Ann Math Artif Intell (2012) 64:209–246
DOI 10.1007/s10472-012-9288-8

Prioritized repairing and consistent query answering in relational databases

Sławek Staworko · Jan Chomicki · Jerzy Marcinkowski

Published online: 27 March 2012

Abstract A consistent query answer in an inconsistent database is an answer obtained in every (minimal) repair. The repairs are obtained by resolving all conflicts in all possible ways. Often, however, the user is able to provide a preference on how conflicts should be resolved. We investigate here the framework of *preferred consistent query answers*, in which user preferences are used to narrow down the set of repairs to a set of *preferred repairs*. We axiomatize desirable properties of preferred repairs. We present three different families of preferred repairs and study their mutual relationships. Finally, we investigate the complexity of preferred repairing and computing preferred consistent query answers.

Keywords Repairing · Consistent query answers · Preferences · Priorities

Mathematics Subject Classification (2010) 68P15

Research partially supported by NSF grants IIS-0119186 and IIS-0307434, Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the' Contrat de Projets Etat Region (CPER) 2007–2013, Enumeration project ANR-07-blanc, and Polish Ministry of Science and Higher Education research project N N206 371339.

Part of this research was done when S. Staworko was a PhD student at the University at Buffalo.

S. Staworko (✉)
Mostrare project,
INRIA & LIFL (CNRS UMR8022), Villeneuve d'Ascq, France
e-mail: slawomir.staworko@inria.fr

J. Chomicki
Computer Science and Engineering,
University at Buffalo, Buffalo, NY, USA
e-mail: chomicki@buffalo.edu

J. Marcinkowski
Institute of Informatics,
Wrocław University, Wrocław, Poland
e-mail: jma@cs.uni.wroc.pl

1 Introduction

In many novel database applications, violations of integrity constraints cannot be avoided. A typical example is integration of two consistent data sources that contribute conflicting information. Inconsistencies also often occur in the context of long-running operations where transaction mechanisms are not employed. Finally, integrity enforcement may be disabled because of efficiency considerations. Integrity constraints, however, capture important semantic properties of the stored data. These properties directly influence the way a user formulates a query. Evaluation of the query over an inconsistent database may yield answers that are meaningless or misleading.

The framework of *repairs* and *consistent query answers* [4] has been proposed to offset the impact of inconsistencies on the accuracy of query answers. A *repair* is a consistent database minimally different from the given one, and a *consistent answer* to a query is an answer present in *every* repair. This approach does not physically remove any facts from the database. The framework of [4] has served as a foundation for most of the subsequent work in the area of querying inconsistent databases (for the surveys of the area see [6–8, 12, 14], other works include [32, 33]).

Recently, the problem of database repairing has received an enlivened interest [2, 17]. Essentially, the goal is to construct a repair of a possibly inconsistent instance by resolving every conflict present in the given instance. In the case of denial constraints, the class of constraints we consider in this paper, a conflict is simply a set of facts that are present in the given instance that together violate a constraint. A resolution of a conflict is the deletion of one of the facts creating the conflict. Typically, there exists more than one repair and a repairing algorithm needs to make some nondeterministic choices when repairing the database instance. It is desirable for the algorithm to be *sound* i.e., always producing a repair, that is, an instance which is not only consistent but also minimally different from the given one. It is even more desirable for the algorithm to be *complete* i.e., allowing to produce every repair, with an appropriate sequence of choices [28].

Example 1 Consider the schema consisting of two relations

$$Emp(Name, Salary, Dept) \quad \text{and} \quad Mgr(Name, Salary, Dept),$$

and the set of constraints F_0 consisting of

$$Emp : Name \to Name\ Salary\ Dept,$$
$$\forall x, y, z, x', y'. \neg\left[Emp(x, y, z) \wedge Mgr\left(x', y', z\right) \wedge y > y'\right].$$

The first constraint is a key dependency requiring the employee information to be associated with her name. The second constraint is a denial constraint requiring that no employee of a department earns more than the manager of the department.

Now, consider the inconsistent database instance

$$I_0 = \{Emp(John, \$40k, IT), Emp(John, \$50k, IT),$$
$$Emp(John, \$80k, IT), Mgr(Mary, \$70k, IT)\}.$$

This instance contains three conflicts w.r.t. the functional dependency and one conflict w.r.t. the denial constraint. I_0 has three repairs w.r.t. F_0:

$$I_1' = \{Emp(John, \$80k, IT)\},$$
$$I_2' = \{Emp(John, \$50k, IT), Mgr(Mary, \$70k, IT)\},$$
$$I_3' = \{Emp(John, \$40k, IT), Mgr(Mary, \$70k, IT)\}.$$

Consider the query $Q_0 = \exists x, y.\ Emp(John, x, y) \wedge x > \$60k$ asking whether *John* earns more than \$60*k*. The answer to Q_1 in the database instance I_0 is **true**. However, **true** is not a consistent answer to Q_1 because of the repairs I_2' and I_3'. □

One of the drawbacks of the framework of consistent query answers is that it considers all possible ways to resolve the existing conflicts. The user, however, may have a preference on what resolutions to consider. Typical information used to express the preference includes:

– the timestamp of creation/last modification of the fact; the conflicts can be resolved by removing from consideration old, outdated facts,
– the source of the fact (in data integration setting); the user can consider the data from one source more reliable than the data from another source,
– the data values stored in the conflicting facts.

To improve the quality of consistent answers we propose extending the framework of repairs and consistent query answers with the preference information. We use the preference information to define a set of *preferred* repairs (a subset of all repairs). Query answers obtained in every preferred repair are called *preferred consistent query answers*. For instance, in the previous example if the database contains an employee who earns more than her manager, then we might prefer to remove the information about the employee rather than the information about the manager of the department. Then the preferred repairs are I_2' and I_3', and consequently, **false** is the preferred consistent answer to Q_0.

We observe, however, that there may be more that one way to select the preferred repairs based on the user preference; especially, when a resolution of one conflict affects the way in which another conflict can be resolved.

Example 2 We take the schema consisting of one relation name

$$Mgr(Name, Salary, Dept)$$

with two functional dependencies

$$Mgr : Name \rightarrow Salary\ Dept \quad \text{and} \quad Mgr : Dept \rightarrow Name\ Salary.$$

Consider the following inconsistent instance

$$I_1 = \{Mgr(Bob, \$70k, RD), Mgr(Mary, \$40k, IT), Mgr(Ken, \$60k, IT),$$
$$Mgr(Bob, \$60k, AD), Mgr(Mary, \$50k, PR), Mgr(Ken, \$50k, PR)\}$$

This instance contains five conflicts:

1. $Mgr(Bob, \$70k, RD)$ and $Mgr(Bob, \$60k, AD)$.
2. $Mgr(Mary, \$40k, IT)$ and $Mgr(Mary, \$50k, PR)$,

3. $Mgr(Ken, \$60k, IT)$ and $Mgr(Ken, \$50k, PR)$,
4. $Mgr(Mary, \$40k, IT)$ and $Mgr(Ken, \$60k, IT)$,
5. $Mgr(Mary, \$50k, PR)$ and $Mgr(Ken, \$50k, PR)$,

These conflicts may arise from changes that are not yet fully propagated. For instance, *Bob* may have been moved to manage *R&D* department while previously being the manager of AD, or *Bob* may have been moved from AD department to RD department. Similarly, *Mary* may have been promoted to manage PR whose previous manager was moved to manage IT, or conversely, *John* may have been moved to manage IT, while *Mary* was moved from IT to PR.

The set of repairs of I_1 consists of four instances:

$$I'_1 = \{Mgr(Bob, \$70k, RD), Mgr(Mary, \$50k, PR), Mgr(Ken, \$60k, IT)\},$$
$$I'_2 = \{Mgr(Bob, \$70k, RD), Mgr(Mary, \$40k, IT), Mgr(Ken, \$50k, PR)\},$$
$$I'_3 = \{Mgr(Bob, \$60k, AD), Mgr(Mary, \$40k, IT), Mgr(Ken, \$50k, PR)\},$$
$$I'_4 = \{Mgr(Bob, \$60k, AD), Mgr(Mary, \$50k, PR), Mgr(Ken, \$60k, IT)\}.$$

Suppose that the user prefers to resolve a conflict created by two facts referring to the same person by *removing the tuples with the smaller salary*. This preference expresses the belief that if a manager is being reassigned, her salary is not decreased. It applies to the first conflict: the fact $Mgr(Bob, \$70k, RD)$ is preferred over $Mgr(Bob, \$60k, AD)$. Similarly, the preference applies to the second and the third conflict. It does not apply to the last two conflicts as each of them involves facts referring to different persons.

The preference information on resolutions of the first conflict allows us to eliminate the last two repairs I'_3 and I'_4. Similarly, by applying the preference to the conflicts 2 and 3 we may also eliminate the repair I'_2. This leaves us with only one preferred repair I'_1.

We observe that while the preference applies to conflicts 1, 2, and 3, it does not apply to conflicts 4 and 5 because conflicts 4 and 5 involve facts about different persons. However, the preferential resolution of conflicts 2 and 3 *implicitly* resolves the conflicts 4 and 5, which may not be desirable. Consequently, one may find the reasons for eliminating I'_2 insufficient. □

In this paper we consider three different families of *preferred repairs*. The families are based on various notions of compliance of a repair with the user preference. The first two notions, global and Pareto optimality, check if the compliance of a repair I' can be improved by replacing a subset of facts $X \subseteq I'$ with a *more preferred* subset of facts $Y \subseteq I \setminus I'$. These notions differ in the way they lift preference on facts to preferences on sets of facts.

Global optimality requires that for every element in X there is a more preferred element in Y. This approach is inspired by the work on preferential reasoning [23] and corresponds to the first way of selecting preferred repairs in the previous example. For instance, I'_2 is not globally optimal because we can replace $X = \{Mgr(Mary, \$40k, IT), Mgr(Ken, \$50k, PR)\}$ with a more preferred

$Y = \{Mgr(Mary, \$50k, PR), Mgr(Ken, \$60k, IT)\}$, obtaining the only globally-optimal instance I_1'.

Pareto optimality requires a stronger support from the preference to conclude that the compliance of a repair with the preference can be improved: every element of Y needs to be preferred over every element of X. This approach is inspired by the construction of the Pareto-optimal set of vectors [24] and it corresponds to the second way of selecting preferred repairs in the previous examples. For instance, I_3' is not Pareto optimal because we can replace $X = \{Mgr(Bob, \$60k, AD)\}$ with $Y = \{Mgr(Bob, \$70, RD)\}$. We remark that for this notion of optimality the compliance of I_2' with the preference cannot be further improved, thus I_2' is Pareto optimal.

The third notion of *completion optimality* uses a different approach to verify an optimal compliance of a repair with the preference. It views the preference only as a step towards a *total* preference i.e., preference that specifies the preferred resolution of every conflict, which yields exactly one repair. A repair is completion optimal if the preference can be extended to a total preference that yields the given repair. In the previous example completion optimality coincides with global optimality. The instance I_1' is completion optimal because we can add an appropriate preference for conflicts 4 and 5.

For every family of preferred repairs we present a repairing algorithm. Each of them is *sound* i.e., it produces a repair belonging to the corresponding family of preferred repairs, and *complete* i.e., every repair from the family of preferred repairs can be constructed using the corresponding repairing algorithm. For the family of globally-optimal repairs and the family of Pareto-optimal repairs we define two pre-orders on repairs whose maximal elements are exactly the globally-optimal repairs and Pareto-optimal repairs respectively. It is an open question whether such an order can be defined for completion-optimal repairs.

We also adapt two basic decision problems: *repair checking* [2, 14] and *consistent query answering* [4] to obtain preferred repair checking and preferred consistent query answering. Basically, *preferred repair checking* is finding if a given database instance is a preferred repair, and *preferred consistent query answering* is finding if an answer to a query is obtained in every preferred repair.

Recall from [14] that the class of denial constraints lies on the tractability frontier of consistent query answering. On the one hand for the class of denial constraints repair checking and computing consistent answers to quantifier-free ground queries is in PTIME. On the other hand, computing consistent answers to conjunctive queries i.e., conjunctions of positive literals with existential quantifiers, becomes coNP-complete even in the presence of one functional dependency i.e., a simple denial constraint. It seems natural that this tractability frontier should shift after incorporating a nontrivial component into the inputs of the definitions of the decision problems and the interesting question is how much.

We show that using the notion of global optimality leads to intractability of both preferred consistent query answering, which becomes Π_2^p-complete, and preferred repair checking, which becomes coNP-complete. The complexity is reduced if we use the notion of Pareto optimality: the preferred consistent query answering becomes coNP-complete and preferred repair checking is in LOGSPACE. Using completion-optimal repairs also reduces the complexity: preferred repair checking is in PTIME and preferred consistent query answering becomes coNP-complete. It is an open question whether in this case the preferred repair checking is PTIME-complete or in LOGSPACE. Finally, we identify a tractable case of quantifier-free ground queries and one FD per relation, for which preferred consistent query answering is in PTIME for every of the aforementioned families of preferred repairs.

The contributions of this paper are:

- A formal framework of families of preferred repairs and preferred consistent query answers for relational databases.
- A list of desirable properties of families of preferred repairs.
- Three different families of preferred repairs based on different notions of optimal compliance with the user preference.
- Repairing algorithm for every family of preferred repairs. The algorithms are both sound and complete.
- A thorough analysis of computational implications of preferences in the context of repairing and consistent query answers.

The presented work is an extension of [29]. The current paper extends the framework of preferred consistent query answers to denial constraints (instead of functional dependencies), provides detailed proofs of all claims, and presents sound and complete repairing algorithms for every considered family of preferred repairs (instead of just the repairing algorithm for the family of completion-optimal repairs only). Additionally, we further broaden the analysis of computational complexity by identifying a family of preferred repairs for which preferred repair checking is in LOGSPACE, offering a possibility of parallel implementation for this decision problem.

The paper is organized as follows. In Section 2 we recall basic notions of relational databases and the framework of repairs and consistent query answers. In Section 3 we extend this framework with preferences on conflict resolution. In Sections 4, 5, and 6 we present the families of globally-, Pareto-, and completion-optimal repairs respectively. We investigate their properties and mutual relationships, and analyze the computational implications of their semantics. In Section 7 we present a tractable case of preferred consistent query answering. Section 8 contains a discussion of related work. Finally, in Section 9 we summarize our results and outline directions for future work.

2 Preliminaries

In this section we recall the basic notions of relational databases [1] and the framework of consistent query answers [4]. A database *schema* $\mathcal{S}$ is a set of relation names of fixed arity (greater than 0) whose attributes are drawn from an infinite set of names U. Every element of U is typed but for simplicity we consider only two disjoint infinite domains: Q (rationals) and D (uninterpreted constants). We assume that two constants are equal if and only if they have the same name, and we allow the standard *built-in* relation symbols $=$ and $\neq$ over D. We also allow the built-in relation symbols $=$, $\neq$, $<$, $\leq$, $>$, and $\geq$ with their natural interpretation over Q. We use these symbols together with the vocabulary $\mathcal{S}$ of relational names to build a first-order language $\mathcal{L}$. An $\mathcal{L}$-formula is:

- *closed* (or a *sentence*) if it has no free variables,
- *ground* if it has no variables whatsoever,
- *quantifier-free* if it has no quantifiers,
- *atomic* if it has no quantifiers and no Boolean connectives.

Finally, a *fact* is an atomic ground $\mathcal{L}$-formula.

Database *instances* are finite, first-order structures over the schema. Often, we find it more convenient to view an instance I as the finite set of all facts satisfied by the instance i.e., $\{R(t) \mid R \in \mathcal{S}, I \models R(t)\}$. In this paper we use the standard notion of *satisfaction* (or *entailment*) of an $\mathcal{L}$-formula ϕ in a database instance I, in symbols $I \models \phi$. An $\mathcal{L}$-formula is *valid* iff it is satisfied in every database instance I. Notice that the validity of a quantifier-free ground formula using only built-in predicates is decided in a straightforward fashion.

