Dichotomies in the Complexity of Preferred Repairs

Ronald Fagin
IBM Research – Almaden
fagin@us.ibm.com

Benny Kimelfeld
Technion, Israel
& LogicBlox, Inc.
bennyk@cs.technion.ac.il

Phokion G. Kolaitis
UC Santa Cruz
& IBM Research – Almaden
kolaitis@cs.ucsc.edu

ABSTRACT

The framework of database repairs provides a principled approach to managing inconsistencies in databases. Informally, a repair of an inconsistence database is a consistent database that differs from the inconsistent one in a "minimal way." A fundamental problem in this framework is the repair-checking problem: given two instances, is the second a repair of the first? Here, all repairs are taken into account, and they are treated on a par with each other. There are situations, however, in which it is natural and desired to prefer one repair over another; for example, one data source is regarded to be more reliable than another, or timestamp information implies that a more recent fact should be preferred over an earlier one. Motivated by these considerations, Staworko, Chomicki and Marcinkowski 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 to define notions of preferred repairs. In this paper we focus on the globally-optimal repairs, in the case where the constraints are functional dependencies. Intuitively, a globally-optimal repair is a repair that cannot be improved by exchanging facts with preferred facts. In this setting, it is known that there is a fixed schema (i.e., signature and functional dependencies) where globally-optimal repair-checking is coNP-complete.

Our main result is a dichotomy in complexity: for each fixed relational signature and each fixed set of functional dependencies, the globally-optimal repair-checking problem either is solvable in polynomial time or is coNP-complete. Specifically, the problem is solvable in polynomial time if for each relation symbol in the signature, the functional dependencies are equivalent to either a single functional dependency or to a set of two key constraints; in all other cases, the globally-optimal repair-checking problem is coNP-complete. We also show that there is a polynomial-time algorithm for distinguishing between the tractable and the intractable cases. The setup of preferred repairs assumes that preferences are only between conflicting facts. In the last part of the paper, we investigate the effect of this assumption on the complexity of globally-optimal repair checking. With this assumption relaxed, we give another dichotomy theorem and another polynomial-time distinguishing algorithm. Interestingly, the two dichotomies turn out to have quite different conditions for distinguishing tractability from intractability.

PODS'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.

http://dx.doi.org/10.1145/2745754.2745762.

Categories and Subject Descriptors

H.2 [**Database Management**]: Miscellaneous

General Terms

Theory, Algorithms

Keywords

Inconsistent databases; database repairs; repair checking; preferred repairs; dichotomy in complexity

1. INTRODUCTION

Managing inconsistency in databases is a long-standing problem. An inconsistent database is a database that fails to satisfy one or more integrity constraints assumed to hold. Inconsistent databases arise for different reasons and in different applications; for example, they may arise if integrity constraints are not properly enforced or when integrating data distributed over different sources. Arenas, Bertossi and Chomicki [3] introduced a principled approach to the management of inconsistency by formulating the notions of a *repair* of an inconsistent database and of the *consistent answers* of a query. Informally, a *repair* of an inconsistent database I is a consistent database J that differs from I in a "minimal" way. The standard definition of minimality refers to the *symmetric difference*, and in the case of functional dependencies this means that J is a *subset repair* (i.e., J is a subinstance of I) that is not properly contained in any consistent subinstance of I. The *consistent answers* of a query q on an inconsistent database I are given by the intersection $\bigcap\{q(J) : J \text{ is a repair of } I\}$. Thus, the inconsistencies in the database are handled at query time by considering all repairs and returning the tuples that are guaranteed to be in the result of the query on every repair.

The *repair checking* problem (i.e., given instances I and J, is J a repair of I?) and the *consistent query answering* problem (i.e., compute the consistent answers of a query q on a given instance I) are the two main algorithmic problems in the framework of database repairs. Since the publication of [3], these two problems have been extensively

studied for different types of repairs and for different types of constraints. Depending on the type of repairs and the type of constraints, these problems may vary from solvable in polynomial time (e.g., the repair-checking problem for subset repairs and functional dependencies) to undecidable (e.g., the consistent query answering problem for conjunctive queries, symmetric-difference repairs, and tuple-generating dependencies [15]); see [4] for an overview of results.

In the above framework, all repairs of a given database instance are taken into account, and they are treated on a par with each other. This often results in having a large number of repairs, which, in turn, may lead to some of the high complexity results for the consistent query answering problem. There are situations, however, in which it is natural to prefer one repair over another; for example, this is the case if one source is regarded to be more reliable than another or if available timestamp information implies that a more recent fact should be preferred over an earlier fact. Motivated by these considerations, Staworko, Chomicki and Marcinkowski [14] 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 to define a notion of *preferred* repairs. Specifically, a *globally-optimal* repair is, intuitively, a repair that cannot be improved by exchanging facts with preferred facts. (The formal definition is in Section 2.4.)

Fagin et al. [7] have built on the concept of preferred repairs (in conjunction with the framework of *document spanners* [6]) 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 existing systems for text analytics [2, 5].

Unfortunately, the notion of globally-optimal repairs may incur high computational complexity; in particular, Staworko et al. [14] showed that there is a fixed schema with functional dependencies, where globally-optimal repair checking is coNP-complete. For this reason, in addition to globally-optimal repairs, they considered alternative notions of preferred repairs, namely, *Pareto-optimal* repairs and *completion-optimal* repairs, where repair checking is solvable in polynomial time.

In this paper, we aim to characterize the class of schemas for which the problem of globally-optimal repair checking is indeed intractable. We investigate in depth the computational complexity of globally-optimal repair checking when the constraints are functional dependencies. Our main result is a *dichotomy* theorem: for each fixed relational signature and each fixed set of functional dependencies, the globally-optimal repair-checking problem either is solvable in polynomial time or is coNP-complete. Specifically, we show that the problem is solvable in polynomial time if for each relation symbol R in the signature, the functional dependencies on R are equivalent to either a single functional dependency or to a set of two key constraints; in all other cases, the problem is coNP-complete. We also give a polynomial-time algorithm for distinguishing between the tractable and the intractable cases.

It should be pointed out that, to this day, only a few dichotomy theorems for database repairs have been obtained. Moreover, conjectures involving the existence of dichotomy theorems for database repairs have resisted resolution, in spite of concerted efforts by different groups of researchers. Consider, for example, the consistent query answering problem for boolean conjunctive queries under key constraints. It was conjectured in [1] that for each conjunctive query and for each set of key constraints, this problem is either solvable in polynomial time or coNP-complete. While some dichotomy theorems for special cases of conjunctive queries and key constraints have been obtained (e.g., [11, 12]), the dichotomy question for the general problem remains open to date. An explanation as to why establishing dichotomy theorems for database repairs may be a challenging task was provided by Fontaine [9], who showed that a dichotomy theorem for unions of conjunctive queries and GAV constraints implies a dichotomy theorem for the *constraint satisfaction* problem, thus resolving the celebrated Feder-Vardi conjecture [8].

We now outline the strategy we developed to establish the dichotomy theorem for the globally-optimal repair-checking problem. As is often the case with other dichotomy theorems, one first identifies certain polynomial-time solvable cases, and then the challenge is to establish that all other cases are hard, which means that they are coNP-complete in our case. The hardness side of our dichotomy theorem is established in two separate steps. The first step is to show that the globally-optimal repair-checking problem is coNP-complete for six different concrete schemas, where a *schema* is a relational signature and a set of functional dependencies. The second step is to show that for every arbitrary schema that does not fall in one of the polynomial-time cases, there is a delicate polynomial-time reduction that, intuitively, preserves consistency and inconsistency from one of the six concrete hard schemas to the schema at hand.

As mentioned earlier, globally-optimal repairs use a priority relation that imposes preferences between conflicting facts. In the last part of the paper, we relax this assumption by considering globally-optimal repairs based on *cross-conflict* priority relations, i.e., priority relations that impose preferences between facts that need not necessarily conflict (for example, one may prefer using facts from one source over another source, even if the facts are not conflicting). We establish a dichotomy theorem for globally-optimal repair checking under cross-conflict priority relations. Specifically, we show that if all functional dependencies are single key constraints or if all functional dependencies are of the form $\emptyset \rightarrow B$, for some attribute B, then the globally-optimal repair-checking problem is solvable in polynomial time; in all other cases, the globally-optimal repair-checking problem is coNP-complete. Again, we show that there is a polynomial-time algorithm for distinguishing between the tractable and the intractable cases.

The dichotomy theorems established in this paper yield a complete classification of the computational complexity of the globally-optimal repair-checking problem, when the constraints are functional dependencies. As the other semantics of preferred repairs (namely, Pareto and completion) admit polynomial-time repair checking [14], our theorems complete the picture for the complexity of preferred-repair checking. Moreover, we believe that the tools developed in this paper may be deployed to establish other complexity classifications in the study of preferred repairs. In particular, they may pave the road towards the classification of the computational complexity of the other major algorithmic problem for repairs, namely, that of consistent query answering, in the framework of preferred repairs.

2. PRELIMINARIES AND BASIC NOTIONS

In this section, we describe the formal setup for this paper, including the framework of preferred repairs.

2.1 Relational Signatures

A *relational signature* or, simply, a *signature* $\mathcal{R}$ is a finite set $\{R_1, \ldots, R_n\}$ of *relation symbols* each with a designated positive integer as its *arity*, denoted $\text{arity}(R_i)$. We assume an infinite set Const of constants that are used as values in database instances. More formally, an *instance* I over a signature $\mathcal{R} = \{R_1, \ldots, R_n\}$ consists of finite relations $R_i^I \subseteq \mathsf{Const}^{\text{arity}(R_i)}$, where $R_i \in \mathcal{R}$. We write $[\![R_i]\!]$ to denote the set $\{1, \ldots, \text{arity}(R_i)\}$, and we refer to the members of $[\![R_i]\!]$ as *attributes* or *indices* of R_i. If I is an instance over $\mathcal{R}$ and $\mathbf{t}$ is a tuple in R_i^I, then we say that $R_i(\mathbf{t})$ is a *fact of* I. Every instance I can be identified with the set of its facts. Thus, $J \subseteq I$ means that $R_i^J \subseteq R_i^I$, for every $R_i \in \mathcal{R}$; in this case, we say that J is *subinstance* of I.

