris

S

Town	Country
Mons	Belgium
Gent	Belgium
Paris	France
Tunis	France
Tunis	Tunisia

T

Country	Cont
Belgium	Europe
France	Europe
Tunisia	Africa

Fig. 1. An uncertain database.

SP

Town	Country	P
Tunis	France	0.16
Tunis	Tunisia	0.60

Fig. 2. A probabilistic relation.

stating that VLDB will be organized in Belgium in some year. A natural follow-up question here is: "How many repairs satisfy q_1?". We may be interested to know, for example, whether a query is true in more than half of the repairs. The query q_1 is satisfied by four repairs (out of six) of our example database.

This leads to a counting variant of CERTAINTY(q), denoted by ♯CERTAINTY(q), which is the following problem: Given an uncertain database **db**, determine the number of repairs that satisfy q [1]. Note that throughout this article, except for Property 1, all complexity results concern data complexity, that is, database schemas and queries are fixed, and the complexity is in terms of varying database size. The main result of this article is the following dichotomy theorem (self-join-free means that no relation name occurs more than once in q):

Theorem 1. *For every Boolean self-join-free conjunctive query q, at least one of the following holds*:

- ♯CERTAINTY(q) *is in* **FP**; *or*
- ♯CERTAINTY(q) *is* ♯**P***-complete under polynomial-time Turing reductions.*

Moreover, the dichotomy of Theorem 1 is effective, in the sense that we give an algorithm that takes as input a Boolean self-join-free conjunctive query q and determines whether ♯CERTAINTY(q) is in **FP** or ♯**P**-complete. The restriction to self-join-free queries has also been used in related works, for example, in [6,7,5]. Later on in Example 1, we illustrate why this restriction is crucial in our theoretical development.

The complexity class **FP** contains all counting, or more generally function problems which can be solved in deterministic polynomial time. The class ♯**P** contains all function problems which consist of counting the number of accepting computation paths of a non-deterministic polynomial-time Turing machine. In other words, ♯**P** captures the counting problems corresponding to decision problems in **NP**. By a counting analogue of Ladner's theorem [8], if **FP** $\neq$ ♯**P**, then there are counting problems in ♯**P** that are neither in **FP** nor ♯**P**-complete. By Toda's theorem [9], every problem in the polynomial-time hierarchy can be solved in polynomial time with an oracle that solves a ♯**P**-complete problem. Thus, ♯**P**-completeness suggests a higher level of intractability than **NP**-completeness, insofar decision problems and counting problems can be compared.

Dalvi et al. recently obtained a dichotomy like that of Theorem 1 for probabilistic databases [10,7]. Our proof techniques are inspired by that work and we use, to the extent it is possible, terminology and notation that is reminiscent of this earlier work. Nevertheless, the probabilistic database model differs from our uncertain database model in several fundamental respects, so that complexity results in either model do not generally carry over to the other. Probabilistic databases, unlike our uncertain databases, attach probabilities to tuples, as illustrated in Fig. 2.

The probabilistic relation in Fig. 2 models three possible worlds, each with its own probability. The three possible worlds and their probabilities are:

1. the singleton {**SP**('Tunis', 'France')} whose probability is 0.16;
2. the singleton {**SP**('Tunis', 'Tunisia')} with probability 0.60; and
3. the empty relation with probability $0.24 = 1 - 0.16 - 0.60$.

Our uncertain database model has no explicit probabilities. It contains some implicit probabilities, to the extent that if a relation contains a block of n tuples that agree on the primary key, then every tuple of that block has probability $1/n$ to be chosen in a repair. As a consequence, deleting or inserting tuples in a block will affect the probability of the other tuples. For example, if we delete the tuple **S**('Tunis', 'France') from the relation **S** in Fig. 1, then the other tuple **S**('Tunis', 'Tunisia') becomes automatically certain. In the probabilistic data model, on the other hand, deleting **SP**('Tunis', 'France' | 0.16) will not change the probability of the remaining possible world {**SP**('Tunis', 'Tunisia')} (it stays at 0.60), while the probability of the empty possible world will increase by 0.16. To conclude, uncertain and probabilistic databases both capture uncertainty, but they differ in some fundamental respects.

The remainder of this article is organized as follows. Section 2 formally defines our data model and the problem of interest. We focus on Boolean conjunctive queries in which each relation name is used at most once. Section 3 discusses