In the sequel, we denote tuples of variables by $\bar{x}, \bar{y}, \ldots$, tuples of constants by $t, s, \ldots$, quantifier-free formulas using only built-in predicates by φ, instances by $I, J, \ldots$, relation names by $R, P, \ldots$, and attribute names by $A, B, C, \ldots$. The symbols $X, Y, \ldots$ are used to denote finite sets of attribute names. We also use $X, Y, \ldots$ to denote finite sets of facts, and it will always be clear from the context which usage is employed.

2.1 Integrity constraints

In general, an integrity constraint is a closed $\mathcal{L}$-formula. In this paper we consider the class of *denial constraints*, $\mathcal{L}$-sentences of the form

$$\forall \bar{x}. \neg \left[R_1(\bar{x}_1) \wedge \ldots \wedge R_n(\bar{x}_n) \wedge \varphi(\bar{x}) \right],$$

where $\varphi(\bar{x})$ is a quantifier-free formula referring to built-in relation names only and $\bar{x}_1 \cup \ldots \cup \bar{x}_n = \bar{x}$. We also make a natural assumption that $n > 0$.

The class of denial constraints contains *functional dependencies* (FDs) commonly formulated as $R : X \rightarrow Y$, where X and Y are sets of attributes of R. An FD $R : X \rightarrow Y$ is expressed by the following denial constraint

$$\forall \bar{x}, \bar{y}_1, \bar{y}_2, \bar{z}, \bar{z}'. \neg \left[R(\bar{x}, \bar{y}_1, \bar{z}) \wedge R\left(\bar{x}, \bar{y}_2, \bar{z}'\right) \wedge \neg (\bar{y}_1 = \bar{y}_2) \right],$$

where $\bar{x}$ is the vector of variables corresponding to the attributes X, and $\bar{y}_1$ and $\bar{y}_2$ are two vectors of variables corresponding to the attributes Y. A *key dependency* is a functional dependency $R : X \rightarrow Y$, where Y comprises all attributes of R. If the relation name is known from context, for clarity we omit it in our notation i.e., we write $X \rightarrow Y$ instead of $R : X \rightarrow Y$. Database consistency is defined in the standard way.

Definition 1 Given a database instance I and a set of integrity constraints F, I is *consistent* with F if $I \models F$ in the standard model-theoretic sense; otherwise I is *inconsistent*.

We observe that an empty instance satisfies any set of denial constraints. This conforms to the behavior of typical SQL database management systems: an empty database satisfies any set of constraints expressed in SQL. Also, note that denial constraints can be represented using standard SQL assertions. We remark, however, that the converse is not necessarily the case.

2.2 Queries

In this paper we deal only with *closed* queries i.e., closed $\mathcal{L}$-formulas. The query answers are Boolean: **true** or **false**. A query is *atomic* (*quantifier-free*) if the $\mathcal{L}$-formula is atomic (quantifier-free respectively). A *conjunctive* query is an existentially quantified conjunction of atomic $\mathcal{L}$-formulas.

Definition 2 Given an instance I and a closed query Q, **true** is the answer to Q in I if $I \models Q$; otherwise the answer to Q in I is **false**.

2.3 Repairing

In the original framework, when repairing a database two operations are considered: inserting a fact and deleting a fact. In the presence of denial constraints inserting facts cannot resolve inconsistencies, and thus the repairs of the original instance are obtained by deleting facts only i.e., the repairs are subsets of the original instance.

Definition 3 (Repair) Given an instance I and a set of denial constraints F, an instance I' is a *repair* of I w.r.t. F if and only if I' is a maximal subset of I that is consistent with F. By $Rep(I, F)$ we denote the set of all repairs of I w.r.t. F.

To identify the facts whose mutual presence causes inconsistency we use the notion of a conflict.

Definition 4 (Conflict) Given a instance I and a set of denial constraints F, a set of facts $\{R_1(t_1), \ldots, R_n(t_n)\} \subseteq I$ is a *conflict* in I w.r.t. F if for some denial constraint in F of the form

$$\forall \bar{x}.\ \neg[R_1(\bar{x}_1) \wedge \ldots \wedge R_n(\bar{x}_n) \wedge \varphi(\bar{x})]$$

there exists a substitution ρ of variables $\bar{x}$ such that $\varphi(\rho(\bar{x}))$ is valid and $\rho(\bar{x}_i) = t_i$ for every $i \in \{1, \ldots, n\}$.

We recall the notion of a conflict hypergraph that allows to visualize all the conflicts present in the instance [5, 13]. We recall that a hypergraph is a generalization of an undirected graph by allowing more than two nodes to be connected by a hyperedge. Formally, a *hypergraph* is a pair consisting of a set of nodes and a set of hyperedges, where a *hyperedge* is a subset of the node set. Given a hypergraph $\mathcal{G}$ we denote its set of nodes by $V(\mathcal{G})$, and its set of hyperedges by $E(\mathcal{G})$.

Definition 5 (Conflict hypergraph) Given a set of integrity constraints F and a database instance I, the *conflict hypergraph* $\mathcal{G}(I, F)$ of I w.r.t. F is a hypergraph whose set of nodes is I and set of hyperedges consists of all conflicts in I w.r.t. F.

The size of the hypergraph is he sum of the size of the node set and the cardinalities of all hyperedges. We observe that assuming F to be fixed, the maximum cardinality of every hyperedge in a conflict hypergraph is bounded from above by a constant. Consequently, the size of a conflict hypergraph $\mathcal{G}(I, F)$ is polynomial in the size of the instance I.

Two nodes are *neighboring* (or are *neighbors*) in a hypergraph if there exists a hyperedge containing both nodes. The *neighborhood* of a node $v \in V(\mathcal{G})$ in a hypergraph $\mathcal{G}$ is

$$n_{\mathcal{G}}(v) = \left\{v' \in V(\mathcal{G}) \mid \exists e \in E(\mathcal{G}).\ \left\{v, v'\right\} \subseteq e\right\}.$$

A hyperedge connecting exactly two nodes is called simply an edge and a hypergraph having only edges is called a graph. Similarly, we define the conflict graph. The conflict graph for the instance in Example 1 is in Fig. 1. The conflict hypergraph is also a compact representation of all repairs as we recall the following fact.

Proposition 1 [5, 13] *A* maximal independent set *of $\mathcal{G}(I, F)$ is any maximal set of vertices that contains no hyperedge. Any maximal independent set is a repair of I w.r.t. F and vice versa.*

We recall that for only one key dependency (per relation name), the conflict graph is a union of pairwise disjoint cliques and every repair consists of exactly one element from each clique [5]. To generalize this observation to FDs we assume only one relation name R and one functional dependency $R : X \rightarrow Y$. Now, given an instance I, an X*-cluster* is the set of all facts (of R) in I that have the same attribute value in X, and similarly, an (X, Y)*-cluster* is the set of all facts (of R) in I that have the same attribute value in X and Y. Clearly, an X-cluster is a union of all (X, Y)-clusters with the same attribute value in X. We recall that every repair contains exactly one (X, Y)-cluster from each X-cluster. We also remark that conflicts are present only inside an X-cluster and two facts from the same X-cluster form a conflict if and only if they belong to different (X, Y)-clusters.

Example 3 Consider the database schema consisting of exactly one relation name $R(A, B, C)$ and the FD $R : A \rightarrow B$. Take the following database instance

$$I_2 = \{R(1, 1, 1), R(1, 1, 2), R(1, 1, 3), R(1, 2, 1), R(1, 2, 2),$$
$$R(2, 1, 1), R(2, 1, 2), R(2, 1, 3), R(2, 2, 1)\}.$$

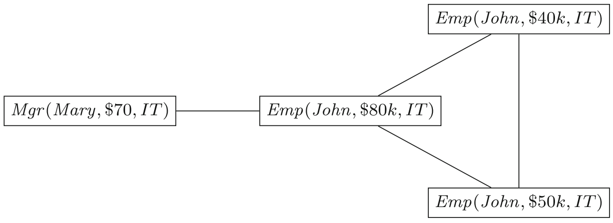

Fig. 1 The conflict graph $\mathcal{G}(I_0, F_0)$

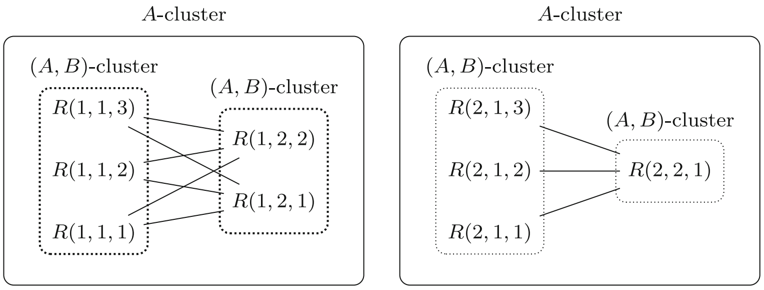

Fig. 2 A- and (A, B)-clusters of I_2

Its conflict graph is presented in Fig. 2. I_2 has two A-clusters each consisting of two (A, B)-clusters (indicated with a dotted line). For instance, the consider the A-cluster $\{R(2, 1, 1), R(2, 1, 2), R(2, 1, 3), R(2, 2, 1)\}$ which consists of two (A, B)-clusters: $\{R(2, 2, 1)\}$ and $\{R(2, 1, 1), R(2, 1, 2), R(2, 1, 3)\}$. □

Finally, we recall the basic database repairing algorithm [28]. Algorithm 1 iterates over the facts of the input instance I in some arbitrary order and creates a repair J. For every fact it adds the fact to J if so does not violate the set of denial constraints F; otherwise the fact is discarded. Naturally, the constructed instance J is consistent with F. Moreover, J is a repair i.e., maximal consistent subset of I, since the algorithm considers adding every fact to the constructed instance. Thus Algorithm 1 is *sound*, it always produces a repair.

We observe that depending on the order in which Algorithm 1 iterates over the facts in input instance, we may obtain different repairs. For instance, in Example 1 the repair I_2' is obtained with the following ordering of the facts of I_0: (1) $Mgr(Mary, \$70k, IT)$, (2) $Emp(John, \$50k, IT)$, (3) $Emp(John, \$40k, IT)$, and (4) $Emp(John, \$80k, IT)$. On the other hand the repair I_3' is obtained with the following ordering of I_0: (1) $Mgr(Mary, \$70k, IT)$, (2) $Emp(John, \$40k, IT)$, (3) $Emp(John, \$50k, IT)$, and (4) $Emp(John, \$80k, IT)$. In fact, for every $I' \in Rep(I, F)$ there exists an ordering of I for which Algorithm 1 returns I': it suffices to take any ordering of I' and append to it any ordering of $I \setminus I'$. Hence, we say that Algorithm 1 is *complete* because it is capable of producing any repair.

Algorithm 1 Constructing a repair of I w.r.t. F

```
1: $I^o \leftarrow I$
2: $J \leftarrow \varnothing$
3: while $I^o \neq \varnothing$ do
4:     choose $R(t) \in I^o$
5:     $I^o \leftarrow I^o \setminus \{R(t)\}$
6:     if $J \cup \{R(t)\} \models F$ then
7:         $J \leftarrow J \cup \{R(t)\}$
8: return $J$
```

2.4 Complexity classes

We make use of the following complexity classes:

- LOGSPACE: the class of decision problems solvable in logarithmic space by deterministic Turing machines (the input tape is read-only);
- PTIME: the class of decision problems solvable in polynomial time by deterministic Turing machines;
- coNP: the class of decision problems whose complements are solvable in polynomial time by nondeterministic Turing machines;
- Π_2^p: the class of decision problems whose complements are solvable in polynomial time by nondeterministic Turing machines with an NP oracle.

We remark that these complexity classes are used only to measure the *data complexity* i.e., the complexity expressed in terms of the size of the database size only [31] (cf. Section 3.3).

3 Conflict resolution preferences

To represent the preference information we use a relation on pairs of neighboring facts i.e., pairs of facts present in a conflict. Resolving a conflict consists of deleting one of its elements and the relation is used to indicate those tuples that the user prefers to keep in the database. We observe, however, that a cycle in the relation may make the choice of the tuple to keep ambiguous, if not impossible. Consequently, we work with acyclic relations only.

Definition 6 (Priority) Given an instance I and a set of denial constraints F, a *priority* $\succ$ of I w.r.t. F is a binary relation on I such that: (1) $\succ$ is acyclic and (2) for every $R(t), R'(t') \in I$ if $R(t) \succ R'(t')$, then $R(t)$ and $R'(t')$ are neighbors.

In the sequel, we omit the reference to the instance I and the set of denial constraints F if they are known from the context.

From the point of the user interface it is often more natural to define the priority as some acyclic binary relation on facts of I and then consider the restriction of the priority relation to the conflicting facts. Clearly, this approach can be handled with the notion of priorities.

To help visualizing the priority we use the *prioritized conflict hypergraphs*. Basically, we extend the conflict hypergraph with directed edges corresponding to the priority relation: $R(t) \rightarrow P(s)$ reads $R(t) \succ P(s)$. The examples we present in this paper use only conflict graphs i.e., conflict hypergraphs where edges connect exactly two nodes. Consequently, a prioritized graph can be seen as a graph with some of its edges oriented. For instance, Fig. 3 contains the conflict graph for the instance in Example 1 with the priority corresponding to the following preference: if the database contains an employee who earns more than her manager, then the information about the employee should be removed.

Definition 7 (Priority extension) Given an instance I, a set of denial constraints F, and two priorities $\succ$ and $\succ'$ of I w.r.t. F, $\succ'$ is an *extension* of $\succ$, denoted $\succ \subseteq \succ'$ if

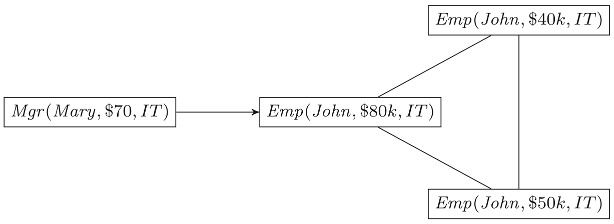
Fig. 3 Prioritized conflict graph

and only if $R(t) \succ' R'(t')$ whenever $R(t) \succ R'(t')$ for $R(t), R'(t') \in I$. A priority $\succ$ of I w.r.t. F is *total* if there exists no priority $\succ'$ of I w.r.t. F that is different from $\succ$ and extends $\succ$.

Note that both an extension of a priority and a total priority are also acyclic and defined on pairs of neighboring facts only.

Proposition 2 *A priority $\succ$ is total if and only if for every conflict C and any two facts $x_1, x_2 \in C$ we have that either $x_1 \succ x_2$ or $x_2 \succ x_1$.*

Proof The *if* part is trivial. For the *only if* part suppose there is a priority $\succ$ that is total yet there exists neighboring x_1 and x_2 such that $x_1 \not\succ x_2$ and $x_2 \not\succ x_1$ i.e., both $\succ_1 = \succ \cup \{(x_1, x_2)\}$ and $\succ_2 = \succ \cup \{(x_2, x_1)\}$ are cyclic. Since $\succ$ is not cyclic, $\succ_1$ has a cycle that traverses (x_1, x_2) i.e., there exists a chain $x_2 \succ y_1 \succ \ldots \succ y_n \succ x_1$. Similarly, $\succ_2$ being cyclic implies that there exists a chain $x_1 \succ z_1 \succ \ldots \succ z_m \succ x_2$. Together this implies that $x_1 \succ \ldots \succ x_2 \succ \ldots \succ x_1$; a contradiction. To finish the proof we observe that the acyclicity of priority implicitly excludes the possibility of both $x \succ y$ and $y \succ x$ being true at the same time for some facts x and y. □

3.1 Preferred repairs and consistent query answers

Now, we introduce the general framework of prioritized repairing and query of inconsistent databases. We begin by defining a general notion of a family of preferred repairs. We do not make any assumptions on how such a family constructs preferred repairs. For generality, we do not even assume that the constructed instances are repairs in the sense of Definition 3. Instead, we list later on the desirable properties that a well-behaved family should satisfy.