EXAMPLE 2.1. We now introduce our running example. The signature $\mathcal{R}$ consists of a ternary relation symbol

$$\mathit{BookLoc}(\text{isbn}, \text{genre}, \text{lib})$$

that specifies in which libraries book copies can be found, and a binary relation symbol

$$\mathit{LibLoc}(\text{lib}, \text{loc})$$

that describes library locations. Our formalism does not include the attributes names (e.g., "isbn"), but rather refers to them by positions in tuples (e.g., "isbn" is attribute 1 in *BookLoc*). Figure 1 depicts an instance I over $\mathcal{R}$. To ease following our running example, the subscript of the symbol that represents each fact is encoding the content of that fact. For example, in g_{1f1} the first "1" stands for "$\mathsf{b1}$," "f" stands for "$\mathsf{fiction}$," and the second "1" stands for "$\mathsf{lib1}$." The reader can easily observe, without referring to Figure 1, that the facts g_{1f1} and f_{1d3} agree on the first attribute (isbn) but not on the second (genre). Such observations will be useful later on in the paper. □

2.2 FDs, Schemas and Instances

Let $\mathcal{R}$ be a signature. A *functional dependency* (*fd*) over $\mathcal{R}$ is an expression of the form $R : A \to B$, where R is a relation symbol of $\mathcal{R}$, and A and B are subsets of $[\![R]\!]$. A *schema* $\mathbf{S}$ is a pair $(\mathcal{R}, \Delta)$, where $\mathcal{R}$ is signature and Δ is a set of fds over $\mathcal{R}$. Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema, let I be an instance over $\mathcal{R}$, and let δ be an fd $R : A \to B$ in Δ. A pair $\{f_1, f_2\}$ of facts in I is a *δ-conflict* if f_1 and f_2 agree on (that is, have the same values for) all the attributes in A, but disagree on at least one attribute in B. We say that I *satisfies* δ, denoted $I \models \delta$, if I contains no δ-conflict. We say that fact f_1 *conflicts with* a fact f_2, or that f_1 and f_2 are *conflicting facts*, if $\{f_1, f_2\}$ is a δ-conflict for some $\delta \in \Delta$. We say that I *satisfies* Δ, denoted $I \models \Delta$, if I satisfies every fd in Δ (that is, I does not contain conflicting facts); in that case, we also say that I is a *consistent instance* (*w.r.t.* $\mathbf{S}$).

We now introduce some terminology, which will be used in the sequel. Let $(\mathcal{R}, \Delta)$ be a schema. If R is a relation symbol in $\mathcal{R}$, then we write $\Delta_{|R}$ to denote the subset of Δ that consists of all fds $R' : A \to B$ such that $R' = R$. If A is a singleton $\{a\}$, then we may write $R : a \to B$, instead of $R : A \to B$. Similarly, if $B = \{b\}$, then we may write $R : A \to b$, and if $A = \{a\}$ and $B = \{b\}$, then we may write $R : a \to b$. Moreover, if R is clear from the context, then we may omit R from $R : A \to B$ and write just $A \to B$. We will often consider the following special cases of fds.

- The fd $R : A \to B$ is *trivial* if $B \subseteq A$. Note that a trivial fd is satisfied by every instance.
- The fd $R : A \to B$ is a *key constraint* if $B = [\![R]\!]$. We may sometimes refer to a key constraint as simply a *key*.

Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema. The *closure* Δ^+ of Δ is the set of all fds that are logically implied by Δ. Note that Δ^+ contains every fd in Δ and every trivial fd. As an example, if $\mathcal{R}$ contains a ternary relation symbol R and Δ consists of the fds $R : 1 \to 2$ and $R : 2 \to 3$, then Δ^+ contains, among others, the fds $R : 1 \to 3$, $R : \{1, 2\} \to 3$, and $R : 3 \to 3$. Let R be a relation symbol of $\mathcal{R}$, and let A be a subset of $[\![R]\!]$. The *closure of A under Δ and R*, denoted $[\![R.A^\Delta]\!]$, is the set of all indices i such that $R : A \to i$ is in Δ^+. Note that for every set B of indices, the fd $R : A \to B$ is in Δ^+ if and only if $B \subseteq [\![R.A^\Delta]\!]$.

Two sets Δ_1 and Δ_2 of fds over a signature $\mathcal{R}$ are *equivalent* if $\Delta_1^+ = \Delta_2^+$. In other words, Δ_1 and Δ_2 are equivalent if the schemas $(\mathcal{R}, \Delta_1)$ and $(\mathcal{R}, \Delta_2)$ have the same set of consistent instances.

EXAMPLE 2.2. We expand on our running example. Consider the schema $\mathbf{S} = (\mathcal{R}, \Delta)$, where $\mathcal{R}$ was defined in Example 2.1 and Δ consists of the following fds:

$$\delta_1 \stackrel{\text{def}}{=} \mathit{BookLoc} : 1 \to 2$$

$$\delta_2 \stackrel{\text{def}}{=} \mathit{LibLoc} : 1 \to 2$$

$$\delta_3 \stackrel{\text{def}}{=} \mathit{LibLoc} : 2 \to 1$$

In words, δ_1 states that in *BookLoc* a book's isbn determines its genre (i.e., two tuples with the same isbn must agree on the genre), δ_2 states that in *LibLoc* a library determines the location, and δ_3 states that every location has one library. The instance I of Figure 1 violates Δ; for example, $\{g_{\mathsf{1f1}}, f_{\mathsf{1d3}}\}$ is a δ_1-conflict, $\{d_{\mathsf{1e}}, e_{\mathsf{1b}}\}$ is a δ_2-conflict, and $\{d_{\mathsf{1a}}, g_{\mathsf{2a}}\}$ is a δ_3-conflict. Note also that δ_2 and δ_3 are key constraints. Moreover, we have that $\Delta_{|\mathit{BookLoc}} = \{1 \to 2\}$ and $\Delta_{|\mathit{LibLoc}} = \{1 \to 2, 2 \to 1\}$. An example of an fd in Δ^+ that is not in Δ is $\mathit{BookLoc} : \{1, 3\} \to \{1, 2\}$. Finally, note that $[\![\mathit{BookLoc}.\{1\}^\Delta]\!] = \{1, 2\}$ and $[\![\mathit{BookLoc}.\{1, 3\}^\Delta]\!] = \{1, 2, 3\}$. □

2.3 Prioritizing Instances

Let $\mathcal{R}$ be a signature. Assume that I is an instance over $\mathcal{R}$, and $\succ$ is a binary relation on the facts of I. A *cycle* in $\succ$ is a sequence $f_1, \ldots, f_k$ of facts in I such that $f_i \succ f_{i+1}$ holds for all $i = 1, \ldots, k-1$, and $f_k \succ f_1$. We say that $\succ$ is *acyclic* if there are no cycles in $\succ$. In particular, we cannot have $f \succ f$ if $\succ$ is acyclic. A *prioritizing instance* over $\mathcal{R}$ is a pair $(I, \succ)$, where I is an instance over $\mathcal{R}$ and $\succ$ is an acyclic binary relation on the facts of I. We say that the relation $\succ$ is a *priority* on I. Thus, the statement $f \succ g$ should be interpreted as "the fact f has higher priority than the fact g."

Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema. An *inconsistent prioritizing instance* over $\mathbf{S}$ is a prioritizing instance $(I, \succ)$ over $\mathcal{R}$ such that I is inconsistent w.r.t.$\mathbf{S}$, and such that if f and g are facts of I with $f \succ g$, then f and g are conflicting facts. This

	BookLoc		
	isbn	genre	lib
g_{1f1}	b1	fiction	lib1
g_{1f2}	b1	fiction	lib2
f_{1d3}	b1	drama	lib3
f_{2p1}	b2	poetry	lib1
h_{3h2}	b3	horror	lib2

	LibLoc	
	lib	loc
d_{1a}	lib1	almaden
d_{1e}	lib1	edenvale
g_{2a}	lib2	almaden
f_{2b}	lib2	bascom
f_{3a}	lib3	almaden
f_{3c}	lib3	cambrian
e_{1b}	lib1	bascom
e_{3b}	lib3	bascom

Figure 1: Inconsistent database of the running example

requirement implies that, whenever $f \succ g$, it is necessarily the case that f and g are in the same relation of I, since all constraints in Δ are functional dependencies (hence, f and g violate Δ only if they belong to the same relation).

EXAMPLE 2.3. Recall the schema $\mathbf{S}$ of our running example. Consider the prioritizing instance $(I, \succ)$ consisting of the instance I of Figure 1 and the priority relation $\succ$ that is defined as follows:

- $g_y \succ f_x$ for all conflicting f_x and g_y;
- $e_y \succ d_x$ for all conflicting d_x and e_y.

As an example, $g_{1f1} \succ f_{1d3}$ and $e_{1b} \succ d_{1a}$.

Observe that $\succ$ is acyclic, as is required. □

2.4 Preferred Repairs

Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema, and let I be an inconsistent instance w.r.t. $\mathbf{S}$. Following Arenas et al. [3], we define a *repair* of I to be a maximal consistent subinstance J of I. That is, we cannot add any fact in I to J without violating consistency. Now let $(I, \succ)$ be an inconsistent prioritizing instance over $\mathbf{S}$. The priority relation $\succ$ that gives preferences among the tuples of I can be extended to a priority relation that gives preferences among the consistent subinstances of I. Staworko et al. [14] considered three such extensions, each of which gives a different notion of *preferred* repairs, namely, the notion of a *globally optimal* repair, a *Pareto-optimal* repair, and a *completion-optimal* repair. As mentioned in the Introduction, the repair-checking problem for the last two notions is in PTIME, while the repair-checking problem for globally-optimal repairs can be coNP-complete. Here, we give the precise definition of the notion of a *globally-optimal* repair, which is the focus of this paper. We also define the notion of a *Pareto-optimal* repair that we will use later in the paper.

