Recognizing Determinism in Prioritized Repairing of Inconsistent Databases

Benny Kimelfeld, Ester Livshits, and Liat Peterfreund

Technion, Haifa 32000, Israel
{bennyk,esterliv,liatpf}@cs.technion.ac.il

Abstract. A repair of an inconsistent database is traditionally defined as a consistent database that differs from the inconsistent one in a "minimal way." As there are often reasons to prefer one repair over another, researchers have introduced and investigated the framework of preferred repairs, where a priority relation between facts is lifted towards a priority relation between consistent databases, and repairs are restricted to ones that are optimal in the lifted sense. In this paper we describe our recent results on the complexity of deciding whether the priority relation suffices to clean the database unambiguously, or in other words, whether there is exactly one optimal repair. In particular, we show that different conventional semantics of priority lifting entail highly different complexities.

1 Introduction

Database inconsistency arises for various reasons and in different applications. In common applications of Big Data, information is obtained from imprecise sources (e.g., social encyclopedias or social networks) via imprecise procedures (e.g., natural-language processing). It may also arise when integrating conflicting data from different sources (each of which may be consistent). Arenas, Bertossi and Chomicki [3] introduced a principled approach to managing inconsistency, via the notions of *repairs* and *consistent query answering*. Informally, a *repair* of an inconsistent database I is a consistent database J that differs from I in a "minimal" way.

There are situations where it is natural to prefer one repair over another [5, 11, 16, 17]. For example, this is the case if one source is regarded more reliable than another (e.g., enterprise data vs. Internet harvesting, and precise vs. imprecise sensing equipment) or if available timestamp information implies that a more recent fact is more up to date than an earlier one.

Motivated by the above considerations, Staworko, Chomicki and Marcinkowski [16, 17] introduced the framework of *preferred repairs*. The main characteristic of this framework is that it uses a *priority* relation between conflicting facts of an inconsistent database, in order to define a notion of *preferred* repairs. Fagin et al. [9] have built on that concept (in conjunction with the framework of *document spanners* [10]) to devise a language for declaring *inconsistency cleaning* in text information-extraction systems. They have shown there that preferred repairs capture ad-hoc cleaning operations and strategies of some prominent systems for text analytics [2, 6].

While the classic complexity problems studied in the theory of repairs include repair checking [1] and consistent query answering [3, 15], the presence of preferences gives rise to the *determinism problem*, which Staworko et al. [17] refer to as *categoricity*: decide whether the provided priority relation suffices to clean the database unambiguously. In this paper we describe our recent results on analyzing the complexity of categoricity in the framework of preferred repairs.

2 Formal Setup

We adopt the framework of preferred repairs devised by Staworko et al. [17], which has also been used by Fagin et al. [8].

Prioritizing Inconsistent Databases. An *inconsistent database* over a schema $\mathbf{S}$ is a database D that violates the *constraints* of $\mathbf{S}$. We consider here two types of constraints. One is the well known (and restricted) Functional Dependencies (FDs). An FD has the form $R : X \to Y$, and it states that every two facts of the relation R either agree on the attribute set Y or disagree on the attribute set X. The more general type of constraints considered here is the one given by means of a *conflict hypergraph* [7]. A conflict hypergraph for a database instance I is a hypergraph $\mathcal{H}$ that has the facts of I as its node set. A subinstance J of I is *consistent* with respect to (w.r.t.) $\mathcal{H}$ if J is an independent set of $\mathcal{H}$ (that is, J contains no hyperedge of $\mathcal{H}$).

Recall that conflict hypergraphs can represent inconsistencies for various types of integrity constraints, including FDs, the more general *conditional FDs* [4], and the more general *denial constraints* [14]. In fact, every constraint that is anti-monotonic (i.e., where subsets of consistent sets are always consistent) can be represented as a conflict hypergraph. In the case of denial constraints, the translation from the logical constraints to the conflict hypergraph can be done in polynomial time under data complexity (i.e., when the schema signature and constraints are assumed to be fixed).

A *priority relation* $\succ$ over a database instance I is an acyclic binary relation over the facts of I. The semantics of $f \succ g$ is that "f is preferred to g" (e.g., because f comes from a more trusted source, or because f is more recent). A *prioritizing inconsistent database* is a triple $(I, \mathcal{H}, \succ)$ where I is a database instance, $\mathcal{H}$ is a conflict hypergraph for I, and $\succ$ is a priority relation over I with the property that $\succ$ compares only between facts that are neighbors in $\mathcal{H}$.[1] We say that $\succ$ is *total* if for every two facts f and g in I, if f and g are neighbors then either $f \succ g$ or $g \succ f$. A priority $\succ_c$ over I is a *completion* of $\succ$ (w.r.t. $\mathcal{H}$) if $\succ$ is a subset of $\succ_c$ and $\succ_c$ is total.