Definition 8 (Preferred repairs) A *family of preferred repairs* is a function $\mathcal{X}Rep$ defined on triplets $(I, F, \succ)$, where $\succ$ is a priority in I w.r.t. a set of denial constraints F, such that $\mathcal{X}Rep(I, F, \succ)$ is a set of database instances over the same schema. We say that a family $\mathcal{Y}Rep$ *subsumes* a family $\mathcal{X}Rep$, denoted $\mathcal{X}Rep \sqsubseteq \mathcal{Y}Rep$, if $\mathcal{X}Rep(I, F, \succ) \subseteq \mathcal{Y}Rep(I, F, \succ)$ for every $(I, F, \succ)$.

We generalize the notion of consistent query answers [4] by considering only preferred repairs when evaluating a query (instead of all repairs). We can easily generalize our approach to open queries as in [13, 15].

Definition 9 ($\mathcal{X}$-preferred consistent query answer) Given a closed query Q, a triple $(I, F, \succ)$, and a family of preferred repairs $\mathcal{X}Rep$, **true** (**false**) is the *$\mathcal{X}$-preferred consistent query answer* to Q in I w.r.t. F and $\succ$ if and only if for every $I' \in \mathcal{X}Rep(I, F, \succ)$ we have $I' \models Q$ ($I' \not\models Q$ respectively).

Note that we obtain the original notion of consistent query answer if we consider the family of all repairs $Rep(I, F)$.

3.2 Desirable properties of preferred repairs

Now, we identify desirable properties of arbitrary families of preferred repairs. The properties should be satisfied for an arbitrary instance I and an arbitrary set of denial constraints F.

3.2.1 $\mathcal{P}1$ **Non-emptiness**

Because the set of preferred repairs is used to define preferred consistent query answers, it is important that for any preference the framework is not trivialized by an empty set of preferred repairs:

$$\mathcal{X}Rep(I, F, \succ) \neq \emptyset.$$

3.2.2 $\mathcal{P}2$ **Monotonicity**

The operation of extending the preference allows to improve the state of our knowledge of the real world. The better such knowledge is the finer the (preferred consistent) answers we should obtain. This is achieved if extending the preference can only narrow the set of preferred repairs:

$$\succ_1 \subseteq \succ_2 \implies \mathcal{X}Rep(I, F, \succ_2) \subseteq \mathcal{X}Rep(I, F, \succ_1).$$

3.2.3 $\mathcal{P}3$ **Non-discrimination**

Removing repairs from consideration must be justified by existing preference. In particular, no repair should be removed if no preference is given:

$$\mathcal{X}Rep(I, F, \emptyset) = Rep(I, F).$$

3.2.4 $\mathcal{P}4$ **Categoricity**

Ideally, a preference that cannot be further extended (the priority is total) should specify how to resolve every conflict:

$$\succ \text{ is total} \implies |\mathcal{X}Rep(I, F, \succ)| = 1.$$

3.2.5 $\mathcal{P}5$ **Conservativeness**

We also note that properties $\mathcal{P}2$ and $\mathcal{P}3$ together imply that preferred repairs are a subset of all repairs:

$$\mathcal{X}Rep(I, F, \succ) \subseteq Rep(I, F).$$

In fact, in the remainder of the paper we consider only families of preferred repairs that satisfy $\mathcal{P}5$. We also observe that $\mathcal{P}5$ with $\mathcal{P}1$ imply that the only preferred repair of a consistent database instance is the instance itself.

3.3 Data complexity

We also adapt the decision problems to include the priority. Note that the priority relation is of size quadratic in the size of the database instance, and therefore, it is natural to make it a part of the input. For a family $\mathcal{X}Rep$ of preferred repairs the decision problems we study are defined as follows:

(i) $\mathcal{X}$*-preferred repair checking* i.e., the complexity of the following set

$$\mathcal{B}_F^{\mathcal{X}} = \{(I, \succ, I') : I' \in \mathcal{X}Rep\,(I, F, \succ)\}.$$

(ii) $\mathcal{X}$*-preferred consistent query answering* i.e., the complexity of the following set

$$\mathcal{D}_{F,Q}^{\mathcal{X}} = \{(I, \succ) : \forall I' \in \mathcal{X}Rep\,(I, F, \succ)\,.I' \models Q\}.$$

4 Globally-optimal repairs

We investigate several different families of preferred repairs. The first family of preferred repairs is based on the notion of optimal compliance of the repair with the priority. Essentially, the compliance of a repair can be improved by replacing a subset of facts with a *more preferred* subset of facts. The way we define a set of facts being more preferred than another set of facts is inspired by the work on preferred models of logic programs [30] and preferential reasoning [23].

Definition 10 (Globally-optimal repairs $\mathcal{G}Rep$) Given an instance I, a set of denial constraints F, and a priority $\succ$, an instance $I' \subseteq I$ is *globally optimal* w.r.t. $\succ$ and F if no nonempty subset X of facts from I' can be replaced with a subset Y of $I \setminus I'$ such that

$$\forall x \in X.\, \exists y \in Y.\, y \succ x \qquad (*_{\mathcal{G}})$$

and the resulting set of facts is consistent with F. $\mathcal{G}Rep$ is the family of globally-optimal repairs i.e., $\mathcal{G}Rep(I, F, \succ)$ is the set of all repairs of I w.r.t. F that are globally optimal w.r.t. $\succ$ and F.

We emphasize that the family $\mathcal{G}Rep$ selects all globally-optimal repairs. In general, it is, however, possible to define a family that selects only some of the globally-optimal repairs, or even more generally, a family that constructs a set of globally-optimal instances that need not be repairs.

The notion of global optimality identifies repairs whose compliance with the priority cannot be further improved. For the instance I_0 in Example 1 with the priority in Fig. 3 the set of globally-optimal repairs consists of I_2' and I_3'.

In the sequel, we fix an instance I and a set of denial constraints F, and omit them when referring to the elements of $\mathcal{G}Rep(I, F, \succ)$. Before investigating the properties of $\mathcal{G}Rep$ we present an alternative characterization of globally-optimal repairs.

Proposition 3 *For a given priority $\succ$ and two repairs I'_1 and I'_2, I'_1* globally dominates *I'_2, denoted $I'_1 \gg_{\mathcal{G}} I'_2$, if*

$$\forall x \in I'_2 \setminus I'_1.\ \exists y \in I'_1 \setminus I'_2.\ y \succ x. \qquad (\bigstar_{\mathcal{G}})$$

The following facts hold:

(i) *a repair I' is globally optimal if and only if it is $\gg_{\mathcal{G}}$-maximal i.e., there is no repair I'' different from I' such that $I'' \gg_{\mathcal{G}} I'$;*
(ii) *if $\succ$ is acyclic, then so is $\gg_{\mathcal{G}}$.*

Proof

(i) We prove the contraposition i.e., I' is not globally optimal if and only if there exists a repair $I'' \neq I'$ such that $I'' \gg_{\mathcal{G}} I'$. For the *if* part take $X = I' \setminus I''$ and $Y = I'' \setminus I'$, and note that $(*_{\mathcal{G}})$ follows from $(\bigstar_{\mathcal{G}})$. Naturally, $(I' \setminus X) \cup Y = I''$ is consistent. For the *only if* part take any nonempty $X \subseteq I'$ and $Y \subseteq I \setminus I'$ such that $(*_{\mathcal{G}})$ is satisfied and $J = (I' \setminus X) \cup Y$ is consistent. We take any repair I'' that contains J. Such a repair exists since J is consistent. Clearly, $I' \setminus I'' \subseteq X$ and also $Y \subseteq I'' \setminus I'$. Hence $(\bigstar_{\mathcal{G}})$ follows from $(*_{\mathcal{G}})$. Consequently, I' is not globally optimal.
(ii) Suppose $\gg_{\mathcal{G}}$ is cyclic i.e., there exists a sequence of different repairs $I'_0, \ldots, I'_{n-1}$ such that $I'_i \gg I'_{i+1}$ for $i \in \{0, \ldots, n-1\}$, where the $+$ operator is interpreted modulo n. We show that $\succ$ is cyclic as well. We construct inductively infinite sequences of facts $y_1, y_2, \ldots$ and numbers $k_1, k_2, \ldots$ such that $y_{j+1} \succ y_j$ for $j \in \mathbb{N}$ and $y_j \notin I'_{k_j}$ and $y_j \in I'_{k_j+1}$ for $j \in \mathbb{N}$.
For $j = 1$ let y_1 be any element of $I'_1 \setminus I'_0$ and $k_1 = 1$. Now, suppose we have constructed the two sequences up to their j-th elements y_j and k_j such that $y_j \notin I'_{k_j}$ and $y_j \in I'_{k_j+1}$. If $y_j \in I'_0$, then y_j must have been *pushed out* somewhere between I'_0 and I'_{k_j} i.e., there exists $k_{j+1} \in \{0, \ldots, k_j - 1\}$ such that $y_j \in I'_{k_{j+1}}$ and $y_j \notin I'_{k_{j+1}+1}$. By $I'_{k_{j+1}+1} \gg_{\mathcal{G}} I'_{k_{j+1}}$ there exists an element $y_{j+1} \in I'_{k_{j+1}+1} \setminus I'_{k_{j+1}}$ such that $y_{j+1} \succ y_j$. The case when $y_j \notin I'_0$ is treated symmetrically: y_j must have been pushed out somewhere between I'_{k_j+1} and $I'_n = I'_0$.
Clearly, I has only a finite number of elements and thus any infinite $\succ$-chain must have a repetition, and consequently $\succ$ is cyclic. □

Proposition 4 *$\mathcal{G}Rep$ satisfies the properties $\mathcal{P}1$–$\mathcal{P}4$.*

Proof We get $\mathcal{P}1$ by acyclicity of $\gg_{\mathcal{G}}$ and Proposition 3. To show $\mathcal{P}2$ we observe that if a repair is globally optimal w.r.t. $\succ_2$, then it is globally optimal w.r.t. any $\succ_1$ such that $\succ_1 \subseteq \succ_2$. $\mathcal{P}3$ follows directly from definition: to show that a repair is not globally optimal, $\succ$ needs to be nonempty.

Showing $\mathcal{P}4$ requires a more elaborate argument. Take a total $\succ$. By $\mathcal{P}1$ there exists at least one globally-optimal repair. Suppose that there exist two different globally-optimal repairs I'_0 and I'_1. In the remaining part of the proof for $i \geq 2$ we let $I'_i = I'_{i \bmod 2}$. We show that $\succ$ is cyclic by creating an infinite chain $\ldots \succ x_1 \succ x_0$ such that $x_i \in I'_i \setminus I'_{i+1}$ for every $i \in \mathbb{N}$. For x_0 we take any element from $I'_0 \setminus I'_1$. Now, assuming that the sequence has been defined up to the i-th element x_i, we choose x_{i+1} to be any element of $I'_{i+1} \setminus I'_i$ such that $x_i \succ x_{i+1}$. We show the existence of

x_{i+1} using global optimality of I'_{i+1} as follows. First, we observe that the instance $I'_{i+1} \cup \{x_i\}$ is inconsistent since $x_i \notin I'_{i+1}$ and I'_{i+1} is a repair i.e., a maximal consistent subset of I. Let $C_1, \ldots, C_k$ be all conflicts present in $I'_{i+1} \cup \{x_i\}$. Clearly, for every $j \in \{1, \ldots, k\}$ the conflict C_j contains a fact $z_j \notin I'_i$ since $C_j \not\subseteq I'_i$ by the consistency of I'_i. Let $X = \{z_1, \ldots, z_k\}$ and $Y = \{x_i\}$. Naturally, $(I'_{i+1} \setminus X) \cup Y$ is consistent, and thus by global optimality of I'_{i+1} there exists an element $x_{i+1} \in X$ such that $x_i \not\succ x_{i+1}$. But by totality of $\succ$ and the fact that every element of X is a neighbor of x_i, we have that $x_{i+1} \succ x_i$. Clearly, $x_{i+1} \notin I'_i$, and moreover, $x_{i+1} \in I'_{i+1}$ because $x_{i+1} \in C_j \setminus \{x_i\} \subseteq I'_{i+1}$ for some $j \in \{1, \ldots, k\}$. This shows that $\succ$ is cyclic; a contradiction. □

Now, we present Algorithm 2 that constructs globally-optimal repairs. It begins with an arbitrary repair I' obtained with Algorithm 1 and then iteratively attempts to improve the compliance of the repair with the priority. At each iteration it replaces a subset $X \subseteq I'$ of facts with a more preferred subset $Y \subseteq I \setminus I'$ and extends the obtained consistent instance $J = (I' \setminus X) \cup Y$ to a repair I'' in a manner analogous to the way Algorithm 1 creates a repair: by attempting to add to J any fact from $I \setminus J$ as long as doing so does not create a conflict.

Algorithm 2 Constructing a globally-optimal repair of I w.r.t. F

```
1: construct a repair $I'$    /*Algorithm 1*/
2: while $\exists X \subseteq I'. \exists Y \subseteq I \setminus I'. \forall x \in X. \exists y \in Y. y \succ x$ do
3:     $J \leftarrow (I' \setminus X) \cup Y$
4:     extend $J$ to a repair $I''$    /*Algorithm 1*/
7:     $I' \leftarrow I''$
8: return $I'$
```

Naturally, Algorithm 2 is sound because its main loop stops only if the instance I' is globally optimal and since $\gg_{\mathcal{G}}$ is acyclic, the loop always terminates. It is also complete because it is based on Algorithm 1 which constructs any repair, in particular any globally-optimal repair can be constructed in the line 1: of Algorithm 2. We observe that if I'_i is the repair constructed in the i-th iteration of the main loop, then $I'_{i+1} \gg_{\mathcal{G}} I'_i$. Since $\gg_{\mathcal{G}}$ is acyclic and the number of repairs bounded by an exponential function of the size of I, the algorithm performs at most an exponential number of iterations. Checking global optimality (line 2:) can be done in exponential time, and thus the algorithm works in exponential time.

Theorem 1 *Algorithm 2 is a sound and complete algorithm constructing globally-optimal repairs. It works in time exponential in the size of the input instance and the priority relation.*

Algorithm 2 follows a rather simple principle: start with an arbitrary repair and iteratively improve its compliance with the priority until an optimal one is obtained. For such an approach to be tractable, two concerns would need to be addressed: (1) the preferred repair checking problem needs to be in PTIME and (2) the number of possible iterations needs to be bounded by a polynomial. Later on we show that $\mathcal{G}$-preferred repair checking is coNP-complete (Theorem 2) which shows that this approach cannot be tractable (unless P = NP), and furthermore, it suggests that there does not exist a tractable sound and complete algorithm constructing $\mathcal{G}$-preferred repairs. However, for other families of preferred repairs considered in this paper the preferred repair checking problem is in PTIME. In the following example we construct a $\ll_{\mathcal{G}}$-chain of exponential length, thus showing that the number of iterations of Algorithm 2 may be exponential. The same construction shows that for the other families of repairs an algorithm based on the same principle might require an exponential number of iterations. Consequently, more sophisticated solutions are required.

Example 4 For a given $n \in \mathbb{N}$ we construct an instance I_n and a priority $\succ_n$ such that the size of I_n is $O(n)$, the size of $\succ_n$ is $O(n^2)$, and there exists a $\gg_G$-chain of length $\Omega(2^n)$.

Intuitively, we construct a chain of repairs which emulates a n-bit binary counter, incremented from $0 = (0 \cdots 0)_2$ to $2^n - 1 = (1 \cdots 1)_2$. Incrementing a counter consists of setting to 1 the least significant bit with value 0 and setting to 0 all the preceding bits (up to this point all set to 1). For instance, if $n = 3$ and we wish to increment the number $3 = (011)_2$, then we obtain $4 = (100)_2$ by setting to 1 the third bit and setting to 0 the first and second bit. This operation can be seen as a (cascading) propagation of the carry bit. Notice that even numbers have their least significant bit set to 0 and thus require no propagation of the carry bit.

We work with instances of one relation only $R(A, B)$ and the constructed instance I_n comprises of the following facts:

- $p_i^0 = R(i, 0)$ representing the i-th bit set to 0, for $i \in \{0, \ldots, n-1\}$
- $p_i^1 = R(i, 1)$ representing the i-th bit set to 1, for $i \in \{0, \ldots, n-1\}$,
- $p_i^c = R(i, 2)$ representing the i-th bit being carried over to the $(i+1)$-th bit, for $i \in \{0, \ldots, n-2\}$.