DEFINITION 2.4. Let $(I, \succ)$ be an inconsistent prioritizing instance over a schema $\mathbf{S} = (\mathcal{R}, \Delta)$. Let J and J' be two consistent subinstances of I. We say that J is a *global improvement* of J' if $J \neq J'$, and for every fact $f' \in J' \setminus J$ there exists a fact $f \in J \setminus J'$ such that $f \succ f'$. We say that J is a *Pareto improvement* of J' is 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 *globally-optimal* repair of I if J does not have a global improvement. Similarly, J is a *Pareto-optimal* repair of I if J does not have a Pareto improvement. □

Note that every globally-optimal repair is Pareto-optimal; as we shall see shortly, the converse is not true. It is easy to see that every globally-optimal or Pareto-optimal repair is indeed a repair, as defined earlier.

One can also show that a consistent subinstance J is a globally-optimal repair of I if and only if no non-empty subset X of J can be replaced with a subset Y of $I \setminus J$, so that the subinstance $(J \setminus X) \cup Y$ is consistent and for every fact f' in X, there is a fact f in Y such that $f \succ f'$.

EXAMPLE 2.5. Let $(I, \succ)$ be the prioritizing instance of our running example. Consider the following subinstances of I.

- $J_1 \stackrel{\text{def}}{=} \{g_{1f1}, g_{1f2}, f_{2p1}, h_{3h2}, d_{1e}, f_{2b}, f_{3a}\}$
- $J_2 \stackrel{\text{def}}{=} \{g_{1f1}, g_{1f2}, f_{2p1}, h_{3h2}, d_{1e}, g_{2a}, e_{3b}\}$
- $J_3 \stackrel{\text{def}}{=} \{g_{1f1}, g_{1f2}, f_{2p1}, h_{3h2}, d_{1e}, f_{2b}, f_{3a}\}$
- $J_4 \stackrel{\text{def}}{=} \{g_{1f1}, g_{1f2}, f_{2p1}, h_{3h2}, e_{1b}, g_{2a}, f_{3c}\}$

Each J_i is consistent, and, in fact, a repair. Observe that $J_1 \setminus J_2 = \{f_{2b}, f_{3a}\}$ and $J_2 \setminus J_1 = \{g_{2a}, e_{3b}\}$; since $g_{2a} \succ f_{2b}$ and $g_{2a} \succ f_{3a}$, we get that J_2 is a Pareto (and global) improvement of J_1. The reader can verify that, as a matter of fact, J_2 is a globally-optimal (hence, Pareto-optimal) repair of I. The reader can also verify that J_3 does not have any Pareto improvement; in particular, J_4 is not a Pareto improvement of J_3 since $J_3 \setminus J_4 = \{d_{1e}, f_{2b}, f_{3a}\}$ and $J_4 \setminus J_3 = \{e_{1b}, g_{2a}, f_{3c}\}$, and no fact f in $J_4 \setminus J_3$ satisfies all of $f \succ d_{1e}$, $f \succ f_{2b}$ and $f \succ f_{3a}$. But J_4 is a global improvement of J_3, since $e_{1b} \succ d_{1e}$, $g_{2a} \succ f_{2b}$, and $g_{2a} \succ f_{3a}$. Hence, although J_3 is a Pareto-optimal repair, it is not a globally-optimal repair. □

3. MAIN RESULT

Our main result is about the complexity of preferred repair checking; this is the problem of deciding, given a subinstance of an inconsistent prioritized instance, whether the subinstance is a prioritized (i.e., Pareto-optimal or globally-optimal) repair. Staworko et al. [14] observed that, for every schema, this problem admits a polynomial-time solution under the Pareto semantics, and is in coNP under the global semantics. They also proved that for a specific schema with four fds, globally-optimal repair checking is coNP-complete. Our main result yields a complete classification of the complexity of globally-optimal repair checking.

THEOREM 3.1. *Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema. Globally-optimal repair checking can be solved in polynomial time if for every relation symbol $R \in \mathcal{R}$ at least one of the following holds.*

1. *$\Delta_{|R}$ is equivalent to a single fd.*
2. *$\Delta_{|R}$ is equivalent to a set of two key constraints.*

In every *other case, globally-optimal repair checking is coNP-complete.*

In the next two sections we discuss the proof of this theorem. In Section 6 we will show that one can test in polynomial time whether a given schema belongs to the tractable or the hard the side of the theorem. Before that, we give a few examples of applying the theorem.

EXAMPLE 3.2. In our running example, $\Delta_{|BookLoc}$ consists of a single fd, and $\Delta_{|LibLoc}$ is a pair of key constraints; hence, globally-optimal repair checking is solvable in polynomial time for this schema. □

EXAMPLE 3.3. Consider the schema $\mathbf{S} = (\mathcal{R}, \Delta)$ with $\mathcal{R}$ consisting of two ternary relation symbols R and S and a quaternary relation symbol T, and Δ consisting of the following fds.

$$R : 1 \to 2 \quad T : 1 \to \{2, 3, 4\} \quad T : \{2, 3\} \to 1$$

The schema $\mathbf{S}$ satisfies the condition of Theorem 3.1, for the following reasons:

- $\Delta_{|R}$ consists of a single fd;
- $\Delta_{|S}$ is empty, and hence, is equivalent to a single (trivial) fd such as $S : \emptyset \to \emptyset$;
- although $\Delta_{|T}$ is neither a single fd nor a pair of keys, it is equivalent to $\{T : 1 \to \{1, 2, 3, 4\}, T : \{2, 3\} \to \{1, 2, 3, 4\}\}$, which is a pair of keys.

Therefore, globally-optimal repair checking is solvable in polynomial time for $\mathbf{S}$. □

EXAMPLE 3.4. Each of the following six schemas violates the condition of Theorem 3.1, and so, is such that the globally-optimal repair-checking problem is coNP-complete. These schemas have the form $\mathbf{S}^i = (\mathcal{R}^i, \Delta^i)$ for $i = 1, \ldots, 6$, where $\mathcal{R}^i$ consists of a single ternary relation symbol R^i. The Δ^i are defined as follows.

1. $\Delta^1 = \{\{1, 2\} \to 3, \{1, 3\} \to 2, \{2, 3\} \to 1\}$
2. $\Delta^2 = \{1 \to 2, 2 \to 1\}$
3. $\Delta^3 = \{\{1, 2\} \to 3, 3 \to 2\}$
4. $\Delta^4 = \{1 \to 2, 2 \to 3\}$
5. $\Delta^5 = \{1 \to 3, 2 \to 3\}$
6. $\Delta^6 = \{\emptyset \to 1, 2 \to 3\}$

As we will show in Section 5, these specific schemas play an important role in the proof of the hardness part of Theorem 3.1. □

3.1 Proof Strategy

A straightforward observation is that, to prove Theorem 3.1, it suffices to consider schemas with a single relation, since each of the constraints we consider is an fd, hence applied to a single relation, and preferences are applied to conflicting facts, hence facts from the same relation. (In Section 7 we study the impact of avoiding the restriction of priorities to conflicting facts.) Formally, we have the following proposition.

PROPOSITION 3.5. *Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema. The following are equivalent.*

1. *Globally-optimal repair checking is solvable in polynomial time for $\mathbf{S}$.*
2. *For every relation symbol $R \in \mathcal{R}$, globally-optimal repair checking is solvable in polynomial time for the schema $\{\{R\}, \Delta_{|R}\}$.*

Moreover, the following are equivalent as well.

1. *Globally-optimal repair checking is coNP-complete for the schema $\mathbf{S}$.*
2. *For at least one relation symbol $R \in \mathcal{R}$, globally-optimal repair checking is coNP-complete for the schema $\mathbf{S}_R = \{\{R\}, \Delta_{|R}\}$.*

Hence, our proof (discussed in the next two sections) is restricted to schemas with a single relation symbol.

4. ALGORITHMS FOR THE TRACTABLE SCHEMAS

In this section, we fix a schema $\mathbf{S} = (\mathcal{R}, \Delta)$, such that $\mathcal{R}$ consists of a single relation symbol R. We will prove that globally-optimal repair checking is solvable in polynomial time if Δ (which is the same as $\Delta_{|R}$) satisfies one of the two conditions of Theorem 3.1. We begin with the first condition.

4.1 Single FD

The case of a single FD seems, on the face of it, to have been resolved by Staworko et al. [14]. Specifically, their Proposition 10 (iii) states that global and completion optimality coincide in the case of a single FD, and their Corollary 4 states that completion optimality can be tested in polynomial time. Hence, by combining these two results it follows that in the case of a single FD, global optimality can be tested in polynomial time. Unfortunately, Proposition 10 (iii) in [14] is incorrect, as we have established in private communication with the authors of [14]. In this section we give a proof of the polynomial-time upper bound for the case of a single FD .

We assume that Δ is the singleton $\{A \to B\}$. Consider the input $(I, \succ)$ and J for globally-optimal repair checking. Since J is a repair, by definition J is a maximal consistent subset of I, and so $J \cup \{f\}$ is inconsistent for every fact $f \in I \setminus J$. Two facts f and g in I are said to *agree* on A (respectively, B) if f and g have the same value in every position in A (respectively, B).

Let f and g be two facts in I, such that

1. $f \in J$,
2. f and g agree on A, and
3. f and g disagree in B.

Note that $g \notin J$ since we assume that J is consistent. We denote by $J[f \leftrightarrow g]$ the instance that is obtained from J by removing all the facts in I that agree with f on A and B, and adding to J all the facts that agree with g on A and B.

EXAMPLE 4.1. Continuing with our running example, we now restrict our attention to *BookLoc* and ignore *LibLoc*. So now we have a single fd, namely $1 \to 2$. Consider the subinstances $J = \{g_{1f1}, g_{1f2}, f_{2p1}\}$ and $J' = \{f_{1d3}, f_{2p1}\}$ (of the instance I in Figure 1). Observe that g_{1f1} and f_{1d3} agree on the first attribute (isbn) but disagree on the second (genre). Then $J[g_{1f1} \leftrightarrow f_{1d3}] = J'$ and $J'[f_{1d3} \leftrightarrow g_{1f1}] = J$. In particular, observe that $J[g_{1f1} \leftrightarrow f_{1d3}]$ misses both g_{1f1} and g_{1f2}, and that $J'[f_{1d3} \leftrightarrow g_{1f1}]$ includes both g_{1f1} and g_{1f2}. □

The following are straightforward observations.