Preferred Repairs. Let D be an inconsistent prioritizing database. Staworko et al. [17] define three different notions of *preferred* repairs: *Pareto optimal*, *globally optimal*, and *completion optimal*. Informally speaking, the first two are based on the question of whether a repair can be *improved* by replacing a set of facts with a preferable set of facts. They differ in the way they define when one set of facts is considered preferable to another. The third is based on the notion of completion.

[1] This requirement has been made with the introduction of the framework [17]. Obviously, the lower bounds we present hold even without this requirement. Moreover, our main upper bound, Theorem 3, holds as well without this requirement.

For the formal definitions, let $(I, \mathcal{H}, \succ)$ be an inconsistent prioritizing database, and J and J' two distinct consistent subinstances of I. We say that J is a *Pareto improvement* of J' if there exists a fact $f \in J \setminus J'$ such that $f \succ f'$ for all facts $f' \in J' \setminus J$. We say that J is a *global improvement* of J' if for every fact $f' \in J' \setminus J$ there exists a fact $f \in J \setminus J'$ such that $f \succ f'$. In words, J is a Pareto improvement of J' if in order to obtain J from J' we insert and delete facts, and one of the inserted facts is preferred to all deleted facts. And J is a global improvement of J' if in order to obtain J from J' we insert and delete facts, and every deleted fact is compensated by some preferable inserted fact.

We then get the following variants of *preferred repairs* for an inconsistent prioritizing database $D = (I, \mathcal{H}, \succ)$. A consistent subinstance J of I is a: *(a) Pareto-optimal repair of* D if there is no Pareto improvement of J, *(b) globally-optimal repair of* D if there is no global improvement of J, *and (c) completion-optimal repair of* D if there exists a completion $\succ_c$ of $\succ$ such that J is a globally-optimal repair of $(I, \mathcal{H}, \succ_c)$.

We remark that in the definition of a completion-optimal repair, we could replace "globally-optimal" with "Pareto-optimal" and obtain an equivalent definition [17]. It can be shown quite easily that under each of the three notions, an optimal repair is also a repair in the sense of Arenas et al. [3].

Categoricity. For each of the three semantics, there is always at least one optimal repair [17]. The problem of *categoricity* is that of deciding whether there is *precisely* one optimal repair, and therefore the priority relation contains enough information to clean the inconsistent database unambiguously. Formally, *Pareto categoricity*, *global categoricity* and *completion categoricity* are the problems of deciding, given an inconsistent prioritizing database D, whether D has a single Pareto-optimal repair, a single globally-optimal repair, and a single completion-optimal repair, respectively. The problem of repair uniqueness (in different repair semantics) is also referred to as *determinism* [13] and *fix certainty* [12].

3 Complexity Results

We now describe our complexity results for categoricity under the three semantics of optimal repairs. We begin with the Pareto case. The following theorem gives a dichotomy in data complexity, covering all possible schemas with functional dependencies. As is often the case in dichotomy results, the main challenge in the proof is that of covering all the hard cases.

Theorem 1. *Let* $\mathbf{S}$ *be a relational schema with a set* Δ *of FDs. Pareto categoricity over* $\mathbf{S}$ *can be solved in polynomial time if the restriction of* Δ *to every relation of* $\mathbf{S}$ *is equivalent to a single FD. In every other case, the problem is coNP-complete.*

Theorem 1 shows that in the case of FDs, the class of schemas that have a tractable Pareto categoricity is very limited. Next, we discuss global categoricity, and again focus on FDs. Here we do not have a dichotomy, and establishing one is left as an open problem for future research.

Theorem 2. *Let* $\mathbf{S}$ *be a relational schema with a set* Δ *of FDs. The following hold.*

1. *If the restriction of Δ to every relation of* $\mathbf{S}$ *is equivalent to a single FD, then global categoricity over* $\mathbf{S}$ *can be solved in polynomial time.*
2. *Suppose that* Δ *consists of two nontrivial FDs* $X \to Y$ *and* $W \to Z$ *on the same relation. Suppose also that each of* W *and* Z *contains an attribute that is in none of the other three attribute sets. Then global categoricity over* $\mathbf{S}$ *is* Π_2^{p}*-complete.*

Theorem 2 states that the tractable case of Pareto categoricity remains tractable in the global case. On other schemas, the complexity can rise to an even higher complexity class. Next, we consider the completion case, where the picture is far more positive.