To ensure proper behavior of the counter we use the following three constraints:

$$R : A \to B,$$
$$\forall i, j.\ \neg[R(i, 2) \wedge R(j, 1) \wedge i > j],$$
$$\forall i, j.\ \neg[R(i, 1) \wedge R(j, 2) \wedge j = i - 1].$$

The first constraint ensures that a bit is set to 0, set to 1, or being carried to the higher bit. The second constraint ensures that propagating a carry bit resets all lower bits to 0. The third constraint ensures that a bit can be carried over only if the immediately higher bit is set to 0. The correct order of increment is ensured by the priority relation $\succ_n$ defined as:

$$p_i^1 \succ_n p_i^0 \qquad \text{for } i \in \{0, \ldots, n-1\},$$
$$p_i^1 \succ_n p_{i-1}^c \qquad \text{for } i \in \{1, \ldots, n-1\},$$
$$p_i^c \succ_n p_j^1 \qquad \text{for } i \in \{1, \ldots, n-2\} \text{ and } j \in \{0, \ldots, i\}.$$

Notice that $p_i^x \succ_n p_j^y$ implies that either $i > j$ or $i = j$, $x = 1$, and $y = 0$. Consequently, $\succ_n$ is acyclic.

Now, we construct a $\ll_{\mathcal{G}}$-chain of repairs that corresponds to subsequent natural numbers ranging from 0 to $2^n - 1$. Additionally, for odd numbers the chain contains also repairs that represent the cascading propagation of the carry bit. Figure 4 contains an example of an instance I_3 and a sequence of repairs that constitutes a $\gg_{\mathcal{G}}$-chain.

For instance, I'_0 and I'_1 correspond to $0 = (000)_2$ and $1 = (001)_2$ respectively while $I'_{1,c}$ corresponds to 1 being incremented with a carry bit. For every $i \in \{0, \ldots, 2^n - 1\}$ let $(b^i_0, b^i_1, \ldots, b^i_{n-1})$ be the binary representation of i, where $b^i_j \in \{0, 1\}$ and b^i_0 denotes the least significant bit i.e., $\sum_{j=0}^{n-1} 2^j b^i_j = i$. The repair corresponding to $i \in \{0, \ldots, 2^n - 1\}$ is

$$I'_i = \left\{ p_0^{b^i_0}, p_1^{b^i_1}, \ldots, p_{n-1}^{b^i_{n-1}} \right\}.$$

For every odd $i \in \{1, 3, \ldots, 2^n - 3\}$ we also construct the repair that propagates the carry bit in a cascading fashion

$$I'_{i,c} = \left\{ p^0_0, \ldots, p^0_{j_i - 2}, p^c_{j_i - 1}, p_{j_i}^{b^i_{j_i}}, \ldots, p_{n-1}^{b^i_{n-1}} \right\},$$

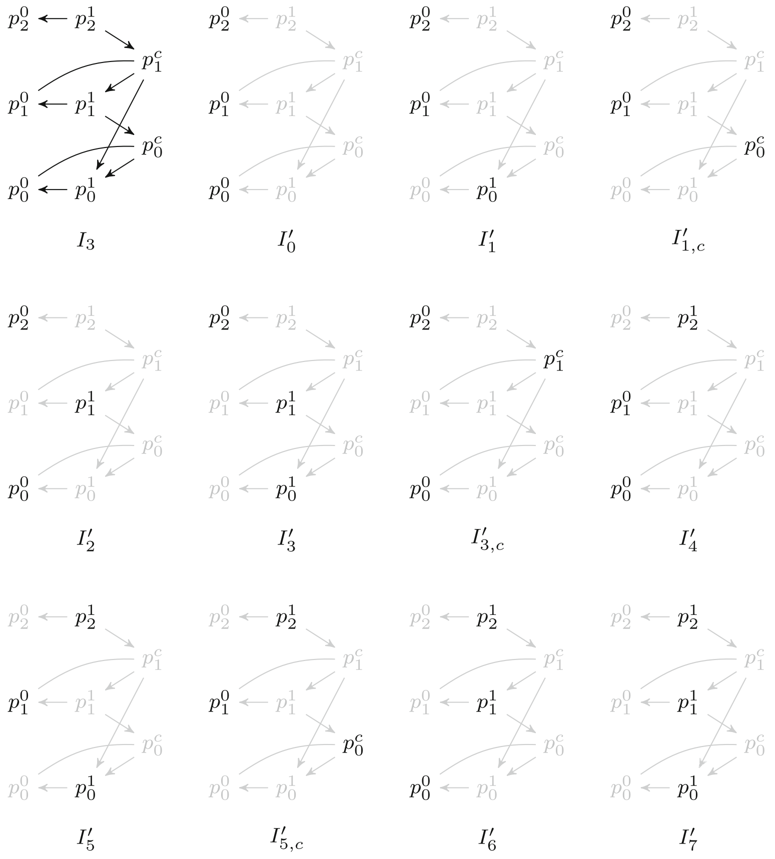

Fig. 4 The instance I_3 and the chain $I'_7 \gg_{\mathcal{G}} I'_6 \gg_{\mathcal{G}} I'_{5,c} \gg_{\mathcal{G}} I'_5 \gg_{\mathcal{G}} I'_4 \gg_{\mathcal{G}} I'_{3,c} \gg_{\mathcal{G}} I'_3 \gg_{\mathcal{G}} I'_2 \gg_{\mathcal{G}} I'_{1,c} \gg_{\mathcal{G}} I'_1 \gg_{\mathcal{G}} I'_0$

where j_i is the position of the least significant bit of the binary representation of i that is set to 0 i.e., the minimal j such that $b^i_j = 0$. It can be easily shown that

$$I'_{2^n-1} \gg_{\mathcal{G}} I'_{2^n-2} \gg_{\mathcal{G}} I'_{2^n-3,c} \gg_{\mathcal{G}} I'_{2^n-3,c} \gg_{\mathcal{G}} \dots$$
$$\dots \gg_{\mathcal{G}} I'_{3,c} \gg_{\mathcal{G}} \gg_{\mathcal{G}} I'_3 \gg_{\mathcal{G}} \gg_{\mathcal{G}} I'_2 \gg_{\mathcal{G}} \gg_{\mathcal{G}} I'_{1,c} \gg_{\mathcal{G}} I'_1 \gg_{\mathcal{G}} I'_0.$$

Finally, we observe that in the worst case scenario Algorithm 2 may traverse the full length of the constructed chain during its execution with I_n and $\succ_n$. We remark, however, that in this example the globally-optimal repair I'_{2^n-1} may be attained in just one iteration of the main loop i.e., $I'_{2^n-1} \gg_{\mathcal{G}} I'_i$ for $i \in \{0, \dots, 2^n - 2\}$ and $I'_{2^n-1} \gg_{\mathcal{G}} I'_{i,c}$ for $i \in \{1, 3, \dots, 2^n - 3\}$. □

Now, we investigate computational properties of globally-optimal repairs. We observe that verifying whether a repair I' is not globally optimal can be easily accomplished with a nondeterministic Turing machine: it suffices to guess the sets X and Y, verify that $(I' \setminus X) \cup Y$ is consistent, and check that $(*_{\mathcal{G}})$ holds. Consequently, $\mathcal{B}^{\mathcal{G}}_F$ is in coNP. The membership of $\mathcal{D}^{\mathcal{G}}_{F,Q}$ in Π^p_2 follows from Definition 9: **true** is not the $\mathcal{G}$-preferred consistent answer to a query if the query is not **true** in some globally-optimal repair.

Proposition 5 *$\mathcal{G}$-preferred repair checking is in coNP and $\mathcal{G}$-preferred consistent query answering is in Π^p_2.*

The upper bounds are tight.

Theorem 2 *There exists a set of 4 FDs and an atomic query for which $\mathcal{G}$-preferred repair checking is coNP-hard and $\mathcal{G}$-preferred consistent query answering is Π^p_2-hard.*

Proof We show Π^p_2-hardness of $\mathcal{D}^{\mathcal{G}}_{F,Q}$ by reducing the satisfaction of $\forall^*\exists^*$QBF formulas to $\mathcal{D}^{\mathcal{G}}_{F,Q}$. Consider the following formula:

$$\Psi = \forall x_1, \dots, x_n.\exists x_{n+1}, \dots, x_{n+m}.\Phi,$$

where Φ is (quantifier-free) 3CNF i.e., Φ equals to $c_1 \wedge \dots \wedge c_s$, and c_k is a clause of three literals $\ell_{k,1} \vee \ell_{k,2} \vee \ell_{k,3}$ for $k \in \{1, \dots, s\}$. We call the variables $x_1, \dots, x_n$ *universal* and $x_{n+1}, \dots, x_{n+m}$ *existential*. We use the function q to identify the type of a variable with a given index: $q(i) = 1$ for $i \le n$ and $q(i) = 0$ for $i > n$. We also use the following two auxiliary functions var and sgn on literals of Φ:

$$var(x_i) = var(\neg x_i) = i, \qquad sgn(x_i) = 1, \qquad sgn(\neg x_i) = -1.$$

A *valuation* is a (possibly partial) function assigning a Boolean value to the variables.

We construct instances over the schema consisting of a single relation

$$R\,(A_1, B_1, A_2, B_2, A_3, B_3, A_4, B_4)\,.$$

The set of integrity constraints is

$$F = \{A_1 \to B_1, A_2 \to B_2, A_3 \to B_3, A_4 \to B_4\}\,.$$

The constructed database instance I_Ψ consists of the following facts:

- v_i and $\bar{v}_i$ corresponding to the positive and negative valuations of x_i resp. (for $i \in \{1, \ldots, n+m\}$)

$$v_i = R\,(0, q(i), i, 1, i, 1, i, 1)\,, \qquad \bar{v}_i = R\,(0, q(i), i, -1, i, -1, i, -1)\,,$$

- d_k corresponding to the clause c_k (for $k \in \{1, \ldots, s\}$)

$$d_k = R\left(0, 1, var(\ell_{k,1}), sgn(\ell_{k,1}), var(\ell_{k,2}), sgn(\ell_{k,2}), var(\ell_{k,3}), sgn(\ell_{k,3})\right),$$

- $p_\exists$ and $p_\forall$ used to partition the set of all repairs into repairs that correspond to the valuations of existential and universal variables respectively:

$$p_\exists = R(0, 0, 0, 0, 0, 0, 0, 0), \qquad p_\forall = R(0, 1, 0, 0, 0, 0, 0, 0).$$

For the ease of reference by $L_{k,p}$ we denote the fact corresponding to the satisfying valuation of literal $\ell_{k,p}$ i.e.:

$$L_{k,p} = \begin{cases} v_i & \text{when } \ell_{k,p} = x_i, \\ \bar{v}_i & \text{when } \ell_{k,p} = \neg x_i. \end{cases}$$

The constructed priority relation $\succ_\Psi$ is the minimal priority of I_Ψ w.r.t. F such that:

$$\begin{aligned} &v_i \succ_\Psi d_k, && \text{if } c_k \text{ uses a positive literal } x_i, \\ &\bar{v}_i \succ_\Psi d_k, && \text{if } c_k \text{ uses a negative literal } \neg x_i, \\ &p_\exists \succ_\Psi v_i, && \text{for all } i \in \{1, \ldots, n\}, \\ &p_\exists \succ_\Psi \bar{v}_i, && \text{for all } i \in \{1, \ldots, n\}, \\ &p_\exists \succ_\Psi p_\forall. && \end{aligned}$$

Figure 5 contains a prioritized conflict graph of the instance and the priority obtained for the formula:

$$\Psi_0 = \forall x_1, x_2, x_3. \exists x_4, x_5. (\neg x_1 \vee x_4 \vee x_2) \wedge (\neg x_2 \vee \neg x_5 \vee \neg x_3).$$

The query used in the reduction is $Q = p_\exists$ and we claim that Ψ is valid if and only if **true** is $\mathcal{G}$-preferred consistent query answer to $p_\exists$ in I_Ψ w.r.t. F and $\succ_\Psi$. The proof is technically elaborate but can be summarized as follows. First, we partition the set of repairs into $\exists$- and $\forall$-repairs that correspond to valuations of existential and

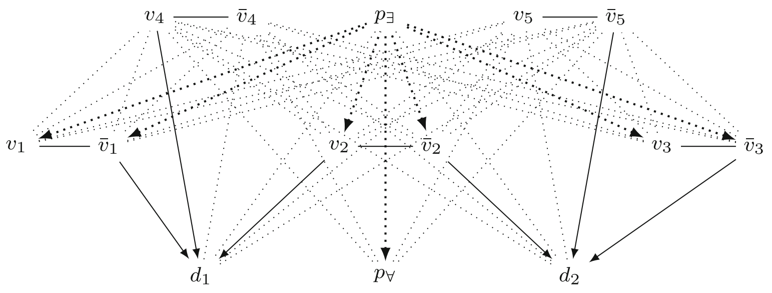

Fig. 5 The prioritized conflict graph for Ψ_0. Dotted lines used for conflicts w.r.t. $A_1 \rightarrow B_1$

universal variables. Next, we show that an $\exists$-repair globally dominates a $\forall$-repair iff the combined valuation satisfies Φ. Consequently, we argue that if the $\exists$-repairs are the only globally-optimal repairs, then for every valuation of universal variables there exists a valuation of existential variables that together satisfy Φ i.e., Ψ is valid.

We partition the set of all repairs of I_Ψ into two disjoint classes: $\exists$-*repairs* that contain $p_\exists$ and $\forall$-*repairs* that do not contain $p_\exists$. Because of the FD $A_1 \rightarrow B_1$ every $\forall$-repair contains $p_\forall$. For the same reason, a $\forall$-repair is always a subset of $\{v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n, d_1, \ldots, d_n, p_\forall\}$ whereas an $\exists$-repair is always a subset of $\{v_{n+1}, \bar{v}_{n+1}, \ldots, v_{n+m}, \bar{v}_{n+m}, p_\exists\}$.

We use $\exists$- and $\forall$-repairs to represent all possible valuation of existential and universal variables respectively. To easily move from a valuation of variables to a repair we define the following two operators:

$$I_\exists[V] = \{v_i \mid V(x_i) = \textbf{true} \wedge q(i) = 0\} \cup \{\bar{v}_i \mid V(x_i) = \textbf{false} \wedge q(i) = 0\} \cup \{p_\exists\},$$

$$I_\forall[V] = \{v_i \mid V(x_i) = \textbf{true} \wedge q(i) = 1\} \cup \{\bar{v}_i \mid V(x_i) = \textbf{false} \wedge q(i) = 1\} \cup \{p_\forall\} \cup \left\{ d_k \,\middle|\, \begin{array}{l} \text{if for every literal } \ell_{k,i} \text{ of } c_k \text{ that uses a universal} \\ \text{variable for which } V \text{ is defined, we have } V \not\models \ell_{k,i} \end{array} \right\}.$$

Note that $I_\forall[V]$ contains the types corresponding to clauses that are not satisfied by the valuation of universal variables V alone.

For instance, take the formula Ψ_0 in Fig. 5 and the following total valuation V_0 of variables $x_1, \ldots, x_5$:

$$V_0(x_1) = \textbf{true}, \quad V_0(x_2) = \textbf{false}, \quad V_0(x_3) = \textbf{false}, \quad V_0(x_4) = \textbf{true}, \quad V_0(x_5) = \textbf{true}.$$

Then, the repairs corresponding to the valuation of existential and universal variables are

$$I_\exists[V_0] = \{v_4, v_5, p_\exists\} \quad \text{and} \quad I_\forall[V_0] = \{v_1, \bar{v}_2, \bar{v}_3, d_1, p_\forall\}.$$

To move in the opposite direction, from a repair to a (possibly partial) valuation we use:

$$V[I'](x_i) = \begin{cases} \textbf{true} & \text{if } v_i \in I', \\ \textbf{false} & \text{if } \bar{v}_i \in I', \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We observe that $V[\cdot]$ defines a one-to-one correspondence between $\exists$-repairs and total valuations of existential variables. A similar statement, however, does not hold for $\forall$-repairs because of the interaction between facts d_k and the facts corresponding to universal variables. For example, for the instance in Fig. 5 if we take the repair $I_0 = \{v_1, v_3, d_1, d_2, p_\forall\}$, the corresponding valuation $V[I_0]$ of universal variables is undefined for x_2.