Algorithm GRepCheck1FD(I, J)

1: **for all** conflicting facts $f \in J$ and $g \in I \setminus J$ **do**
2: **if** $J[f \leftrightarrow g]$ is a global improvement of J **then**
3: **return false**
4: **end if**
5: **end for**
6: **return true**

Figure 2: Globally-optimal repair checking in the case where the schema consists of a single relation symbol R and a single fd $A \to B$

1. $J[f \leftrightarrow g]$ is consistent (that is, $A \to B$ is satisfied).
2. Whether $J[f \leftrightarrow g]$ is a global improvement of J can be tested in polynomial time.

Consequently, to show that J is not a globally-optimal repair, it is sufficient to find some f and g as above, such that $J[f \leftrightarrow g]$ is a global improvement of J. The next lemma shows that this procedure is also necessary to show that J is not a globally-optimal repair.

LEMMA 4.2. *If J has a global improvement, then there are facts f and g (as defined above) such that $J[f \leftrightarrow g]$ is a global improvement of J.*

PROOF. Suppose that J' is a global improvement of J. Let f be a fact in $J \setminus J'$. Note that such f indeed exists, since we assumed J cannot be extended without violating Δ. Let $g \in J'$ be a fact such that $g \succ f$. Then f and g are as defined above, that is, f and g agree on A but disagree on B. So, it remains to prove that $J[f \leftrightarrow g]$ is a global improvement of J. In other words, we need to show that if $\hat{f}$ is a fact in J that agrees with f on A (hence, on B), then there is a fact $\hat{g}$ in I that agrees with g on A and B (hence, $\hat{g} \in J[f \leftrightarrow g]$), such that $\hat{g} \succ \hat{f}$. So, let $\hat{f}$ be such a fact. Since $\hat{f}$ disagrees with g on B, it must be the case that $\hat{f}$ is not in J' (since J' contains g, and J' is consistent). Therefore, $J' \setminus J$ contains a fact g' such that $g' \succ \hat{f}$. Let $\hat{g}$ be such a fact g'. Then $\hat{g}$ and $\hat{f}$ agree on A, and so $\hat{g}$ and g agree on A. And since J' contains both g and $\hat{g}$, we have that g and $\hat{g}$ agree on B. It follows that $\hat{g}$ is as claimed. □

Consequently, we conclude that the simple (and obviously polynomial-time) algorithm GRepCheck1FD of Figure 2 solves globally-optimal repair checking in the case of this section.

4.2 Two Key Constraints

We now consider the case where Δ is equivalent to two key constraints, which we shall refer to as simply "two keys." For presentation sake, we give the algorithm for the specific case where R is binary and $\Delta = \{1 \to 2, 2 \to 1\}$. The generalization to the general case of two keys will be straightforward, as we shall discuss. For the inputs $(I, \succ)$ and J, the idea is as follows. To improve J, we try to replace a fact $R(a_1, a_2)$ in J with a preferred fact in $I \setminus J$, say $R(a_1', a_2)$; if we succeed (i.e., the resulting instance is consistent), then the replacement results in a Pareto (and in particular global) improvement, and we are done. Otherwise, $R(a_1', a_2)$ conflicts with a fact $R(a_1', a_2')$ in J. ($R(a_1', a_2)$ cannot conflict with a fact that has a_2 as the second attribute, since only $R(a_1, a_2)$ has this property in J.) So, we also need to replace $R(a_1', a_2')$ with a preferred fact $R(a_1'', a_2')$. We continue with this process and, assuming that J does not have a Pareto improvement (which can be tested in polynomial time), we eventually succeed (find a global improvement) in this process if we close a cycle (that is, each fact we remove is improved by a fact we added). Next, we formalize this idea.

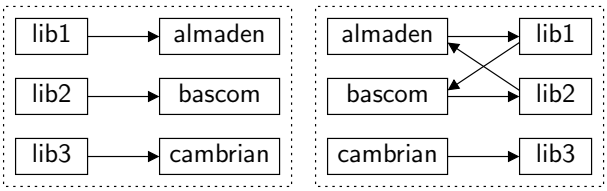

Figure 3: The graphs G_J^{12} (left) and G_J^{21} (right)

Consider the given inputs $(I, \succ)$ and J. For $i = 1, 2$, denote by $[J]_i$ the set of constants that occur in the ith component of J. Denote by G_J^{12} the bipartite directed graph that has $[J]_1$ on its left side, $[J]_2$ on its right side, and the following edges:

- $a_1 \to a_2$ for every $R(a_1, a_2) \in J$;
- $a_1' \leftarrow a_2$ for every $R(a_1', a_2) \in I \setminus J$ such that $R(a_1', a_2) \succ R(a_1, a_2)$ for some $R(a_1, a_2) \in J$.

Similarly, denote by G_J^{21} the bipartite directed graph that has $[J]_2$ on its left side, $[J]_1$ on its right side (that is, we swap between the two sides of G^{12}), and the following edges:

- $a_2 \to a_1$ for every $R(a_1, a_2) \in J$;
- $a_2' \leftarrow a_1$ for every $R(a_1, a_2') \in I \setminus J$ such that $R(a_1, a_2') \succ R(a_1, a_2)$ for some $R(a_1, a_2) \in J$.

EXAMPLE 4.3. Continuing with our running example, we now restrict to *LibLoc* and ignore *BookLoc*. So now we have the fds $1 \to 2$ and $2 \to 1$. Consider the subinstance $J = \{d_{1a}, f_{2b}, f_{3c}\}$ (of the instance I in Figure 1). Figure 3 depicts the graphs G_J^{12} and G_J^{21}. Observe that G_J^{12} does not have right-to-left edges (since no relevant priorities exist), and G_J^{21} has two such edges. The edge from lib2 to almaden is due to $g_{2a} \succ f_{2b}$, and the edge from lib1 to bascom is due to $e_{1b} \succ d_{1a}$. □

Our algorithm is due to the following characterization of (not) being a globally-optimal repair.

LEMMA 4.4. *Assume that J is a consistent subinstance. Then J has a global improvement if and only if at least one of the following conditions is true:*

1. *J has a Pareto improvement;*
2. *G_J^{12} has a cycle;*
3. *G_J^{21} has a cycle.*

PROOF. We prove each direction separately.

The "if" direction. Since a Pareto improvement is also a global one, we get that if the first condition is true then

J has a global improvement. So, suppose that the second condition is true, that is, G_J^{12} has a cycle. Let $a_1^1 \to a_2^1 \to a_1^2 \to \cdots \to a_1^n$ be a simple cycle in G_J^{12}, where each a_1^i is in $[J]_1$, each a_2^i is in $[J]_2$, and $a_1^1 = a_1^n$. Let F be the set of facts $R(a_1^i, a_2^i)$ for $i = 1, \ldots, n-1$, and let F' be the set of facts $R(a_1^{i+1}, a_2^i)$ for $i = 1, \ldots, n-1$. We will prove that $(J \setminus F) \cup F'$ is a global improvement of J.

Observe that $(J \setminus F) \cup F'$ satisfies Δ, since both J and F' satisfy Δ, and $(J \setminus F)$ and F' do not share any left component or any right component. From the definition of G_J^{12} it follows that F' is a subset of $I \setminus J$. It also follows from the definition of G_J^{12} that each $R(a_1^{i+1}, a_2^i)$ is a fact in I, and there exists some fact $R(a_1', a_2^i) \in J$ such that $R(a_1^{i+1}, a_2^i) \succ R(a_1', a_2^i)$; but then, since also $R(a_1^i, a_2^i) \in J$ and J satisfies $2 \to 1$, it follows that a_1' is necessarily a_1^i. We conclude that every fact $f \in F$ has a fact $f' \in F'$ such that $f' \succ f$. Hence, $(J \setminus F) \cup F'$ is a global improvement of J, as claimed.

The proof that the third condition implies that J has a global improvement is symmetric to that of the second condition.

The "only if" direction. We assume that J has a global improvement, and we need to prove that at least one of the three conditions holds true. So assume that J does not have a Pareto improvement; we will prove that at least one of G_J^{12} and G_J^{21} has a cycle.

Suppose that $(J \setminus F) \cup F'$ is a global improvement of J. Then F is necessarily nonempty, since otherwise $(J \setminus F) \cup F'$ is a Pareto improvement of J. Let $R(a_1^1, a_2^1)$ be a fact in F. Then there exists a fact $f_1' \in F'$ such that $f_1' \succ R(a_1^1, a_2^1)$, so f_1' conflicts with $R(a_1^1, a_2^1)$. We first consider the case where $f_1' = R(a_1^2, a_2^1)$ for some a_1^2. We will construct an infinite path in G_J^{12}, starting with $a_1^1 \to a_2^1 \to a_1^2$, where the edges alternate between corresponding to facts in F and facts in F'. If J does not contain any fact with a_1^2 in its first component, then $(J \setminus \{R(a_1^1, a_2^1)\}) \cup \{f_1'\}$ is a Pareto improvement of J, in contradiction to our assumption. So, let $R(a_1^2, a_2^2) \in J$ be such a fact. Then $R(a_1^2, a_2^2)$ cannot be in $J \setminus F$, since it conflicts with a member of F'; hence, $R(a_1^2, a_2^2)$ is in F. We claim that $f_1' \not\succ R(a_1^2, a_2^2)$. Indeed, otherwise we could obtain a Pareto improvement of J by removing $R(a_1^1, a_2^1)$ and $R(a_1^2, a_2^2)$ and adding $f_1' = R(a_1^2, a_2^1)$. Since $R(a_1^2, a_2^2)$ is in F, it follows that F' contains a fact f_2' such that $f_2' \succ R(a_1^2, a_2^2)$. From what we showed earlier, we know that $f_2' \neq f_1'$. Observe that f_2' conflicts with $R(a_1^2, a_2^2)$, but it cannot have a_1^2 as its first component, since it would then conflict with f_1'; hence, f_2' conflicts with $R(a_1^2, a_2^2)$ on the second component, and so is of the form $R(a_1^3, a_2^2)$. So we add $a_1^2 \to a_2^2 \to a_1^3$ to our path. We can then continue to do so indefinitely. In particular, G_J^{12} contains a cycle since G_J^{12} is finite.