Theorem 3. *Completion categoricity is decidable in polynomial time, even in the general case where conflicts are given by a conflict hypergraph.*

Our polynomial time algorithm for solving completion categoricity is extremely simple, but its proof of correctness is quite intricate.

Lastly, we observe that in our proof of Π_2^{p}-hardness for global categoricity, our reduction constructs a nontransitive priority relation, and we ask whether transitivity makes a difference. Let $(I, \mathcal{H}, \succ)$ be an inconsistent prioritizing database. We say that $\succ$ is *transitive* if for every two facts f and g in I, if f and g are neighbors in $\mathcal{H}$ and $\succ$ contains a path from f to g, then $f \succ g$. Transitivity is a natural assumption when $\succ$ is interpreted as a partial order such as "is of better quality than" or "is more current than." We can show that the three semantics remain different in the presence of transitivity. Yet, quite remarkably, the completion and global semantics coincide as far as categoricity is concerned.

Theorem 4. *Consider an inconsistent prioritizing database, where conflicts are given by a conflict hypergraph and the priority relation is transitive. Then, there is a single completion-optimal repair if and only if there is a single globally optimal repair.*

Combining Theorems 3 and 4, we get the following corollary that shows the major impact of transitivity on global categoricity, when contrasted with Theorem 2.

Corollary 1. *For transitive priority relations, global categoricity is decidable in polynomial time, even in the general case where conflicts are given by a conflict hypergraph.*

4 Conclusions

We introduced our recent results on the complexity of the categoricity problem in the framework of preferred repairs [17], where the goal is to determine whether the provided priority relation suffices to repair the database unambiguously. We did not address any qualitative discrimination among the three notions of repairs. Rather, we continue the line of work [9, 10] that explores the impact of the choice on the entailed computational complexity. It has been established that, as far as repair checking is concerned, the Pareto and the completion semantics behave way better than the global one [9]. In this work we have shown that from the viewpoint of categoricity, the Pareto semantics departs from the completion one by being likewise intractable (while the global semantics hits an even higher complexity class); hence, the completion semantics outstands so far as the most efficient option to adopt.

References

1. F. N. Afrati and P. G. Kolaitis. Repair checking in inconsistent databases: algorithms and complexity. In *ICDT*, pages 31–41. ACM, 2009.
2. D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *TIPSTER Text Program: Phase III*, pages 23–30. Association for Computational Linguistics, 1998.
3. M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *PODS*, pages 68–79. ACM, 1999.
4. P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *ICDE*, pages 746–755. IEEE, 2007.
5. Y. Cao, W. Fan, and W. Yu. Determining the relative accuracy of attributes. In *SIGMOD*, pages 565–576. ACM, 2013.
6. L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, and S. Vaithyanathan. SystemT: An algebraic approach to declarative information extraction. In *ACL*, pages 128–137, 2010.
7. J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Inf. Comput.*, 197(1-2):90–121, 2005.
8. R. Fagin, B. Kimelfeld, and P. G. Kolaitis. Dichotomies in the complexity of preferred repairs. In *PODS*, pages 3–15. ACM, 2015.
9. R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Cleaning inconsistencies in information extraction via prioritized repairs. In *PODS*, pages 164–175. ACM, 2014.
10. R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Document spanners: A formal approach to information extraction. *J. ACM*, 62(2):12, 2015.
11. W. Fan, F. Geerts, and J. Wijsen. Determining the currency of data. *ACM Trans. Database Syst.*, 37(4):25, 2012.
12. W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *VLDB J.*, 21(2):213–238, 2012.
13. W. Fan, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. *J. Data and Information Quality*, 4(4):16:1–16:38, 2014.
14. T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *J. Intell. Inf. Syst.*, 1(2):123–157, 1992.
15. P. Koutris and J. Wijsen. The data complexity of consistent query answering for self-join-free conjunctive queries under primary key constraints. In *PODS*, pages 17–29. ACM, 2015.
16. S. Staworko, J. Chomicki, and J. Marcinkowski. Preference-driven querying of inconsistent relational databases. In *EDBT Workshops*, volume 4254 of *LNCS*, pages 318–335. Springer, 2006.
17. S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intell.*, 64(2-3):209–246, 2012.