Consequently, for some $\forall$-repair I' the function $V[I']$ may be only a partial valuation of universal variables. We call a $\forall$-repair I' *strict* if $V[I']$ is a total valuation of universal variables. In this way, $V[\cdot]$ defines a one-to-one correspondence between strict $\forall$-repairs and total valuations of the universal variables. The following result allows us to remove non-strict $\forall$-repairs from consideration.

Lemma 1 *Strict $\forall$-repairs are exactly $\gg_{\mathcal{G}}$-maximal $\forall$-repairs.*

Proof First, we prove that no non-strict $\forall$-repair is $\gg_{\mathcal{G}}$-maximal. For that we show how to construct from a non-strict $\forall$-repair I' a strict $\forall$-repair I'' such that $I'' \gg_{\mathcal{G}} I'$. We take the partial valuation $V' = V[I']$ and extend it to a total valuation V'' of universal variables by assigning **false** value to variables undefined by V' i.e.,

$$V'' = V' \cup \{(x_i, \textbf{false}) \mid 1 \leq i \leq n \land V'(x_i) \text{ is undefined}\}.$$

Now, we go back to the repair $I'' = I_{\forall}[V'']$ and show that

$$\forall q' \in I' \setminus I''.\ \exists q'' \in I'' \setminus I'.\ q'' \succ q'.$$

There are 4 cases of values of q' to consider:

1. $q' = p_{\exists}$, $q' = v_i$, or $q' = \bar{v}_i$ for $i \in \{n+1, \ldots, n+m\}$ is not possible because neither of I' and I'' contains these facts (being $\forall$-repairs)
2. $q' = p_{\forall}$ is not possible because both I' and I'' are $\forall$-repairs.
3. $q' = v_i$ or $q' = \bar{v}_i$ for some $i \in \{1, \ldots, n\}$ is also impossible because from the construction of I'' we know that
$$I'' \cap \{v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n\} \subseteq I' \cap \{v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n\}.$$
4. $q' = d_k$ for some $k \in \{1, \ldots, s\}$. The neighborhood of d_k in the conflict graph consists of facts $p_{\exists}$, $L_{k,1}$, $L_{k,2}$, and $L_{k,3}$. We observe that none of these facts belongs to I'. However, one of the facts must belong to I'' because $q' \notin I''$ and I'' is a maximal consistent subset of I_{Ψ}. Since I'' is a $\forall$-repair, $p_{\exists}$ does not belong to I''. Therefore, for some $p \in \{1, 2, 3\}$ the fact $L_{k,p}$ must belong to I''. Consequently, $q'' = L_{k,p} \succ_{\Psi} q'$.

Now, we show that every strict $\forall$-repair is also $\gg_{\mathcal{G}}$-maximal among $\forall$-repairs. Suppose otherwise i.e., for some strict $\forall$-repair I' there exists an $\forall$-repair I'' such that $I' \gg_{\mathcal{G}} I''$. Since I' is strict it contains v_i or $\bar{v}_i$ for every $i \in \{1, \ldots, n\}$. By the construction of the priority $\succ_{\Psi}$ the repairs I' and I'' must agree on facts $v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n$. Therefore $I' = I_{\forall}[V[I'']]$ and using the reasoning from the previous part we can show that $I'' \gg_{\mathcal{G}} I'$. Since $\succ_{\Psi}$ is acyclic, by Proposition 3 this gives us $I' = I''$. □

The central result in our reduction follows.

Lemma 2 *For any total valuation V, $I_{\exists}[V] \gg_{\mathcal{G}} I_{\forall}[V]$ if and only if $V \models \Phi$.*

Proof For the *if* part, because a $\forall$-repair is disjoint with any $\exists$-repair, it is enough to show that for any fact $q' \in I_{\forall}[V]$ there exists a fact $q'' \in I_{\exists}[V]$ such that $q'' \succ q'$. For $p_{\forall}, v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n$ we simply choose $p_{\exists}$. If d_k belongs to $I_{\forall}[V]$, then none of the neighbors of d_k belongs to $I_{\forall}[V]$. This implies that none of the literals using a universal variable is satisfied by V. Hence there must exist a literal $\ell_{k,p}$ of the clause $c_{k,p}$ that uses an existential variables and that is satisfied by V. Consequently, we have $L_{k,p} \in I_{\exists}[V]$ and $L_{k,p} \succ_{\Psi} d_k$.

For the *only if* part take any $k \in \{1, \ldots, s\}$ and consider the conjunct $c_k = \ell_{k,1} \lor \ell_{k,2} \lor \ell_{k,3}$. If none of the literals, which use universal variables, is satisfied by V, then none of the corresponding $L_{k,p}$ belongs to $I_{\forall}[V]$, and consequently, d_k is in $I_{\forall}[V]$.

Then $I_\exists[V]$ must contain a fact $L_{k,p'}$ corresponding to one of the literals of c_k using an existential variable. This implies that $V \models \ell_{k,p'}$, and consequently, $V \models c_k$. □

This gives us.

Corollary 1 *The QBF Ψ is valid if and only if for any strict $\forall$-repair I' there exists an $\exists$-repair I'' such that $I'' \gg_{\mathcal{G}} I'$.*

Because only an $\exists$-repair can be preferred over a strict $\forall$-repair and for every non-strict $\forall$-repair there is a more preferred strict $\forall$-repair, we can make a more general statement.

Corollary 2 *The QBF Ψ is valid if and only if for any $\forall$-repair I' there exists a repair I'' such that $I'' \gg_{\mathcal{G}} I'$.*

$\forall$-repairs are defined as repairs that do not contain the fact $p_\exists$ and thus:

$$\begin{aligned}
&\models \forall x_1, \ldots, x_n.\, \exists x_{n+1}, \ldots, x_{n+m}.\, \Phi && \text{iff}\\
&\forall I' \in Rep(I_\Psi, F).\, [I' \models \neg p_\exists] \Rightarrow [\exists I'' \in Rep(I_\Psi, F).\, I'' \gg_{\mathcal{G}} I'] && \text{iff}\\
&\forall I' \in Rep(I_\Psi, F).\, [\nexists I'' \in Rep(I_\Psi, F).\, I'' \gg_{\mathcal{G}} I'] \Rightarrow [I' \models p_\exists] && \text{iff}\\
&\forall I' \in \mathcal{G}Rep(I_\Psi, F, \succ_\Psi).\, I' \models p_\exists && \text{iff}\\
&(I_\Psi, \succ_\Psi) \in \mathcal{D}^{\mathcal{G}}_{F,p_\exists}.
\end{aligned}$$

We finish by observing that the reduction can be carried out in polynomial time.

To show coNP-hardness of $\mathcal{B}^{\mathcal{G}}_F$ we remark that a 3CNF formula Φ can be treated as a $\forall^*\exists^*$QBF with no universal variables. This way, we use the previous transformation to reduce the complement of 3SAT to $\mathcal{B}^{\mathcal{G}}_F$; If I_Φ is the instance obtained in the reduction with Φ, then $\{p_\exists\}$ is a globally-optimal repair of I_Φ if and only if $\Phi \notin$ 3SAT. □

5 Pareto-optimal repairs

The high computational cost of using global optimality compels us to seek different notions of optimality that may reduce the computational complexity. The next family of repairs that we consider is closely related to $\mathcal{G}Rep$. Similarly to $\mathcal{G}Rep$, it selects a set of repairs whose compliance with the priority cannot be further improved by replacing a set of facts with a more preferred set of facts. The only difference is in the way we lift the priority relation to a preference relation of sets of facts. This notion is inspired by the construction of the Pareto optimal set of vectors [24].

Definition 11 (Pareto-optimal repairs $\mathcal{P}Rep$) Given an instance I, a set of integrity constraints F, and a priority $\succ$, an instance $I' \subseteq I$ is *Pareto optimal* w.r.t. $\succ$ and F if no nonempty subset X of facts from I' can be replaced with a nonempty set Y of facts from $I \setminus I'$ such that

$$\forall x \in X.\, \forall y \in Y.\, y \succ x \qquad (*_{\mathcal{P}})$$

and the resulting set of facts is consistent with F. $\mathcal{P}Rep$ is the family of Pareto-optimal repairs i.e., $\mathcal{P}Rep(I, F, \succ)$ is the set of all repairs of I w.r.t. F that are Pareto optimal w.r.t. $\succ$ and F.

We emphasize that the family $\mathcal{P}Rep$ selects all Pareto-optimal repairs. In general, it is, however, possible to define a family that selects only some of the Pareto-optimal repairs, or even more generally, a family that constructs a set of Pareto-optimal instances that need not be repairs. This will allow us to state some general results e.g., Theorem 3 states that any family of Pareto-optimal repairs that satisfies $\mathcal{P}1$ and $\mathcal{P}$ leads inadvertently to intractability of preferred consistent query answering. In the sequel, we fix an instance I and a set of denial constraints F, and omit them when referring to the elements of $\mathcal{P}Rep(I, F, \succ)$.

Proposition 6 *$\mathcal{P}Rep$ satisfies $\mathcal{P}1$-$\mathcal{P}4$. Also, $\mathcal{G}Rep \sqsubseteq \mathcal{P}Rep$.*

Proof $\mathcal{G}Rep \sqsubseteq \mathcal{P}Rep$ follows from Definitions 10 and 11. The arguments used to prove $\mathcal{P}1$ through $\mathcal{P}4$ are essentially the same as in Proposition 4. □

To show that $\mathcal{P}Rep \not\sqsubseteq \mathcal{G}Rep$ we recall the instance I_1 from Example 2 whose prioritized conflict graph is in Fig. 6. The repairs I_1' and I_2' are Pareto optimal but only I_1' is globally optimal.

The family of Pareto-optimal repairs can be viewed as an approximation of $\mathcal{G}Rep$ enjoying better computational properties. We believe, however, that Pareto optimality is a more cautious and conservative alternative to global optimality because it requires a stronger support from the priority to eliminate a repair. For instance, recall that I_2' from Example 2 is not globally optimal because we can replace $Mgr(Mary, \$40k, IT)$ with the more preferred $Mgr(Mary, \$50k, PR)$ and $Mgr(Ken, \$50k, PR)$ with the more preferred $Mgr(Ken, \$60k, IT)$. However, the same process can be seen as replacing $Mgr(Mary, \$40k, IT)$ with $Mgr(Ken, \$60k, IT)$ and $Mgr(Ken, \$50k, PR)$ with $Mgr(Mary, \$50k, PR)$, and neither of those swaps improves the compliance with the preference. Consequently, I_2' is Pareto optimal.

Similarly to $\mathcal{G}Rep$, $\mathcal{P}$-preferred repairs have an alternative characterization that is based on extending the priority to a pre-order on repairs.

Proposition 7 *For a given priority $\succ$ and two repairs I_1' and I_2', I_1'* Pareto dominates *I_2', denoted $I_1' \gg_{\mathcal{P}} I_2'$ if*

$$\exists y \in I_1' \setminus I_2'.\ \forall x \in I_2' \setminus I_1'.\ y \succ x. \qquad (\ast_{\mathcal{P}})$$

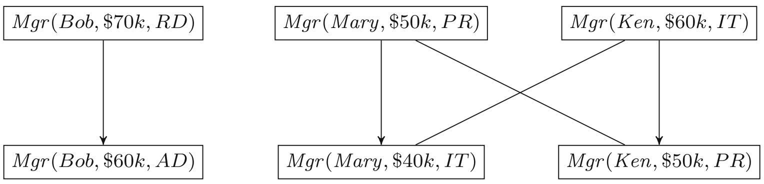

Fig. 6 Prioritized conflict graph from Example 2

The following facts hold:

(i) *a repair I' is Pareto optimal if and only if it is $\gg_{\mathcal{P}}$-maximal i.e., there is no repair I'' different from I' such that $I'' \gg_{\mathcal{P}} I'$;*
(ii) *if $\succ$ is acyclic, then so is $\gg_{\mathcal{P}}$.*

Proof

(i) We prove the contraposition i.e., I' is not Pareto optimal if and only if there exists a repair $I'' \neq I'$ such that $I'' \gg_{\mathcal{P}} I'$.
For the *if* part take $X = I' \setminus I''$ and $Y = \{y\}$, where $y \in I'' \setminus I'$ such that $\forall x \in X$ we have $y \succ x$ (it exists by $I'' \gg_{\mathcal{P}} I'$). Clearly, X and Y validate ($*_{\mathcal{P}}$) and $(I' \setminus X) \cup Y$ is consistent (as a subset of I''). Consequently, I' is not Pareto optimal. For the *only if* part take any nonempty $X \subseteq I'$ and $Y \subseteq I \setminus I'$ such that ($*_{\mathcal{P}}$) holds and $J = (I' \setminus X) \cup Y$ is consistent. Take any repair I'' that contains all facts of J. Clearly, $I' \setminus I'' = X$ and $Y \subseteq I'' \setminus I'$ so it suffices to take X and any $y \in Y$ to verify ($\star_{\mathcal{P}}$).
(ii) We observe that $I' \gg_{\mathcal{P}} I''$ implies $I' \gg_{\mathcal{G}} I''$. Thus, if $\gg_{\mathcal{P}}$ has cycles, then so does $\gg_{\mathcal{G}}$, and consequently, $\succ$. □

The class of Pareto-optimal repairs is the largest class of preferred repairs we consider in this paper. The remaining families select subsets of Pareto optimal, and in general, it is possible to consider other families that select only Pareto-optimal repairs. The following result states a rather general observation on the computational implications of introducing preferences to the framework of consistent query answers.

Theorem 3 *There exists an atomic query Q and a set F of two FDs such that for any family $\mathcal{X}Rep$ of Pareto optimal repairs satisfying $\mathcal{P}1$ and $\mathcal{P}2$ the problem of $\mathcal{X}$-consistent query answering i.e., the membership of the set*

$$\mathcal{D}^{\mathcal{X}}_{F,Q} = \left\{(I, \succ) \mid \forall I' \in \mathcal{X}Rep(I, F, \succ).\ I' \models Q\right\},$$

is coNP-hard.

Proof We show the hardness by reducing the complement of SAT to $\mathcal{D}^{\mathcal{X}}_{F,Q}$. Take then any CNF formula $\Phi = c_1 \wedge \ldots \wedge c_k$ over variables $x_1, \ldots, x_n$ and let $c_j = \ell_{j,1} \vee \ldots \vee \ell_{j,m_j}$. We assume that there are no repetitions of literals in a clause (i.e., $\ell_{j,k_1} \neq \ell_{j,k_2}$). We construct a relation instance I_Φ over the schema $R(A_1, B_1, A_2, B_2)$ in the presence of two functional dependencies $F = \{A_1 \to B_1, A_2 \to B_2\}$. The instance I_Φ consists of the following facts:

- $w_i = R(i, 1, i, 1)$ corresponding to the positive valuation of x_i (for $i \in \{1, \ldots, n\}$),
- $\bar{w}_i = R(i, -1, -i, 1)$ corresponding to the negative valuation of x_i (for every $i \in \{1, \ldots, n\}$),
- $d_j = R(n + j, 1, 0, 1)$ corresponding to the clause c_j (for every $j \in \{1, \ldots, m\}$),
- $v_i^j = R(n + j, 1, -i, 0)$ encoding the use of x_i in the clause c_j (for any $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ such that c_j uses x_i),

- $\bar{v}_i^j = R(n+j, 1, i, 0)$ encoding the use of $\neg x_i$ in the clause c_j (for any $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ such that c_j uses $\neg x_i$),
- $b = R(0, 0, 0, 0)$ corresponding to the formula Φ.