When we consider the case where f_1' is of the form $R(a_1^1, a_2^2)$ for some a_2^2, we similarly get to the conclusion that G_J^{21} contains cycle. This concludes our proof. □

As we said above, the proof extends straightforwardly to the case where Δ is a set of two keys on a relation. In particular, suppose that $\Delta = \{A_1 \to [\![R]\!], A_2 \to [\![R]\!]\}$. We assume that $A_1 \not\subseteq A_2$ and $A_2 \not\subseteq A_1$, since otherwise one of the fds can be removed (and then we are in the previous case). For a fact f over R, we denote by $f[A_i]$, where $i \in \{1, 2\}$, the tuple that is obtained from f by taking the components in the positions of A_i in some predefined order. Moreover, in the graph G_J^{12} we now have the following edges:

Algorithm GRepCheck2Keys(I, J)

1: **if** J has a Pareto improvement **then**
2: **return false**
3: **end if**
4: **if** both G_J^{12} and G_J^{21} are acyclic **then**
5: **return true**
6: **else**
7: **return false**
8: **end if**

Figure 4: Globally-optimal repair checking in the case where the schema consists of a single relation symbol R and the key constraints $A_1 \to [\![R]\!]$ and $A_2 \to [\![R]\!]$

- $f[A_1] \to f[A_2]$ for every fact $f \in J$;
- $f'[A_1] \leftarrow f'[A_2]$ for every fact $f' \in I \setminus J$ such that $f[A_2] = f'[A_2]$ for some $f \in J$ such that $f' \succ f$.

We similarly extend the definition of G_J^{21}. Observe that in every edge of G_J^{12} and G_J^{21}, the two endpoints agree on all the attributes of $A_1 \cap A_2$.

Consequently, the algorithm GRepCheck2Keys of Figure 4 solves globally-optimal repair checking in the case of this section. This algorithm terminates in polynomial time, since both having a Pareto improvement and graph acyclicity can be tested in polynomial time.

5. PROOF STRATEGY FOR HARDNESS

In this section, we describe our proof of the hardness side of Theorem 3.1, namely, if the condition is violated then globally-optimal repair checking is coNP-complete.

5.1 General Strategy

Our proof strategy consists of two steps, similarly to the proof for the dichotomy of the complexity in deletion propagation by Kimelfeld [10].

In the first step, we consider several specific schemas, and prove that globally-optimal repair checking is coNP-complete for these schemas. The specific schemas we consider are precisely those of Example 3.4.

LEMMA 5.1. *For each of the six schemas $\mathbf{S}^1, \ldots, \mathbf{S}^6$ of Example 3.4, globally-optimal repair checking is coNP-complete.*

Next, we consider an arbitrary schema $\mathbf{S}$ that violates the condition of Theorem 3.1 and define a reduction from globally-optimal repair checking for one of the schemas of Example 3.4 to globally-optimal repair checking for $\mathbf{S}$. The specific choice of the schema $\mathbf{S}^i$ from Example 3.4 depends on a case analysis that we describe later in this section. All of our reductions follow a general pattern that we describe next.

Suppose that we want to reduce globally-optimal repair checking in $\mathbf{S}^i$ (which is one of the six schemas in Example 3.4) to globally-optimal repair checking in $\mathbf{S}$ (which is an arbitrary schema that violates the condition of Theorem 3.1). Recall that $\mathbf{S}^i$ consists of a single relation symbol

R^i. Also recall from Proposition 3.5 that we can assume that $\mathbf{S}$ consists of a single relation symbol, say R. We begin with the input $(I^i, \succ^i)$ and J^i for globally-optimal repair checking under $\mathbf{S}^i$, and construct an input $(I, \succ)$ and J for $\mathbf{S}$. The construction is done by defining a function Π that takes as input fact f^i from I^i and constructs, in constant time, a fact $\Pi(f^i)$ over $\mathbf{S}$. For a subinstance K^i of I^i we define $\Pi(K^i) = \{\Pi(f^i) \mid f^i \in K^i\}$. Hence, $\Pi(K^i)$ is an instance over $\{R\}$. Every reduction uses a different definition of Π, and in each reduction we prove that Π has the following key properties.

1. Π is injective over the facts of I^i; that is, for all facts f^i and g^i of I^i, if $\Pi(f^i) = \Pi(g^i)$ then $f^i = g^i$. It thus follows that Π is injective on the subinstances of I^i; that is, for all instances $K^i, L^i \subseteq I^i$, if $\Pi(K^i) = \Pi(L^i)$ then $K^i = L^i$.
2. Π preserves consistency and inconsistency; that is, for every instance K^i over $\mathcal{R}^i$ it holds that K^i satisfies Δ^i if and only if $\Pi(K^i)$ satisfies Δ.

With Properties 1 and 2 shown, the definition of the input for globally-optimal repair checking over $\mathbf{S}$ is straightforward:

- $I \stackrel{\text{def}}{=} \Pi(I^i)$.
- $\succ \stackrel{\text{def}}{=} \{(\Pi(f^i), \Pi(g^i)) \mid f^i, g^i \in I^i \ \wedge \ f^i \succ^i g^i\}$.
- $J \stackrel{\text{def}}{=} \Pi(J^i)$.

The construction is correct due to the following.

1. A repair $K^i \subseteq I^i$ is a global improvement of J^i for $(I^i, \succ^i)$ if and only if $\Pi(K^i)$ is a global improvement of J for $(I, \succ)$.
2. J^i is a globally-optimal repair of $(I^i, \succ^i)$ if and only if J is a globally-optimal repair of $(I, \succ)$.

In summary, for each reduction it suffices to define Π and prove that the two key properties are satisfied.

5.2 Case Branching

In this section, we fix a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ that violates the condition of Theorem 3.1 (that is, Δ is equivalent to neither a single fd nor two keys). We assume that $\mathcal{R}$ consists of a single relation symbol R. We will describe the different cases that our proof of hardness considers. We begin with the first case.

Case 1: Three or more keys. In this case, $\Delta = \{A_1 \to [\![R]\!], \ldots, A_k \to [\![R]\!]\}$ for $k \geq 3$, and $A_i \not\subseteq A_j$ for all $i \neq j$ (otherwise one of the fds can be removed). Here we show a reduction from globally-optimal repair checking for the schema $\mathbf{S}^1$ of Example 3.4.

For the remaining cases, we need some notation. Let $A \subseteq [\![R]\!]$ be a set of indices. We say that A is a *nontrivial determiner* if $A \subsetneq [\![R.A^{\Delta}]\!]$, and a *non-redundant determiner* if there is no set $B \subsetneq A$ such that $([\![R.A^{\Delta}]\!] \setminus A) \subseteq [\![R.B^{\Delta}]\!]$. In words, A is a non-redundant determiner if the set of attributes not in A that A determines is not already determined by any proper subset of A. Observe that a non-redundant determiner is necessarily nontrivial, but a nontrivial determiner is not necessarily non-redundant. We say that A is a *minimal determiner* if A is a nontrivial determiner and A does not strictly contain any nontrivial determiner. Observe that a minimal determiner is also non-redundant, but a non-redundant determiner is not necessarily minimal.

Suppose that $\mathbf{S}$ is not in Case 1 (in addition to violating the conditions of Theorem 3.1). We fix a minimal determiner $A \subseteq [\![R]\!]$, such that A is not a key. Note that such A exists since we assume that Δ is not equivalent to any set of key constraints. Observe that A may be the empty set. Since Δ is not equivalent to any single fd, there is at least one non-redundant determiner that is different from A; we select such a non-redundant determiner B that is minimal w.r.t. set containment. Observe that B can be a key, and B may contain A.

We will use the following notation:

- $A^+ \stackrel{\text{def}}{=} [\![R.A^{\Delta}]\!]$ and $\hat{A} \stackrel{\text{def}}{=} A^+ \setminus A$
- $B^+ \stackrel{\text{def}}{=} [\![R.B^{\Delta}]\!]$ and $\hat{B} \stackrel{\text{def}}{=} B^+ \setminus B$

Cases 2–7: Not all keys. The cases we consider here are the following.

- **Case 2:** $A^+ = B^+$
- **Case 3:** $B^+ \not\subseteq A^+$, $A \cap \hat{B} \neq \emptyset$ and $\hat{A} \cap B \neq \emptyset$
- **Case 4:** $B^+ \not\subseteq A^+$, $A \cap \hat{B} \neq \emptyset$ and $\hat{A} \cap B = \emptyset$
- **Case 5:** $B^+ \not\subseteq A^+$, $A \cap \hat{B} = \emptyset$, and $\hat{B} \subseteq \hat{A}$
- **Case 6:** $B^+ \not\subseteq A^+$, $A \cap \hat{B} = \emptyset$, and $\hat{B} \not\subseteq \hat{A}$
- **Case 7:** $A^+ \not\subseteq B^+$

Note that Cases 2–6 cover all the subcases of $B^+ \not\subseteq A^+$. Hence, together with Cases 1 and 7 we cover all the possible cases. In Cases 2–6 we show reductions from globally-optimal repair checking for the schemas $\mathbf{S}^i$ of Example 3.4 for $i = 2, \ldots, 6$, respectively. For Case 7 we show symmetry to the case of $B^+ \not\subseteq A^+$. Observe that some argument is required for this symmetry, since A and B are not defined in a symmetric manner.

5.3 End-to-End Case

In this section, we illustrate our proof strategy by giving the complete proof for one of the cases above, namely Case 1. We begin by showing coNP-hardness for the schema $\mathbf{S}^1$. Recall that $\mathbf{S}^1 = (\mathcal{R}^1, \Delta^1)$, where $\mathcal{R}^1$ consists of a single ternary relation symbol R^1, and $\Delta^1 = \{\{1,2\} \to 3, \{1,3\} \to 2, \{2,3\} \to 1\}$.

LEMMA 5.2. *The problem of globally-optimal repair checking is coNP-hard for the schema* $\mathbf{S}^1$.

PROOF. We will show a reduction from the undirected Hamiltonian Cycle problem, which is the following. Given an undirected graph $G = (V, E)$ with $V = \{v_0, \ldots, v_{n-1}\}$, where the v_i's are distinct, determine whether there is a permutation π over the set $\{0, \ldots, n-1\}$ such that there is an edge between $v_{\pi(i)}$ and $v_{\pi(i+1)}$ for all $i = 0, \ldots, n-1$, where addition (i.e., $+1$) is taken modulo n. So, let $G = (V, E)$ be a given graph with $V = \{v_0, \ldots, v_{n-1}\}$. We will construct inputs $(I, \succ)$ and J for globally-optimal repair checking. Our construction is illustrated in Figure 5 for the special case

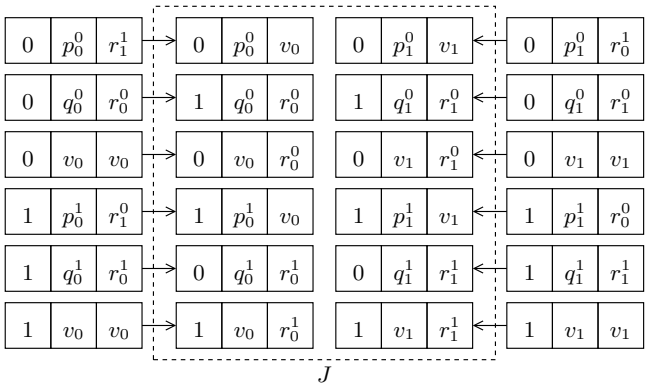

Figure 5: Illustration of the reduction from Hamiltonian Cycle to globally-optimal repair checking for $\mathbf{S}^1$, when $G = (V, E)$ where $V = \{v_0, v_1\}$ and E consists of the edge $\{v_0, v_1\}$

where G consists of two nodes v_0 and v_1 that are connected by an edge. The facts are represented as tuples without mentioning the relation symbol R^1.

The instance I has the following facts for every index $i \in \{0, \ldots, n-1\}$ and for every node $v_j \in V$: $R^1(i, p_j^i, v_j)$, $R^1(i-1, q_j^i, r_j^i)$, $R^1(i, v_j, r_j^i)$, $R^1(i, q_j^i, r_j^i)$, and $R^1(i, v_j, v_j)$. As before, throughout this proof, the sum $i+1$ is interpreted modulo n (i.e., it refers to the number $(i+1) \mod n$), and similarly for the difference $i-1$. Each symbol p_j^i, q_j^i and r_j^i is assumed to be a fresh constant.

In addition, I has the fact $R^1(i, p_j^i, r_k^{i+1})$ for every index $i \in \{0, \ldots, n-1\}$ and edge $\{v_j, v_k\} \in E$.

The priority $\succ$ is defined as follows for every index i, node v_j and edge $\{v_j, v_k\}$.

- $R^1(i, p_j^i, r_k^{i+1}) \succ R^1(i, p_j^i, v_j)$
- $R^1(i, q_j^i, r_j^i) \succ R^1(i-1, q_j^i, r_j^i)$
- $R^1(i, v_j, v_j) \succ R^1(i, v_j, r_j^i)$

Finally, the instance J consists of the following facts for every index $i \in \{0, \ldots, n-1\}$ and node $v_j \in V$:

- $R^1(i, p_j^i, v_j)$
- $R^1(i-1, q_j^i, r_j^i)$
- $R^1(i, v_j, r_j^i)$

The reader can verify that the input we have defined is legal; that is, $\succ$ is acyclic and gives preferences only between conflicting facts, and J is consistent. It is now left to prove that there is a global improvement J' of J if and only if G has a Hamiltonian cycle.

The "if" direction. Suppose that G has a Hamiltonian cycle π (which is a permutation over $\{0, \ldots, n-1\}$). We construct from π a global improvement J' by starting with J, and replacing:

- every $R^1(i, p_j^i, v_j)$ with $R^1(i, p_j^i, r_k^{i+1})$ where $j = \pi(i)$ and $k = \pi(i+1)$;
- every $R^1(i-1, q_j^i, r_j^i)$ with $R^1(i, q_j^i, r_j^i)$ where $j = \pi(i)$;
- every $R^1(i, v_j, r_j^i)$ with $R^1(i, v_j, v_j)$ where $j = \pi(i)$.

The construction is such that every fact in $J \setminus J'$ is improved by some fact in $J' \setminus J$. It remains to prove that J' is consistent.

We have already claimed that J is consistent. The reader can easily verify that the facts in $J' \setminus J$ are consistent among themselves. So, it remains to show that every fact $f \in J \setminus J'$ is consistent with every fact $f' \in J' \setminus J$. We do so via a case by case analysis.

Suppose first that $f = R^1(i, p_j^i, v_j)$. If f' agrees with f on the attributes 1 and 2, then f' is necessarily the fact $R^1(i, p_j^i, r_k^{i+1})$ that replaced f (hence, a contradiction). If f' agrees with f on the attributes 1 and 3, then f' is necessarily $R^1(i, v_j, v_j)$ where $j = \pi(i)$; but in that case, f has been replaced with $R^1(i, p_j^i, r_k^{i+1})$. Finally, it is clear that f does not agree with f' on the second and third attributes.

Suppose now that $f = R^1(i-1, q_j^i, r_j^i)$. Then it is clear that f' does not agree with f on the attributes 1 and 2. If f' agrees with f on the attributes 1 and 3, then f' is necessarily of the form $R^1(i', p_{j'}^{i'}, r_{k'}^{i'+1})$ where $i' = i-1$, $j' = \pi(i')$ and $k' = \pi(i'+1)$; but this means that $j = k' = \pi(i)$, and then f has been removed in the construction of J'. Finally, if f' agrees with f on the attributes 2 and 3, then f' is necessarily $R^1(i, q_j^i, r_j^i)$ where $j = \pi(i)$, which replaced f.

Finally, suppose that $f = R^1(i, v_j, r_j^i)$. If f' agrees with f on attributes 1 and 2, then f' is necessarily the fact $R^1(i, v_j, v_j)$, which replaced f. If f' agrees with f on attributes 1 and 3, then f' is necessarily $R^1(i, q_j^i, r_j^i)$ where $j = \pi(i)$; but then $j = \pi(i)$ implies that f has been replaced in our construction. Finally, it is clear that f' cannot agree with f on the attributes 2 and 3.

We conclude that J' is indeed a global improvement of J, as claimed.

The "only if" direction. We now assume that J' is a global improvement of J, and we will construct a Hamiltonian cycle π in G.

Let $i \in \{0, \ldots, n-1\}$ be given. We will first prove that if J' contains a fact of the form $R^1(i, v_j, v_j)$, then it must also contain a fact of the form $R^1(i', v_{j'}, v_{j'})$ where $i' = i+1$ and $v_{j'}$ is a neighbor of v_j. So, suppose that J' contains $R^1(i, v_j, v_j)$. Then $J \setminus J'$ must contain the conflicting $R^1(i, p_j^i, v_j)$ (since J contains $R^1(i, p_j^i, v_j)$ by construction, and $R^1(i, p_j^i, v_j)$ cannot be in J' since it conflicts with $R^1(i, v_j, v_j)$, which is in J'). Consequently, $J' \setminus J$ must contain $R^1(i, p_j^i, r_{j'}^{i+1})$ for some neighbor $v_{j'}$ of v_j. Therefore, $J \setminus J'$ must contain the conflicting $R^1(i'-1, q_{j'}^{i'}, r_{j'}^{i'})$ where $i' = i+1$. Hence, $J' \setminus J$ must contain $R^1(i', q_{j'}^{i'}, r_{j'}^{i'})$. Thus, $J \setminus J'$ must contain the conflicting $R^1(i', v_{j'}, r_{j'}^{i'})$, implying that $J' \setminus J$ must contain $R^1(i', v_{j'}, v_{j'})$, as claimed.

The arguments of the previous paragraph also show that if $J' \setminus J$ is nonempty, then J' must include at least one $R^1(i, v_j, v_j)$. (Thus, for each of the three types of facts in $I \setminus J$, namely those of the form $R^1(i, q_j^i, r_j^i)$, $R^1(i, v_j, v_j)$, and $R^1(i, p_j^i, r_k^{i+1})$ where $\{v_j, v_k\} \in E$, we see that by starting at some point in the middle of the previous paragraph, we end up concluding at the end of the paragraph that J' must include at least one fact of the form $R^1(i, v_j, v_j)$.) Therefore, J' contains a fact $R^1(i, v_j, v_j)$ for every $i = 0, \ldots, n-1$. Moreover, due to the constraint $\{2, 3\} \to 1$ we get that every v_j occurs in $R^1(i, v_j, v_j)$ with at most one i. Therefore, since

we have n indices and n nodes, we get that for every index i there is a unique fact $R^1(i, v_{j_i}, v_{j_i})$ in J'. As explained above, there is an edge between v_{j_i} and $v_{j_{i+1}}$ for all i, and consequently, the permutation π defined by $\pi(i) = j_i$ for $i = 0, \ldots, n-1$ is a Hamiltonian cycle. □

Recall in the case we consider, $\mathbf{S} = (\mathcal{R}, \Delta)$, $\mathcal{R} = \{R\}$ and $\Delta = \{A_1 \to [\![R]\!], \ldots, A_k \to [\![R]\!]\}$ where $k \geq 3$, and for all $i \neq j$ we have $A_i \not\subseteq A_j$. We now define the function Π that maps facts over R^1 into facts over R. For clarity, we denote A_1 by $A_{1,2}$, we denote A_2 by $A_{2,3}$, and we denote A_3 by $A_{1,3}$.

For a fact $f = R^1(c_1, c_2, c_3)$, the fact $\Pi(f)$ is the fact $R(d_1, \ldots, d_k)$, where for all $i = 1, \ldots, k$ the value d_i is defined as in the following equation. Each of the first two lines in this equation is quantified universally over (i.e., repeated for) all series $\langle a, b, c\rangle$ among the series $\langle 1, 2, 3\rangle$, $\langle 1, 3, 2\rangle$ and $\langle 2, 3, 1\rangle$.

$$d_i = \begin{cases} \langle c_a, c_b\rangle & \text{if } i \in A_{\{a,b\}} \setminus (A_{\{a,c\}} \cup A_{\{b,c\}}); \\ c_b & \text{if } i \in (A_{\{a,b\}} \cap A_{\{b,c\}}) \setminus A_{a,c}; \\ \diamond & \text{if } i \in A_{\{1,2\}} \cap A_{\{1,3\}} \cap A_{\{2,3\}}; \\ \langle c_1, c_2, c_3\rangle & \text{otherwise.} \end{cases}$$

We need to prove that Π has the two key properties, and we prove so in the following two lemmas.