The constructed priority $\succ_\Phi$ is the minimal priority of I_Φ w.r.t. F such that:

$$\bar{w}_i \succ_\Phi v_i^j, \qquad v_i^j \succ_\Phi d_j, \qquad d_j \succ_\Phi b,$$
$$w_i \succ_\Phi \bar{v}_i^j, \qquad \bar{v}_i^j \succ_\Phi d_j.$$

Figure 7 presents prioritized conflict graph obtained from the formula $\Phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4 \vee x_5)$. The query we consider is $Q = \neg b$. We claim that

$$(I_\Phi, \succ_\Phi) \in \mathcal{D}^{\mathcal{X}}_{F,Q} \iff \forall I' \in \mathcal{X}Rep\,(I_\Phi, F, \succ_\Phi)\,.\, b \notin I' \iff \Phi \notin \text{SAT}.$$

For the *if* part, suppose there exists a repair $I' \in \mathcal{X}Rep(I_\Phi, F, \succ_\Phi)$ such that $b \in I'$. Obviously, for every $j \in \{1, \ldots, m\}$ the fact d_j does not belong to I'. Also, for every j at least one fact neighboring to d_j, other than b, is present in I', or otherwise I' is not a Pareto-optimal repair. Similarly, I' has either w_i or $\bar{w}_i$ for every $i \in \{1, \ldots, n\}$, and hence, the following valuation is properly defined:

$$V(x_i) = \begin{cases} \textbf{true} & \text{if } w_i \in I', \\ \textbf{false} & \text{if } \bar{w}_i \in I'. \end{cases}$$

We claim that $V \models \Phi$. Suppose otherwise and take any clause c_j unsatisfied by V. Let $x \neq b$ be the fact neighboring to d_j that is present in I'. W.l.o.g. we can assume that $x = \bar{v}_{j,i_0}$ for some i_0 and then $\neg x_{i_0}$ is a literal of c_j. Also then, w_{i_0} does not belong to I' and so $V(x_{i_0}) = \textbf{false}$. This implies that $V \models \neg x_{i_0}$ and $V \models c_j$; a contradiction.

For the *only if* part, suppose there exists a valuation V such that $V \models \Phi$ and consider the following instance

$$I' = \{w_i \mid V(x_i) = \textbf{true}\} \cup \{\bar{w}_i \mid V(x_i) = \textbf{false}\} \cup$$
$$\left\{v_i^j \mid V(x_i) = \textbf{true}\right\} \cup \left\{\bar{v}_i^j \mid V(x_i) = \textbf{false}\right\} \cup \{b\}.$$

First, we note that I' is a repair and a Pareto-optimal one. Next, we show that $I' \in \mathcal{X}Rep(I_\Phi, F, \succ_\Phi)$. To prove this consider the following priority $\succ' = \succ_\Phi \cup \{(v_i, \bar{v}_i) \mid V(x_i) = \textbf{true}\} \cup \{(\bar{v}_i, v_i) \mid V(x_i) = \textbf{false}\}$. It can be easily verified that I' is the only Pareto-optimal repair of I_Φ w.r.t. F and $\succ'$. Since $\mathcal{X}Rep$ satisfies $\mathcal{P}1$, we get $I' \in \mathcal{X}Rep(I_\Phi, F, \succ')$. Note that $\succ'$ is an extension of $\succ_\Phi$ and thus I' belongs to

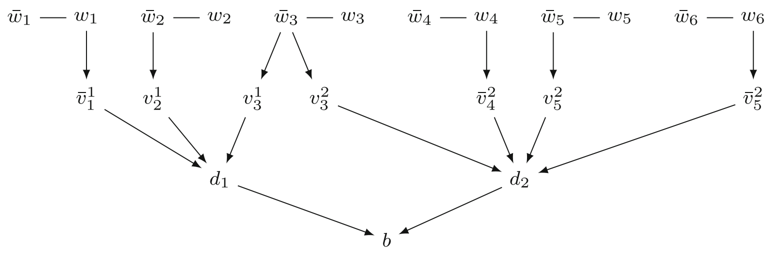

Fig. 7 The prioritized conflict graph for $\Phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4 \vee x_5)$

$\mathcal{X}Rep(I_\Phi, F, \succ)$ by $\mathcal{P}2$. Finally, we observe that $b \in I'$ which implies that **true** is not an $\mathcal{X}$-preferred consistent query answer to Q in I_Φ w.r.t. F and $\succ_\Phi$; a contradiction.

We finish the proof with the observation that the described reduction requires time polynomial in the size of the formula Φ. □

We also present an alternative characterization of Pareto-optimal repairs that yields a tractable procedure for repair checking.

Lemma 3 *A repair I' is not Pareto optimal w.r.t. $\succ$ if and only if there exists a fact $y \in I \setminus I'$ such that for every conflict C in $I' \cup \{y\}$ there is $x \in C$ such that $y \succ x$.*

Proof For the *if* part, let $C_1, \ldots, C_k$ be all conflicts in $I' \cup \{y\}$ and x_i be the element of C_i such that $y \succ x_i$ (for $i \in \{1, \ldots, k\}$). Clearly, $(I' \setminus \{x_1, \ldots, x_k\}) \cup \{y\}$ is consistent, which shows that I' is not Pareto optimal.

For the *only if* part, take any nonempty X and Y such that $(I' \setminus X) \cup Y$ is consistent and $\forall y \in Y.\ \forall x \in X.\ y \succ x$. Fix any $y \in Y$ and take any conflict C in $I' \cup \{y\}$. Clearly, C contains an element x of X since $(I' \setminus X) \cup Y$ is consistent. Naturally $y \succ x$. □

Corollary 3 *$\mathcal{P}$-preferred repair checking is in LOGSPACE and $\mathcal{P}$-preferred consistent query answering is coNP-complete.*

Proof We observe that to check the condition of Lemma 3 we need to iterate over $I \setminus I'$ which can be accomplished with two pointers: one to iterate over I and the other to scan I'. Recall that a conflict is a set of facts and its cardinality is bounded by the size of F which is assumed to be a constant parameter. Hence, we can iterate over all conflicts of I' (extended with one fact) using a constant number of pointers scanning I'. Consequently, $\mathcal{P}$-preferred repair checking is in LOGSPACE. $\mathcal{D}^{\mathcal{P}}_{F,Q}$ belongs to coNP from the definition of $\mathcal{P}$-preferred consistent query answers and is coNP-complete by Theorem 3. □

Now, we investigate a sound and complete algorithm for computing Pareto-optimal repairs. First, we observe that it is possible to use an algorithm similar to Algorithm 2, starting with an arbitrary repair and attempting to iteratively improve its compliance with the priority until a Pareto-optimal repair is reached. While checking Pareto optimality can be done in polynomial time, we note that the sequence of repairs, constructed in Example 4, of exponential length is also a $\gg_{\mathcal{P}}$-chain. Consequently, such an algorithm may require an exponential number of iterations to obtain a Pareto-optimal repair.

We propose a simpler approach where we construct an arbitrary repair and if it is not Pareto optimal we discard it and construct a completion-optimal repair using Algorithm 4 presented in the next section. Completion-optimal repairs constitute a subset of Pareto-optimal repairs and thus if the algorithm fails to construct a Pareto-optimal repair in the first stage, then the repair constructed in the second stage is Pareto optimal. Consequently, Algorithm 3 is sound. We recall that Algorithm 1 is complete, i.e it may return any repair, in particular, any Pareto-optimal repair. If the repair constructed in step 1 of Algorithm 3 is Pareto optimal, then this repair is returned. Consequently, the algorithm is complete. Finally, it works in polynomial time since checking Pareto optimality is in LOGSPACE and Algorithms 1 and 4 work in polynomial time.

Algorithm 3 Constructing a Pareto-optimal repair of I w.r.t. F and $\succ$

```
1: construct a repair $I'$ of $I$   /*Algorithm 1*/
2: if $I'$ is Pareto optimal w.r.t. $\succ$ then
3:     return $I'$
4: else
5:     return any completion-optimal repair of $I$ w.r.t. $\succ$   /*Algorithm 4*/
```

Proposition 8 *Algorithm* 3 *is a sound and complete algorithm constructing Pareto-optimal repairs. It works in time polynomial in the size of the input instance and the priority relation.*

6 Completion-optimal repairs

The last family of preferred repairs is based on a notion of optimality different from global and Pareto optimality and intuitively can be described as follows. When repairing a database with a priority that is not total and resolving a conflict that is not prioritized, we commit to a particular prioritization of this conflict. In this view, constructing a repair that conforms to a given priority is equivalent to constructing a total extension of that priority such that the constructed repair is the only repair globally optimal w.r.t. the total priority. We remark that this notion is quite robust as it remains identical if we replace in it global optimality by Pareto optimality. The same holds for all results stated in this section. This is because $\mathcal{G}Rep$ and $\mathcal{P}Rep$ coincide for total priorities by $\mathcal{G}Rep \sqsubseteq \mathcal{P}Rep$ and $\mathcal{P}4$ for $\mathcal{P}Rep$ and $\mathcal{G}Rep$. Another motivation for the family of repairs presented in this section is a fairly intuitive and natural repairing algorithm which we present later on.

Definition 12 (Completion-optimal repairs $\mathcal{C}Rep$) Given an instance I, a set of denial constraints F, and a priority $\succ$, an instance $I' \subseteq I$ is *completion optimal* w.r.t. $\succ$ and F if and only if there exists a total priority $\succ' \supseteq \succ$ such that I' is globally optimal w.r.t. $\succ'$ and F. $\mathcal{C}Rep$ is the family of completion-optimal repairs i.e., $\mathcal{C}Rep(I, F, \succ)$ is the set of all repairs of I w.r.t. F that are completion optimal w.r.t. $\succ$ and F.

We remark that $\mathcal{C}Rep$ selects all completion-optimal repairs and that it is possible to consider families that select only some completion-optimal repairs. We fix an instance I and a set of denial constraints F, and omit them when referring to the elements of $\mathcal{C}Rep(I, F, \succ)$.

Example 5 Consider the schema of one relation name $R(A, B, C, D)$ with a set of two functional dependencies $F_4 = \{R : A \to B, R : C \to D\}$. Take the following instance

$$I_4 = \{R(1, 1, 1, 1), R(1, 2, 1, 2), R(1, 3, 0, 0), R(0, 0, 1, 3)\}$$

and the following priority relation

$$\succ_4 = \{(R(1,1,1,1), R(1,3,0,0)), (R(1,2,1,2), R(0,0,1,3))\}.$$

The corresponding prioritized conflict graph is presented in Fig. 8. The instance I_4 has 3 repairs:

$$I'_1 = \{R(1,1,1,1)\}, \quad I'_2 = \{R(1,2,1,2)\}, \quad I'_3 = \{R(1,3,0,0), R(0,0,1,3)\}.$$

We note that all three repairs are globally optimal w.r.t. $\succ_4$. The repairs I'_1 and I'_2 are completion optimal as witnessed by the following total extensions of $\succ_4$ ($\succ'_4$ for I'_1 and $\succ''_4$ for I'_2):

$$\succ'_4 = \succ_4 \cup \{(R(1,1,1,1), R(0,0,1,3)), (R(1,2,1,2), R(1,3,0,0)), \\ (R(1,1,1,1), R(1,2,1,2))\}$$

$$\succ''_4 = \succ_4 \cup \{(R(1,1,1,1), R(0,0,1,3)), (R(1,2,1,2), R(1,3,0,0)), \\ (R(1,2,1,2), R(1,1,1,1))\}.$$

On the other hand, there is no total extension of $\succ_4$ for which the repair I'_3 is globally optimal, and hence, I'_3 is not completion optimal. □

It is an open question whether there exists an intuitive definition of a pre-order on repairs whose maximal elements are exactly completion-optimal repairs. We show, however, the family of completion-optimal repairs is the smallest family of globally-optimal repairs that satisfies the properties $\mathcal{P}1$ and $\mathcal{P}2$. Notice that $\mathcal{G}Rep$ is only one of the possible families of globally-optimal repairs satisfying $\mathcal{P}1$ and $\mathcal{P}2$. $\mathcal{C}Rep$ is another one.

Lemma 4 *$\mathcal{C}Rep \sqsubseteq \mathcal{X}Rep$ for every family $\mathcal{X}Rep$ of globally-optimal repairs that satisfies $\mathcal{P}1$ and $\mathcal{P}2$. In other words, a repair I' is completion optimal w.r.t. F and $\succ$ if and only if $I' \in \mathcal{X}Rep(I, F, \succ)$ for every family $\mathcal{X}Rep$ of globally-optimal repairs that satisfies $\mathcal{P}1$ and $\mathcal{P}2$.*

Proof For the *only if* part, observe that by $\mathcal{P}4$ for $\mathcal{G}Rep$ I' is the only globally-optimal repair w.r.t. $\succ$. Consequently, $I' \in \mathcal{X}Rep(I, F, \succ')$ for any family $\mathcal{X}Rep$ satisfying $\mathcal{P}1$. Moreover, $I' \in \mathcal{X}Rep(I, F, \succ)$ because $\mathcal{X}Rep$ satisfies $\mathcal{P}2$ and $\succ \subseteq \succ'$. Thus, I' is a completion-optimal repair of I w.r.t. F and $\succ$.

For the *if* part, suppose $\succ$ has no acyclic total extension $\succ'$ for which I' is globally optimal w.r.t. $\succ'$. Consider the following family of globally-optimal repairs

$$\mathcal{X}Rep\left(I^o, F^o, \succ^o\right) = \begin{cases} \mathcal{G}Rep\left(I^o, F^o, \succ^o\right) \setminus \{I'\} & \text{if } \succ^o \supseteq \succ, I^o = I, \text{and } F^o = F, \\ \mathcal{G}Rep\left(I^o, F^o, \succ^o\right) & \text{otherwise.} \end{cases}$$

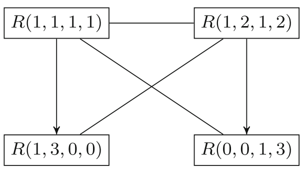

Fig. 8 The prioritized conflict graph $G(I_4, F_4, \succ_4)$

It can be easily seen that $\mathcal{X}Rep$ satisfies $\mathcal{P}1$ and $\mathcal{P}2$. We observe that $I' \notin \mathcal{X}Rep(I, F, \succ)$. Consequently, I' is not a completion-optimal repair of I w.r.t. F and $\succ$. □

Proposition 9 *$\mathcal{C}Rep$ satisfies $\mathcal{P}1$-$\mathcal{P}4$ and $\mathcal{C}Rep \sqsubseteq \mathcal{G}Rep$.*

Proof $\mathcal{C}Rep \sqsubseteq \mathcal{G}Rep$ because $\mathcal{G}Rep$ is a family of globally-optimal repairs that satisfies both $\mathcal{P}1$ and $\mathcal{P}2$ (cf. Proposition 4).

$\mathcal{P}1$ follows from the definition of completion-optimal repairs and the observation that any priority $\succ$ can be extended to some total $\succ'$ (the same argument as in the proof of Proposition 4). Therefore, $\varnothing \neq \mathcal{G}Rep(I, F, \succ') \subseteq \mathcal{C}Rep(I, F, \succ)$ by $\mathcal{P}4$ for $\mathcal{G}Rep$. $\mathcal{P}2$ follows directly from Lemma 4.