LEMMA 5.3. *Π is injective.*

PROOF. To prove that Π is injective, it suffices to prove that for $f = R^1(c_1, c_2, c_3)$, each c_i occurs in $\Pi(f)$ at least once, and in a position that depends only on i (hence, we can restore f from $\Pi(f)$). We will show that for c_1, and by symmetry we will conclude the same for c_2 and c_3. Since $A_{\{2,3\}}$ is a minimal key, there is at least one index $i \in A_{\{1,2\}}$ that is not in $A_{\{2,3\}}$. For such i, the value in $\Pi(f)$ is either $\langle c_1, c_2\rangle$ or c_1 (depending on whether or not i is in $A_{\{1,3\}}$). In any case, c_1 occurs in a specific position within the ith attribute, as claimed. □

LEMMA 5.4. *Π preserves consistency and inconsistency.*

PROOF. Let $f = R^1(c_1, c_2, c_3)$ and $f' = R^1(c'_1, c'_2, c'_3)$ be two facts. We will show that f and f' are consistent w.r.t. $\mathbf{S}^1$ if and only if $\Pi(f)$ and $\Pi(f')$ are consistent w.r.t. $\mathbf{S}$.

The "if" direction. Suppose that $\Pi(f)$ and $\Pi(f')$ are consistent w.r.t. $\mathbf{S}$. We will show that $\{f, f'\}$ satisfies the fd $\{1, 2\} \to 3$. By symmetry, we also cover the other two fds. Suppose that $c_1 = c'_1$ and $c_2 = c'_2$. We must show $c_3 = c'_3$. For that, it suffices to prove that $\Pi(f)$ and $\Pi(f')$ agree on $A_{\{1,2\}}$. But, from the definition of Π it follows none of the attributes in $A_{\{1,2\}}$ mentions c_3 and c'_3 in f and f', respectively. It thus follows that f and f' agree on $A_{\{1,2\}}$, since $c_1 = c'_1$ and $c_2 = c'_2$.

The "only if" direction. Suppose that f and f' are consistent w.r.t. $\mathbf{S}^1$. We need to show that $\Pi(f)$ and $\Pi(f')$ are consistent w.r.t. $\mathbf{S}$. Recall that Δ is equivalent to a set of key constraints. So, we assume that Δ is, in fact, a set of key constraints. Let $R : A \to [\![R]\!]$ be a key constraint in Δ, and suppose that $\Pi(f)$ and $\Pi(f')$ agree on A. We need to show that $\Pi(f) = \Pi(f')$. If A contains attributes i that, in the definition of Π at least two of c_1, c_2 and c_3 are mentioned on the left hand sides, then f and f' must be the same due to the key constraints of $\mathbf{S}^1$. The only remaining case is where the left hand sides in the attributes i in A contain only c_b and $\diamond$ for some $b \in \{1, 2, 3\}$. This means that A is a subset of $A_{\{a,b\}} \cap A_{\{b,c\}}$ for corresponding a and c, and hence a strict subset of $A_{\{a,b\}}$ (and $A_{\{b,c\}}$), because $A_{\{a,b\}}$ and $A_{\{b,c\}}$ are different and minimal. However, this contradicts the fact that $A_{\{a,b\}}$ is minimal (since $A_{\{a,b\}}$ strictly contains the key A). □

This completes Case 1, where we have established the following result.

LEMMA 5.5. *Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema such that Δ is equivalent to a set of three or more keys, but not fewer. Then globally-optimal repair checking is coNP-complete over $\mathbf{S}$.*

6. DISTINGUISHING HARD SCHEMAS FROM TRACTABLE SCHEMAS

In this section, we investigate the problem of determining, given a schema $\mathbf{S}$, whether preferred repair checking is solvable in polynomial time or is coNP-complete; that is, whether $\mathbf{S}$ belongs to the tractable or the hard side of the dichotomy of Theorem 3.1. (Of course, this problem is of interest only under the assumption that P $\neq$ NP). We prove the following.

THEOREM 6.1. *Whether a schema $\mathbf{S}$ belongs to the tractable or the hard side of Theorem 3.1 can be decided in polynomial time in the size of $\mathbf{S}$.*

In the remainder of this section, we prove Theorem 6.1. Given Theorem 3.1, we can consider every relation symbol R separately, and test for every R whether $\Delta_{|R}$ is equivalent to a single fd or two keys.

LEMMA 6.2. *Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema that consists of a single relation symbol R. The following hold.*

1. *If Δ is equivalent to a nontrivial fd $A \to B$, then at least one fd in Δ has A as the left hand side.*
2. *If Δ is equivalent to a set $\{A_1 \to B_1, A_2 \to B_2\}$ of nontrivial fds where $A_1 \not\subseteq A_2$ and $A_2 \not\subseteq A_1$, then Δ has at least one fd with A_1 as the left hand side, and at least one fd with A_2 as the left hand side.*

We also need the following well-known theorem.

THEOREM 6.3. [13] *Given a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ and an fd δ over $\mathbf{S}$, it can be tested in polynomial time whether δ is implied by Δ (that is, Δ is equivalent to $\Delta \cup \{\delta\}$).*

With Lemma 6.2 and Theorem 6.3, we can now devise the following polynomial-time algorithm for deciding whether $\Delta_{|R}$ is equivalent to a single fd. For each left hand side A in $\Delta_{|R}$, use Theorem 6.3 to find the set B of all the indices b such that $A \to \{b\}$ is implied by Δ. Then, test whether every fd in $\Delta_{|R}$ is implied by $A \to B$. Part 1 of Lemma 6.2 implies that the test succeeds for some A, if and only if $\Delta_{|R}$ is equivalent to a single fd (with A as the left hand side).

To decide whether $\Delta_{|R}$ is equivalent to two key constraints, we consider two cases. In the case where one key is implied by the other (i.e., one key contains the other), we test whether $\Delta_{|R}$ is implied by a single key constraint similarly to the previous paragraph. Otherwise, we can use Part 2

of Lemma 6.2. Specifically, we consider every two left hand sides A_1 and A_2 in $\Delta_{|R}$, verify (using Theorem 6.3) that every index is functionally dependent on each of A_1 and A_2 (i.e., both are keys), and test whether every fd in $\Delta_{|R}$ is implied by $\{A_1 \to [\![R]\!], A_2 \to [\![R]\!]\}$.

7. CROSS-CONFLICT PRIORITIES

In this section, we relax the assumption that priorities are allowed only between conflicting facts, and consider the impact of this relaxation on the computational complexity of globally-optimal repair checking. Formally, the definition of a prioritizing instance $(I, \succ)$ is the same as the original definition in Section 2.3, except that $f \succ g$ can hold even if f and g are not in conflict. To distinguish from the ordinary case, we call such $(I, \succ)$ a *cross-conflict-prioritizing* instance, or *ccp-instance* for short. The remaining definitions, including global/Pareto improvement and globally/Pareto-optimal repairs, do not change.

Observe that the complexity of globally-optimal repair checking can only go higher (or remain unchanged), since now we allow inputs that were previously illegal. It is a straightforward observation that Pareto-optimal repair checking remains polynomial-time solvable for every schema. How does the relaxation affect globally-optimal repair checking? We answer this question in the remainder of this section. To present our main result, we need to introduce some notation.

7.1 Dichotomy for CCP-Instances

Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema, and let R be a relation symbol in $\mathcal{R}$. The fd $R : A \to B$ is a *constant-attribute constraint* if $A = \emptyset$. Note that such an fd states that all the tuples of R have the same values in each of the attributes of B. We say that Δ is a *primary-key assignment* if for every relation symbol $R \in \mathcal{R}$ the set $\Delta_{|R}$ (i.e., the restriction of Δ to the fds over R) is equivalent to a single key constraint, that is, a set of the form $\{A \to [\![R]\!]\}$. We say that Δ is a *constant-attribute assignment* if for every relation symbol $R \in \mathcal{R}$ the set $\Delta_{|R}$ is equivalent to a constant-attribute constraint, that is, a set of the form $\{\emptyset \to B\}$. Note that, as a special case, if Δ is empty then Δ is both a primary-key assignment and a constant-attribute assignment. Our main result for this section is the following.

THEOREM 7.1. *For a schema $\mathbf{S} = (\mathcal{R}, \Delta)$, the following hold.*

1. *If Δ is either a primary-key assignment or a constant-attribute assignment, then global optimality is solvable in polynomial time over ccp-instances.*
2. *Otherwise, global optimality is* coNP*-complete over ccp-instances.*

As an example, recall that under the schema $\mathbf{S}$ of Example 3.3, (ordinary) globally-optimal repair checking is solvable in polynomial time. For ccp-instances over $\mathbf{S}$, globally-optimal repair checking is coNP-complete, since $\Delta_{|R}$ is not equivalent to any single key constraint (nor is $\Delta_{|S}$) and is not equivalent to any constant-attribute asignment (nor is $\Delta_{|T}$). As another example, suppose that in Example 3.3 we replace Δ with $\{R : 1 \to 2, 3, S : \emptyset \to 1\}$. Then globally-optimal repair checking would still be coNP-complete, since Δ is neither a primary-key assignment nor a constant-attribute assignment. But if we replace Δ with $\{R : 1 \to 2, 3, S : \{1, 2\} \to 3\}$, then globally-optimal repair checking is solvable in polynomial time since Δ is now a primary-key assignment. (In particular, recall that we can always add a trivial constraint for the relation symbol T.)

In what follows, we discuss the proof of Theorem 7.1.

7.2 Algorithms for Tractable Schemas

We now prove the tractability part of Theorem 7.1 by presenting two polynomial-time algorithms for globally-optimal repair checking, one for the case where Δ is a primary-key assignment and one for the case where it is a constant-attribute assignment. In the case of a primary-key assignment, we again reduce globally-optimal repair checking to graph acyclicity, but the graph construction is different from that of Section 4.2. In the case of a constant-attribute assignment, we show that we can actually enumerate all the repairs in polynomial time (and in particular, there are only polynomially many repairs); once we do so, we can check whether any of the repairs improves upon the given J.

7.2.1 Primary-Key Assignment

We will now show that when Δ is a primary-key assignment for $\mathcal{R}$, then globally-optimal repair checking is testable in polynomial time over ccp-instances. Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be such a schema, and let $(I, \succ)$ and J be input for globally-optimal repair checking. We assume that J is a repair (i.e., J is a maximal consistent subinstance of I), since the problem is straightforward otherwise. The idea in the algorithm is as follows. To improve J, we need to add a fact $g_1 \in I \setminus J$ to J, but then g_1 conflicts with some fact $f_1 \in J$. So, we need to remove f_1 from J; but for that, we need to add to J a fact $g_2 \in I \setminus J$ such that $g_2 \succ f_1$. Then again, g_2 conflicts with some $f_2 \in J$, and so on. We succeed in this process (i.e., produce an improvement of J) if we close a cycle (that is, we add g_i or f_i that we already encountered in this process). Next, we formalize this idea.

We define the graph $G_{J,I\setminus J}$ over the facts of I, as follows. The graph $G_{J,I\setminus J}$ is a directed bipartite graph that has J on one side, and $I \setminus J$ on the other side. There is an edge $f \to g$ from $f \in J$ to $g \in I \setminus J$ if f conflicts with g, and there is an edge $g \to f$ if $g \succ f$.

EXAMPLE 7.2. Suppose that $\mathcal{R} = \{R\}$ where R is a binary relation symbol, and $\Delta = \{R : 1 \to 2\}$. Suppose that I is the instance such that

$$R^I = \{(0,1), (0,2), (0,c), (1,a), (1,b), (1,3)\}.$$

Moreover, suppose the following.

- $R(1,3) \succ R(0,2) \succ R(0,1)$.
- $R(0,c) \succ R(1,b) \succ R(1,c)$.

Finally, let J be the repair that consists of the facts $R(0,2)$ and $R(1,b)$. Figure 6 depicts the graph $G_{J,I\setminus J}$. The thickness of the weights can be ignored for now; those will be discussed later. □

LEMMA 7.3. *J has a global improvement if and only if $G_{J,I\setminus J}$ has a cycle.*

PROOF. We prove each direction separately.

The "if" direction:
We assume that $G_{J,I\setminus J}$ has a cycle. Then $G_{J,I\setminus J}$ has a simple cycle. Consider such a simple cycle $f_1 \to g_1 \to \cdots \to$

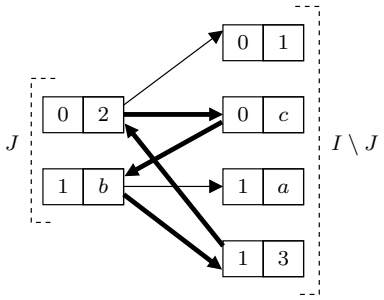

Figure 6: The graph $G_{J,I\setminus J}$ for Example 7.2

$g_k \to f_{k+1}$, where each f_i belongs to J, each g_i belongs to $I \setminus J$, and $f_{k+1} = f_1$. Define the following.

$$J' \stackrel{\text{def}}{=} (J \setminus \{f_1, \ldots, f_k\}) \cup \{g_1, \ldots, g_k\}$$

We claim that J' is a global improvement of J. To prove that, we first need to show that J' is consistent. Suppose, by way of contradiction, that J' has a Δ-conflict. Since J is consistent, such a conflict must be of the form $\{f_j, g_i\}$ or $\{g_i, g_j\}$ for $i \neq j$. Suppose that the conflict is $\{f_j, g_i\}$. From the construction of $G_{J,I\setminus J}$ we know that $\{f_i, g_i\}$ is a conflict; hence, $\{f_i, f_j\}$ is also a conflict (because all the fds are primary key constraints), which contradicts the fact that J is consistent. So now, suppose that the conflict is $\{g_i, g_j\}$. Since both $\{f_i, g_i\}$ and $\{f_j, g_j\}$ are conflicts, we get that $\{f_i, f_j\}$ is a conflict (again due to the fact that the fds are primary key constraints), and again we get a contradiction. We conclude that J' is consistent.

It remains to show that J' is a global improvement of J. For that, we need to show that for each f_i there is a g_j such that $g_j \succ f_i$. So, as g_j we select g_{i-1} if $i > 1$, and g_k if $i = 1$. From the constriction of $G_{J,I\setminus J}$ it follows that, indeed, we have $g_j \succ f_i$, as claimed.

The "only if" direction:
Now we assume that J has a global improvement J', and we need to show a cycle in $G_{J,I\setminus J}$. We do so by simply repeating the argument at the beginning of this section, and obtain a path $f_1 \to g_1 \to \cdots \to g_k \to f_{k+1}$ that is long enough to contain a cycle (since $G_{J,I\setminus J}$ is a finite graph). □

As a consequence, we get the following proposition.

PROPOSITION 7.4. *For a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ where Δ is a primary-key assignment, globally-optimal repair checking is solvable in polynomial time over ccp-instances.*

7.2.2 Constant-Attribute Assignment

We will now show that when Δ is a constant-attribute assignment for $\mathcal{R}$, then globally-optimal repair checking is testable in polynomial time. Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema such that Δ is a constant-attribute assignment for $\mathcal{R}$. Let $(I, \succ)$ and J be input for globally-optimal repair checking. Consider a relation symbol R of Δ. A *consistent partition* of R^I is a maximal subset of R^I that agrees on (i.e., has the same value in each attribute in) $[\![R.\emptyset^{\Delta}]\!]$. An easy observation is that a subinstance K of I is a repair of I if and only if K consists of one consistent partition from each R^I. Therefore, we can simply enumerate all such K in polynomial time and test (in polynomial time) whether any of them is a global improvement of J. Observe that the degree of the polynomial is bounded by the number of relations in $\mathcal{R}$. As a consequence, we get the following proposition.

PROPOSITION 7.5. *For a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ where Δ is a constant-attribute assignment, globally-optimal repair checking is solvable in polynomial time over ccp-instances.*

7.3 Proof Strategy for Hardness

The hardness part of Theorem 7.1 is proved in a pattern similar to that of the proof of the hardness part of Theorem 3.1. In particular, we prove coNP-hardness for a set of specific schemas, and then build reductions (by means of the function Π) from these specific schemas to the rest of the schemas. The specific schemas in this case are $\mathbf{S}^x = (\mathcal{R}^x, \Delta^x)$ for $x = \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}$, where:

- $\mathcal{R}^{\mathsf{a}}$ consists of two binary relation symbols R and S, and $\Delta^{\mathsf{a}} = \{R : 1 \to 2,\ S : \emptyset \to 1\}$;
- $\mathcal{R}^{\mathsf{b}}$ consists of a single relation symbol, which is ternary, and $\Delta^{\mathsf{b}} = \{1 \to 2\}$.
- $\mathcal{R}^{\mathsf{c}}$ consists of a single relation symbol, which is ternary, and $\Delta^{\mathsf{c}} = \{1 \to 2, \emptyset \to 3\}$.
- $\mathcal{R}^{\mathsf{d}}$ consists of a single relation symbol, which is binary, and $\Delta^{\mathsf{d}} = \{1 \to 2, 2 \to 1\}$.

7.4 Distinguishing Between the Cases

By repeating the arguments of Section 6, we get the following theorem, stating that we can test in polynomial time whether a schema belongs to one of the other side of the dichotomy of Theorem 7.1.

THEOREM 7.6. *Whether a schema $\mathbf{S}$ belongs to the tractable or the hard side of Theorem 7.1 can be decided in polynomial time in the size of $\mathbf{S}$.*

8. CONCLUDING REMARKS

Globally-optimal repair checking is the problem of deciding, given an inconsistent database I and a subinstance J of I, whether J is a globally-optimal repair of I. When the constraints are given by a set Σ of functional dependencies, we gave a complete characterization, in terms of Σ, of the computational complexity of the globally optimal repair-checking problem. In fact, we proved a dichotomy theorem, stating that that this problem is either solvable in polynomial time or is coNP-complete, and we gave a polynomial-time algorithm that decides, given Σ, which of these two cases holds. We believe that our work (including the tools we developed) will pave the road towards other complexity classifications in the study of preferred repairs. One important such direction is the computational complexity of the preferred consistent query-answering problem. Another interesting direction to pursue is to determine the number of globally-optimal repairs, and in particular, to characterize when precisely one such repair exists. The latter is an important problem because the existence of precisely one repair implies that the constraints and priorities define an unambiguous *cleaning* of inconsistencies.

Acknowledgments

We are grateful to Slawek Staworko for helpful discussions on this paper. Phokion Kolaitis is partially supported by NSF Grant IIS-1217869.

9. REFERENCES

[1] F. N. Afrati and P. G. Kolaitis. Repair checking in inconsistent databases: algorithms and complexity. In *ICDT*, pages 31–41, 2009.

[2] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *Proceedings of the TIPSTER Text Program: Phase III*, pages 23–30, Baltimore, Maryland, USA, 1998.

[3] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *PODS*, pages 68–79. ACM Press, 1999.

[4] L. E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[5] 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.

[6] R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Spanners: a formal framework for information extraction. In *PODS*, pages 37–48, 2013.

[7] R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Cleaning inconsistencies in information extraction via prioritized repairs. In *PODS*. ACM, 2014.

[8] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[9] G. Fontaine. Why is it hard to obtain a dichotomy for consistent query answering? In *LICS*, pages 550–559, 2013.

[10] B. Kimelfeld. A dichotomy in the complexity of deletion propagation with functional dependencies. In *PODS*, pages 191–202, 2012.

[11] P. G. Kolaitis and E. Pema. A dichotomy in the complexity of consistent query answering for queries with two atoms. *Inf. Process. Lett.*, 112(3):77–85, 2012.

[12] P. Koutris and D. Suciu. A dichotomy on the complexity of consistent query answering for atoms with simple keys. In *ICDT*, pages 165–176, 2014.

[13] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469, 1979.

[14] 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.

[15] B. ten Cate, G. Fontaine, and P. G. Kolaitis. On the data complexity of consistent query answering. In *ICDT*, pages 22–33, 2012.