To show $\mathcal{P}3$ we take an arbitrary repair I' and construct a priority $\succ$ such that I' is globally optimal w.r.t. $\succ$. For that we take any total ordering $\succ_1$ of I' and any total ordering of $\succ_2$ of $I \setminus I'$. We obtain $\succ$ by a diligent composition of $\succ_1$ with $\succ_2$:

$$R(t) \succ R'\left(t'\right) \iff \begin{cases} R(t) \succ_1 R'\left(t'\right) & \text{if } R(t), R'\left(t'\right) \in I', \\ \textbf{true} & \text{if } R(t) \in I' \text{ and } R'\left(t'\right) \in I \setminus I', \\ R(t) \succ_2 R'\left(t'\right) & \text{if } R(t), R'\left(t'\right) \in I \setminus I', \\ \textbf{false} & \text{if } R(t) \in I \setminus I' \text{ and } R'\left(t'\right) \in I', \end{cases}$$

for any two neighboring facts $R(t)$ and $R'(t')$ ($R(t) \not\succ R'(t')$ if $R(t)$ and $R'(t')$ are not neighboring). Clearly, $\succ$ is acyclic since it is based on the acyclic components $\succ_1$ and $\succ_2$, and we add an element $(R(t), R'(t'))$ only if $R(t) \in I'$ and $R'(t') \notin I'$. Naturally, $\succ$ is a total priority. It is also easy to verify that I' is globally optimal w.r.t. $\succ$. $\mathcal{P}4$ follows from $\mathcal{C}Rep \sqsubseteq \mathcal{G}Rep$, $\mathcal{P}4$ for $\mathcal{G}Rep$, and $\mathcal{P}1$ for $\mathcal{C}Rep$ proved above. □

Completion-optimal repairs can be also characterized as exactly those repairs that can be obtained with an iterative accumulation of facts selected with the *winnow operator* [11]:

$$\omega_{\succ}(I) = \left\{R(t) \in I \mid \nexists R'(t') \in I.\ R'(t') \succ R(t)\right\}.$$

Theorem 4 *Algorithm 4 is a sound and complete algorithm constructing completion-optimal repairs. It works in time polynomial in the size of the input instance and the priority relation.*

Algorithm 4 Constructing a completion-optimal repair

```
1: $I^o \leftarrow I$
2: $J \leftarrow \varnothing$
3: while $\omega_{\succ}(I^o) \neq \varnothing$ do
4:     choose $R(t) \in \omega_{\succ}(I^o)$
5:     $I^o \leftarrow I^o \setminus \{R(t)\}$
6:     if $J \cup \{R(t)\} \models F$ then
7:         $J \leftarrow J \cup \{R(t)\}$
8: return $J$
```

Proof We observe that the instance resulting from an execution of Algorithm 4 can be associated with the *sequence of choices* made in line 4 during the execution. We also observe that this sequence is an ordering of the facts of the original instance I.

To show soundness, we take an instance I' obtained with the sequence of choices $x_1, \ldots, x_n$. We show that I' is completion optimal by extending $\succ$ to a total priority $\succ'$ for which I' is globally optimal. The priority $\succ'$ is defined as

$$x_i \succ' x_j \iff x_i \text{ and } x_j \text{ are neighboring and } i < j.$$

Clearly, $\succ'$ is acyclic and a total priority. We also observe that $\succ \subseteq \succ'$ because $x_i \succ x_j$ implies that $i < j$ i.e., x_i is selected before x_j and the choices are constrained by $\omega_\succ$.

To show that I' is globally optimal w.r.t. $\succ'$ take any $X \subseteq I'$ and any $Y \subseteq I \setminus I'$ such that $(I' \setminus X) \cup Y$ is consistent. Now, take any $x_j \in Y$ and observe that adding x_j to the instance being created by Algorithm 4 must have been prevented by some conflict $\{x_{i_1}, \ldots, x_{i_k}, x_j\}$ with the facts added previously i.e., $i_\ell < j$ for $\ell \in \{1, \ldots, k\}$. Consequently, $x_{i_\ell} \succ' x_j$ for $\ell \in \{1, \ldots, k\}$. We observe that at least one of $x_{i_1}, \ldots, x_{i_k}$ must be present in X since Y contains x_j, I' contains $x_{i_1}, \ldots, x_{i_k}$, and $(I' \setminus X) \cup Y$ is consistent. Thus, $x_j \not\succ' x_\ell$ for some $x_\ell \in X$, I' is globally optimal w.r.t. $\succ'$, and by $\mathcal{P}2$ for $\mathcal{G}Rep$ we get that I' is globally optimal w.r.t. $\succ$.

To show completeness, we take a completion-optimal repair I' and the total priority $\succ'$ for which I' is globally optimal and use $\succ'$ to construct a valid sequence of choices yielding I'. Naturally, the same choice sequence is valid for an execution with $\succ$ because $\succ'$ extends $\succ$.

Take an execution of Algorithm 4 on I with $\succ'$ that constructs some instance I'' with the sequence of choices $x_1, \ldots, x_n$. Note that if x_i and x_j are neighboring, then $x_i \succ' x_j$ if and only if $i < j$. Suppose that $I'' \neq I'$ and take the minimal index i of the element x_i on which I' and I'' differ. Note that $I' \cap \{x_1, \ldots, x_{i-1}\} = I'' \cap \{x_1, \ldots, x_{i-1}\}$ and either $x_i \in I'$ and $x_i \notin I''$, or $x_i \notin I'$ and $x_i \in I''$. The first case is not possible because Algorithm 4 would have discarded x_i only if there had been a conflict involving x_i and some facts of $I'' \cap \{x_1, \ldots, x_{i-1}\}$. Then, however, the same conflict would have been included in I' i.e., I' would have not been consistent. Suppose then, $x_i \notin I'$ and $x_i \in I''$. Let $C_1, \ldots, C_k$ be all conflicts present in $I' \cup \{x_i\}$ w.r.t. F, and since $I' \cup \{x_i\}$ is not consistent, there exists at least one conflict in $I' \cup \{x_i\}$. Naturally, $I' \cap \{x_1, \ldots, x_{i-1}\} \cup \{x_i\}$ is consistent, and thus for every $j \in \{1, \ldots, k\}$ the conflict C_j contains a fact x_{i_j} such that $i_j > i$. Let $X = \{x_{i_1}, \ldots, x_{i_k}\}$ and $Y = \{x_i\}$, and observe that $(I' \setminus X) \cup Y$ is consistent. Moreover, X and Y satisfy $(*_\mathcal{G})$ (Definition 10) since $i_j > i$ implies that $x_i \succ x_{i_j}$. Consequently, I' is not globally optimal; a contradiction. □

Corollary 4 *$\mathcal{C}$-preferred repair checking is in PTIME and $\mathcal{C}$-preferred consistent query answering is coNP-complete.*

Proof To check if a repair I' is completion optimal we use Algorithm 4 to simulate the construction of I' by restricting the choice in line 4 to facts $\omega_\succ(J) \cap I'$. It can be easily shown that the repair I' is completion optimal if and only if such a simulation can be performed successfully (i.e., it produces I'). Naturally, $\mathcal{D}^{\mathcal{C}}_{F,Q}$ belongs to coNP and its coNP-completeness follows from Theorem 3. □

The exact complexity of $\mathcal{C}$-preferred repair checking, namely whether it is PTIME-complete or in LOGSPACE, remains an open question.

The introduced families of preferred repairs create a hierarchy:

$$\mathcal{C}Rep \sqsubseteq \mathcal{G}Rep \sqsubseteq \mathcal{P}Rep.$$

Recall from the previous section that $\mathcal{P}Rep \neq \mathcal{G}Rep$ (cf. Fig. 6). We note that in Example 5 all repairs are globally optimal but only I'_1 and I'_2 are completion optimal which shows that $\mathcal{C}Rep \neq \mathcal{G}Rep$. Thus, the hierarchy is proper. We observe, however, that under certain conditions this hierarchy collapses.

Proposition 10 *$\mathcal{P}Rep$, $\mathcal{G}Rep$, and $\mathcal{C}Rep$ coincide under one of the following conditions:*

(i) *the set of constraints F consists of one key dependency only;*
(ii) *the priority $\succ$ can be extended to acyclic priorities only.*

Moreover, $\mathcal{G}Rep$ and $\mathcal{C}Rep$ coincide if

(iii) *the set of constraints F consists of one functional dependency only.*

Proof For (i) to show that $\mathcal{P}Rep \sqsubseteq \mathcal{C}Rep$ in the presence of exactly one key dependency, we use the fact that the conflict graph is a union of pairwise disjoint cliques and every repair consists of exactly one element selected from each clique.

We fix an instance I, a key dependency F, and a priority $\succ$. Let $C_1, \ldots, C_n$ be the cliques of $G(I, F)$. Take any $I' \in \mathcal{P}Rep(I, F, \succ)$ and let $R_1(t_1), \ldots, R_n(t_n)$ be the elements of I' such that $R_i(t_i) \in C_i$. Since I' is Pareto optimal, then for every i there is no $y \in C_i \setminus \{R(t_i)\}$ such that $y \succ R_i(t_i)$, and consequently, $R_i(t_i) \in \omega_{\succ}(C_i)$. Hence, $R_1(t_1), \ldots, R_n(t_n)$ is a proper choice sequence for Algorithm 4. Finally, we observe that if the fact $R_i(t_i)$ has been added to the constructed repair, then none of the facts of $C_i \setminus \{R_i(t_i)\}$ can be further added.

For (ii) We take any $I' \in \mathcal{P}Rep(I, F, \succ)$ and construct a total extension $\succ'$ of $\succ$ by prioritizing in favor of I' all conflicts unprioritized by $\succ$ i.e., $\succ'$ is any total priority such that for any $x \in I'$ and any y conflicting with x if $y \not\succ x$ then $x \succ' y$. Since $\succ$ can be extended to acyclic orientations only, $\succ'$ is acyclic. Clearly, I' is a Pareto-optimal repair w.r.t. $\succ'$ and a unique one by $\mathcal{P}4$ for $\mathcal{P}Rep$. Therefore $I' \in \mathcal{C}Rep(I, F, \succ')$ and by $\mathcal{P}2$ we get $I' \in \mathcal{C}Rep(I, F, \succ)$.

For (iii) we assume a single relation name R with the functional dependency $X \to Y$ and use the notions of X-cluster and (X, Y)-cluster (Section 2.3, page 8). Let the instance I be the union of the X-clusters $C_1, \ldots, C_n$. Take any globally-optimal repair I' and let it be the union of the (X, Y)-clusters $D_1, \ldots, D_n$ ($D_i \subseteq C_i$ for every $i \in \{1, \ldots, n\}$). By global optimality of I' we have that for every $i \in \{1, \ldots, n\}$

$$\exists R_i(t_i) \in D_i . \forall y \in C_i \setminus D_i . y \not\succ R_i(t_i).$$

Therefore, Algorithm 4 can perform the first n iterations with a choice sequence beginning with $R_i(t_1), \ldots, R_n(t_n)$. Because $n(R_i(t_i)) = C_i \setminus D_i$ and elements of D_i conflict only with elements of $C_i \setminus D_i$, the remaining choices can consist of any ordering of $(D_1 \setminus \{R_i(t_1)\}) \cup \ldots \cup (D_n \setminus \{R_i(t_n)\})$. Consequently, I' is a result of Algorithm 4. □

We note that the conditions are sufficient but not necessary e.g., the hierarchy trivially collapses for any set of denial constraints and an empty priority relation.

7 Tractable case

The intractability proofs for consistent query answering use at least 2 FDs. Next, we investigate the case when only one FD is present. We begin by considering queries that are conjunctions of ground literals, and next, we generalize this approach to arbitrary ground queries.

We observe that if only functional dependencies are considered, facts can create conflicts only with facts of the same relation, and therefore, we can limit our consideration to a schema consisting of one relation name only. Consequently, we assume a single relation name R with the FD $R : X \rightarrow Y$ and use the notions of X-cluster and (X, Y)-cluster (Section 2.3). We fix an instance I and a priority $\succ$. For every fact $R(t) \in I$, by $C_{R(t)}$ we denote the X-cluster to which the fact $R(t)$ belongs to and by $D_{R(t)}$ we denote its (X, Y)-cluster. We also fix the query

$$\Phi = R(t_1) \wedge \ldots \wedge R(t_k) \wedge \neg R(t_{k+1}) \wedge \ldots \wedge \neg R(t_m).$$

We assume that the facts $R(t_1), \ldots, R(t_k)$ belong to I; otherwise there is no repair satisfying Φ. We assume that also the facts $R(t_{k+1}), \ldots, R(t_n)$ belong to I; otherwise we can remove any negative literal from Φ if it is not in I. We also recall that in the presence of one FD only, the family of globally-optimal and completion-optimal repairs coincide (Proposition 10).

Lemma 5 *A (completion-) globally-optimal repair I' satisfying Φ exists if and only if the following conditions are satisfied:*

(i) $\{R(t_1), \ldots, R(t_k)\}$ *is conflict-free;*
(ii) $\{D_{R(t_1)}, \ldots, D_{R(t_k)}\} \cap \{D_{R(t_{k+1})}, \ldots, D_{R(t_m)}\} = \emptyset$;
(iii) $D_{R(t_j)} \cap \omega_{\succ}(C_{R(t_j)}) \neq \emptyset$ *for every* $j \in \{1, \ldots, k\}$.
(iv) $\omega_{\succ}(C_{R(t_j)}) \setminus (D_{R(t_{k+1})} \cup \ldots \cup D_{R(t_n)}) \neq \emptyset$ *for every* $j \in \{k+1, \ldots, m\}$.

Proof For the *only if* part, we take any globally-optimal repair I' satisfying Φ. (i) and (ii) are trivially satisfied. Assume that I' is the result of Algorithm 4 with a choice sequence $R(s_1), \ldots, R(s_\ell)$. Take any $j \in \{1, \ldots, k\}$ and let j' be the smallest index of a fact from $C_{R(t_j)}$ in the sequence. Clearly, $R(s_{j'}) \in I'$. Since $R(t_j)$ also belongs to I', both $R(s_{j'})$ and $R(t_j)$ belong to the same (X, Y)-cluster i.e., $R(s_{j'}) \in D_{R(t_j)}$. Also prior to selecting $R(s_{j'})$ the temporary instance I^o contains $C_{R(t_j)}$. Therefore $R(s_{j'}) \in \omega_{\succ}(C_{R(t_j)})$ which proves (iii).

We show (iv) similarly. For any $j \in \{k+1, \ldots, m\}$ let j' be the smallest index of a fact from $C_{R(t_j)}$ in the sequence of choices used to construct I'. Prior to making the choice $R(s_{j'})$ the temporary instance I^o contains $C_{R(t_j)}$, $R(s_{j'}) \in \omega_{\succ}(C_{R(t_j)})$, and $R(s_{j'})$ does not belong to any of $D_{R(t_{k+1})}, \ldots, D_{R(t_n)}$.

For the *if* part, we construct I' using Algorithm 4 with a choice sequence $R(s_1), \ldots, R(s_\ell)$ defined as follows. By (i) and (iii), for $j \in \{1, \ldots, k\}$ the choice $R(s_j)$ is any fact from $D_{R(t_j)} \cap \omega_{\succ}(C_{R(t_j)})$. By (ii) and (iv), for any $j \in \{k+1, \ldots, m\}$ the choice $R(s_j)$ is any fact from $\omega_{\succ}(C_{R(t_j)}) \setminus (D_{R(t_{k+1})} \cup \ldots \cup D_{R(t_n)})$. The remaining choices $R(t_j)$ for $j \in \{m+1, \ldots, \ell\}$ are selected in an arbitrary way. We observe that the first k steps guarantees that the facts $R(t_1), \ldots, R(t_k)$ belong to the repair instance

I' (possibly placed there in later consecutive steps) and that I' does not contain any of the facts $R(t_{k+1}), \ldots, R(t_m)$. □

Lemma 6 *A Pareto-optimal repair I' satisfying Φ exists if and only if the following conditions are satisfied:*

(i) $\{R(t_1), \ldots, R(t_k)\}$ *is conflict-free;*
(ii) $\{D_{R(t_1)}, \ldots, D_{R(t_k)}\} \cap \{D_{R(t_{k+1})}, \ldots, D_{R(t_m)}\} = \emptyset$*;*
(iii) *for every $j \in \{1, \ldots, k\}$, for every fact $R(t) \in C_{R(t_j)} \setminus D_{R(t_j)}$ there exists $R(t') \in D_{R(t_j)}$ such that $R(t) \not\succ R(t')$.*
(iv) *for every $j \in \{k+1, \ldots, m\}$ there exists an (X, Y)-cluster D of $C_{R(t_j)}$ different from $D_{R(t_{k+1})}, \ldots, D_{R(t_m)}$ such that for every $t \in D_{R(t_{k+1})} \cup \ldots \cup D_{R(t_m)}$, there exists $R(t') \in D$ such that $R(t) \not\succ R(t')$.*

Proof For the *only if* part, (i) and (ii) are trivially implied by $I' \models \Phi$. To show (iii) and (iv) we observe that a Pareto-optimal repair contains exactly one Pareto-optimal (X, Y)-cluster for every X-cluster. For clusters $C_{R(t_1)}, \ldots, C_{R(t_k)}$ this together with the fact that $\{R(t_1), \ldots, R(t_k)\} \subseteq I'$ implies (iii). For clusters $C_{R(t_{k+1})}, \ldots, C_{R(t_m)}$ this together with the fact that $\{R(t_{k+1}), \ldots, R(t_m)\} \cap I' = \emptyset$ implies (iv).

For the *if* part, we construct the repair I' by selecting an (X, Y)-cluster from every X-cluster. Because Pareto optimality is defined in terms of neighboring facts and for one FD conflicts can be present only inside an X-cluster, to show that the repair I' is Pareto optimal it is enough to show that for every X-cluster the selected (X, Y)-cluster is Pareto optimal (among all (X, Y)-clusters in the X-cluster).

For X-clusters $C_{R(t_1)}, \ldots, C_{R(t_k)}$ we select $D_{R(t_1)}, \ldots, D_{R(t_k)}$ resp. We note that by (i) the (X, Y)-clusters belong to different X-clusters and by (ii) we do not include any of the facts $R(t_{k+1}), \ldots, R(t_m)$. Pareto optimality is implied by (iii). For X-clusters $C_{R(t_{k+1})}, \ldots, C_{R(t_m)}$ we select the (X, Y)-clusters as described in (iv). Pareto optimality of those clusters is also implied by (iv). For an X-cluster other than $C_1, \ldots, C_m$ we select any (X, Y)-cluster that is Pareto optimal (for the X-cluster). Since all selected (X, Y)-clusters are Pareto optimal, the instance I' is a Pareto-optimal repair such that $I' \models \Phi$. □

Theorem 5 *If the set of integrity constraints contains at most one functional dependency per relation name and no other constraints, then computing preferred consistent answers to quantifier-free queries is in PTIME for $\mathcal{P}Rep$, $\mathcal{G}Rep$, and $\mathcal{C}Rep$.*

Proof We adopt the algorithm from [13]. We assume that the query Ψ is in CNF i.e., $\Psi = \Psi_1 \wedge \ldots \wedge \Psi_n$. By definition **true** is not a preferred consistent query answer to Ψ if and only if there exists a preferred repair I' and $i \in \{1, \ldots, n\}$ such that $I' \not\models \Psi_i$ i.e., $I' \models \neg\Psi_i$. Note that the negation of Ψ_i is a conjunction of literals. Consequently, the algorithm attempts to verify for every $i \in \{1, \ldots, n\}$ whether a preferred repair satisfying $\neg\Psi_i$ exists using tests from Lemma 5 or 6 (depending on the class of preferred repairs considered) If this condition is satisfied for some $i \in \{1, \ldots, n\}$, then **true** is not the preferred consistent answer to Ψ. On the other hand, **true** is the preferred consistent answer if the test fails for every $i \in \{1, \ldots, n\}$. Finally, we remark that the test can be performed in time polynomial in the size of the instance I (the size of the query is assumed to be a constant) □

8 Related work

We limit our discussion to the work on using priorities to maintain consistency and facilitate resolution of conflicts.

The first article to notice the importance of priorities in information systems is [16]. There, the problem of conflicting updates in (propositional) databases is solved in a manner similar to $\mathcal{C}Rep$. The considered priorities are transitive, which is more restrictive than acyclicity and does not bring any computational benefits in our framework: our reductions can be modified to use only transitive priorities. Brewka [9] is another example of $\mathcal{C}Rep$-like prioritized conflict resolution of first-order theories. The basic framework is defined for priorities which are weak orders. A partial order is handled by considering every extension to weak order. This approach also assumes transitivity of the priority.

In the context of logic programs, priorities among rules can be used to handle inconsistent logic programs (where rules imply contradictory facts). More important rules are satisfied, possibly at the cost of violating less important ones. In a manner analogous to Proposition 3, [30] lifts a total order on rules to a preference on (extended) answers sets. When computing answers only maximally preferred answers sets are considered.

In [22], Grosof presents a simpler approach to handling of inconsistent logic programs with user priorities. Conflicting facts are removed from the model unless the priority specifies how to resolve the conflict. Because only programs without disjunction are considered, this approach always returns exactly one model of the input program. Constructing preferred repairs in a corresponding fashion (by removing all conflicts unless the priority indicates a resolution) would similarly return exactly one database instance (fulfillment of $\mathcal{P}1$ and $\mathcal{P}4$). However, if the priority is not total, the returned instance is not a repair and therefore $\mathcal{P}5$ is not satisfied. Such an approach leads to a loss of (disjunctive) information and does not satisfy $\mathcal{P}2$ and $\mathcal{P}3$.

In [10], Caroprese et al. propose the framework of *conditioned active integrity constraints*, which allows the user to specify the way some of the conflicts created with a constraint can be resolved. This framework satisfies properties $\mathcal{P}1$ and $\mathcal{P}2$ but not $\mathcal{P}3$ and $\mathcal{P}4$. The authors also describe how to translate conditioned active integrity constraints into a prioritized logic program [27], whose preferred models correspond to maximally preferred repairs.

In [26], Motro et al. use ranking functions on facts to resolve conflicts by taking only the fact with highest rank and removing others. This approach constructs a unique repair under the assumption that no two different facts are of equal rank (satisfaction of $\mathcal{P}4$). If this assumption is not satisfied and the facts contain numeric values, a new value, called the fusion, can be calculated from the conflicting facts (then, however, the constructed instance is not necessarily a repair in the sense of Definition 3 which means a possible loss of information).

In [21], Greco et al. study a different approach based on ranking is studied. The authors consider polynomial functions that are used to rank repairs. When computing preferred consistent query answers, only repairs with the highest rank are considered. The properties $\mathcal{P}2$ and $\mathcal{P}5$ are trivially satisfied, but because this form of preference information does not have natural notions of extensions and maximality, it is hard to discuss postulates $\mathcal{P}3$ and $\mathcal{P}4$. Also, the preference among repairs in this method is not based on the way in which the conflicts are resolved.

In [20], Greco and Lenbo study an approach where the user has a certain degree of control over the way the conflicts are resolved. Using repair constraints the user can restrict considered repairs to those where facts from one relation have been removed only if similar facts have been removed from some other relation. This approach satisfies $\mathcal{P}3$ but not $\mathcal{P}1$. A method of weakening the repair constraints is proposed to get $\mathcal{P}1$, however this comes at the price of losing $\mathcal{P}3$.

In [3], Andritsos et al. extend the framework of consistent query answers with techniques of probabilistic databases. Essentially, only one key dependency per relation is considered and user preference is expressed by assigning a probability value to each of mutually conflicting facts. The probability values must sum to 1 over every clique in the conflict graphs. This framework generalizes the standard framework of consistent query answers: the repairs correspond to possible worlds and have an associated probability. We also note that no repairs are removed from consideration (unless the probability of the world is 0). The query is evaluated over all repairs and the probability assigned to an answer is the sum of probabilities of worlds in which the answer is present. Although the considered databases are repairs, the use of the associated probability values makes it difficult to compare this framework with ours.

In [19], Gatterbauer and Suciu study the problem of conflict resolution in the setting of community databases, where a group of users, each having their own database over the same schema, consolidate their knowledge of facts using mappings. This is essentially a simplified peer-to-peer data exchange setting [18]: the schema consists of one relation with a key dependency and mappings permit to import facts not present in the database of one user from the databases of other users. Facts imported from different users may be conflicting and the authors propose to use a total trust ordering on the mappings to resolve the conflicts. If the mapping network is acyclic, then there exists a unique solution (for every user), and in fact, this solution can be obtained using for instance Algorithm 4 with a specially precomputed, total priority and an instance containing the union of all (accessible) facts. The main challenge addressed by the paper is the setting where the mapping network is cyclic, which may yield several solutions. Furthermore the users are allowed to specify negative facts i.e., facts that are not believed to be true. These two features render the setting incomparable with our approach: cyclic mappings may possibly lead to a cyclic priority relation and conflicts between negative and positive facts cannot be captured with denial constraints.

In [25], Martinez et al. present an interesting framework of *inconsistency management policies* (IMPs) incorporating several complementary approaches to handling inconsistencies in relational databases. An IMP permits to resolve a group of applicable conflicts using general actions: deleting all but one tuple from a *culprit*, a set of tuples forming a connected component in the conflict graph, or fusing their values with an aggregate functions. Additionally, user priorities can be incorporated into IMPs allowing to chose the appropriate actions depending on the source of the tuples, their values etc. IMPs need not, however, specify the resolution of all conflicts and the result of their application is a database that might continue to be inconsistent but on a smaller scale. This framework is much different form the framework of repairs: because the resolution is focused on culprits and not individual conflicts, the produced instances, even if consistent, need not be repairs of the original database instance. On the other hand, the mechanisms allowing to express user priorities are

Fig. 9 Summary of complexity results

	Repair Check	Consistent Answers to	
		$\{\forall,\exists\}$-free queries	conjunctive queries
Rep	PTIME	PTIME	co-NP-complete
$\mathcal{G}$*Rep*	co-NP-complete	Π_2^p-complete	
$\mathcal{P}$*Rep*	LOGSPACE	co-NP-complete	
$\mathcal{C}$*Rep*	PTIME	co-NP-complete	

powerful enough to capture, in specific settings, the Pareto-optimal semantics of priorities.

9 Conclusions and future work

In this paper we have proposed a general framework of preferred repairs and preferred consistent query answers. We have also proposed a set of desirable properties of a family of preferred repairs. We have presented three families of preferred repairs: $\mathcal{P}$*Rep*, $\mathcal{G}$*Rep*, and $\mathcal{C}$*Rep* based on different notions of optimality of compliance with the priority. For every repair family we have presented a sound and complete database repairing algorithm. Figure 9 summarizes the computational complexity results; its first row is taken from [13].

We envision several directions for further work. We plan to investigate other interesting ways of selecting preferred repairs with priorities. Also, extending our approach to cyclic priorities is an intriguing and challenging issue. Including priorities in similar frameworks of preferences [20] leads to losing monotonicity i.e., the property $\mathcal{P}5$ of the resulting family of preferred repairs. A modified, conditional, version of monotonicity may be necessary to capture non-trivial families of repairs.

Along the lines of [5], the computational complexity results could be further studied, by assuming the conformance of functional dependencies with BCNF. Finally, the class of constraints can be extended to universal constraints [28]. This class of constraints allows to express conflicts caused not only by the presence of some facts but also by simultaneous absence of other facts. Conflict hypergraphs can be generalized to extended conflict hypergraphs which include negative facts.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Afrati, F., Kolaitis, P.: Repair checking in inconsistent databases: algorithms and complexity. In: International Conference on Database Theory (ICDT), pp. 31–41. ACM (2009)
3. Andritsos, P., Fuxman, A., Miller, R.J.: Clean answers over dirty databases: a probabilistic approach. In: International Conference on Data Engineering (ICDE), p. 30 (2006)
4. Arenas, M., Bertossi, L., Chomicki, J.: Consistent query answers in inconsistent databases. In: ACM Symposium on Principles of Database Systems (PODS), pp. 68–79 (1999)
5. Arenas, M., Bertossi, L., Chomicki, J., He, X., Raghavan, V., Spinrad, J.: Scalar aggregation in inconsistent databases. Theoret. Comput. Sci. **296**(3), 405–434 (2003)
6. Bertossi, L.: Consistent query answering in databases. SIGMOD Record **35**(2), 68–76 (2006)
7. Bertossi, L., Chomicki, J.: Query answering in inconsistent databases. In: Chomicki, J., van der Meyden, R., Saake, G. (eds.) Logics for Emerging Applications of Databases, pp. 43–83. Springer (2003)
8. Bertossi, L.: Database repairing and consistent query answering. Synthesis Lectures on Data Management **3**(5), 1–121 (2011)

9. Brewka, G.: Preferred subtheories: An extended logical framework for default reasoning. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 1043–1048 (1989)
10. Caroprese, L., Greco, S., Zumpano, E.: Active integrity constraints for database consistency maintenance. IEEE Trans. Knowl. Data Eng. **21**(7), 1042–1058 (2009)
11. Chomicki, J.: Preference formulas in relational queries. ACM T. Database Syst. **28**(4), 427–466 (2003)
12. Chomicki, J.: Consistent query answering: five easy pieces. In: International Conference on Database Theory (ICDT), pp. 1–17 (2007)
13. Chomicki, J., Marcinkowski, J.: Minimal-change integrity maintenance using tuple deletions. Inform. Comput. **197**(1–2), 90–121 (2005)
14. Chomicki, J., Marcinkowski, J.: On the computational complexity of minimal-change integrity maintenance in relational databases. In: Bertossi, L., Hunter, A., Schaub, T., (eds.) Inconsistency Tolerance, pp. 119–150. LNCS 3300. Springer (2005)
15. Chomicki, J., Marcinkowski, J., Staworko, S.: Computing consistent query answers using conflict hypergraphs. In: International Conference on Information and Knowledge Management (CIKM), pp. 417–426. ACM Press (2004)
16. Fagin, R., Ullman, J.D., Vardi, M.Y.: On the semantics of updates in databases. In: ACM Symposium on Principles of Database Systems (PODS), pp. 352–356 (1983)
17. Fan, W.: Dependencies revisited for improving data quality. In: ACM Symposium on Principles of Database Systems (PODS), pp. pages 159–170 (2008)
18. Fuxman, A., Kolaitis, P., Miller, R., Tan, W.-C.: Peer data exchange. ACM T. Database Syst. **31**(4), 1454–1498 (2006)
19. Gatterbauer, W., Suciu, D.: Data conflict resolution using trust mappings. In: ACM SIGMOD International Conference on Management of Data, pp. 219–230 (2010)
20. Greco, G., Lembo, D.: Data integration with preferences among sources. In: International Conference on Conceptual Modeling (ER), pp. 231–244. Springer (2004)
21. Greco, S., Sirangelo, C., Trubitsyna, I., Zumpano, E.: Feasibility conditions and preference criteria in quering and repairing inconsistent databases. In: International Conference on Database and Expert Systems Applications (DEXA), pp. 44–55 (2004)
22. Grosof, B.N.: Prioritized conflict handling for logic programs. In: International Logic Programming Symposium, pp. 197–211 (1997)
23. Halpern, J.Y.: Defining relative likehood in partially-ordered preferential structures. J. Artificial Intelligence Res. **7**, 1–24 (1997)
24. Koltun, V., Papadimitriou, C.: Approximately dominating representatives. Theoret. Comput. Sci. **371**(3), 148–154 (2007)
25. Martinez, M.V., Parisi, F., Pugliese, A., Simari, G.I., Subrahmanian, V.S.: Inconsistency management policies. In: International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 367–377 (2008)
26. Motro, A., Anokhin, P., Acar, A.C.: Utility-based resolution of data inconsistencies. In: International Workshop on Information Quality in Information Systems (IQIS), pp. 35–43. ACM (2004)
27. Sakama, C., Inoue, K.: Prioritized logic programming and its application to commonsense reasoning. Artif. Intell. **123**, 185–222 (2000)
28. Staworko, S., Chomicki, J.: Consistent query answers in the presence of universal constraints. Inform. Syst. **35**(1), 1–22 (2010)
29. Staworko, S., Chomicki, J., Marcinkowski, J.: Preference-driven querying of inconsistent relational databases. In: EDBT Workshops (IIDB), pp. 318–335. Springer (2006)
30. D. Van Nieuwenborgh and D. Vermeir. Preferred answer sets for ordered logic programs. In: European Conference on Logics for Artificial Intelligence (JELIA), pp. 432–443. LNCS 2424. Springer (2002)
31. Vardi, M.Y.: The complexity of relational query languages. In: ACM Symposium on Theory of Computing (STOC), pp. 137–146 (1982)
32. Wijsen, J.: Database repairing using updates. ACM T. Database Syst. **30**(3), 722–768 (2005)
33. Wijsen, J.: On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases. In: ACM Symposium on Principles of Database Systems (PODS), pp. 179–190 (2010)