# A Dichotomy in the Complexity of Deletion Propagation with Functional Dependencies

Benny Kimelfeld

IBM Research–Almaden, San Jose, CA 95120, USA
kimelfeld@us.ibm.com

## ABSTRACT

A classical variant of the view-update problem is deletion propagation, where tuples from the database are deleted in order to realize a desired deletion of a tuple from the view. This operation may cause a (sometimes necessary) side effect—deletion of additional tuples from the view, besides the intentionally deleted one. The goal is to propagate deletion so as to maximize the number of tuples that remain in the view. In this paper, a view is defined by a self-join-free conjunctive query (sjf-CQ) over a schema with functional dependencies. A condition is formulated on the schema and view definition at hand, and the following dichotomy in complexity is established. If the condition is met, then deletion propagation is solvable in polynomial time by an extremely simple algorithm (very similar to the one observed by Buneman et al.). If the condition is violated, then the problem is NP-hard, and it is even hard to realize an approximation ratio that is better than some constant; moreover, deciding whether there is a side-effect-free solution is NP-complete. This result generalizes a recent result by Kimelfeld et al., who ignore functional dependencies. For the class of sjf-CQs, it also generalizes a result by Cong et al., stating that deletion propagation is in polynomial time if keys are preserved by the view.

**Categories and Subject Descriptors:** H.2 [Database Management]: Systems, Database Administration, F.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

**General Terms:** Algorithms, Theory

**Keywords:** Deletion propagation, functional dependencies, complexity dichotomy

## 1. INTRODUCTION

A classical problem in database management is that of *view update* [1, 4–6, 10–12]: translate an update operation on the view to an update of the source database, so that the update on the view is properly realized. A special case of this problem is that of *deletion propagation* in relational databases: given an *undesired* tuple in the view (defined by some monotonic query), delete some tuples from the base relations (where such tuples are referred to as *facts*), so that the undesired tuple disappears from the view. The database resulting from this deletion of tuples is called a *solution*. The *side effect* is the set of deleted view tuples that are different from the undesired one. If *only* the undesired tuple gets to be deleted from the view, then the solution is *side-effect free*. A solution that is side-effect free does not necessarily exist, and therefore, the task is relaxed to that of *minimizing* the side effect [3–5, 14]. That is, the goal is to delete tuples from the base relations so that the undesired tuple disappears from the view, but as many as possible other tuples remain.

Following is a simple example by Cui and Widom [7] (also referenced and used later by Buneman et al. [3] and Kimelfeld et al. [14]). Suppose that we have two relations, GroupUser(group, user) and GroupFile(group, file), representing memberships of users in groups and access permissions of groups to files, respectively. A user $u$ can access the file $f$ if $u$ belongs to a group that can access $f$; that is, there is some $g$, such that GroupUser$(g, u)$ and GroupFile$(g, f)$. Suppose that we want to restrict the access of a specific user to a specific file, by eliminating some user-group or group-file pairings. Furthermore, we would like to do so in a way that a maximum number of user-file access permissions remain. This is exactly the deletion-propagation problem, where the view is defined by the following conjunctive query (CQ):

$$\text{Access}(u, f) :- \text{GroupUser}(g, u), \text{GroupFile}(g, f) \qquad (1)$$

Formally, for a CQ $Q$ (the view definition), the input for the deletion-propagation problem consists of a database instance $I$ and a tuple $\mathbf{a} \in Q(I)$. A *solution* is a subinstance $J$ of $I$ (i.e., $J$ is obtained from $I$ by deleting facts) such that $\mathbf{a} \notin Q(J)$, and an *optimal solution* maximizes the number of tuples in $Q(J)$. The *side effect* is the set $(Q(I) \setminus \{\mathbf{a}\}) \setminus Q(J)$. Buneman et al. [3] identify some classes of CQs (e.g., projection-free CQs) for which a straightforward algorithm produces an optimal solution in polynomial time. But those tractable classes are restricted, as Buneman et al. show that even for a CQ as simple as the above Access$(u, f)$, testing whether there is a solution that is side-effect free is NP-complete. Therefore, finding an *optimal solution* minimizing the side effect is NP-hard for Access$(u, f)$. More recently, Kimelfeld et al. [14] considered views that are defined by a self-join-free CQ (sjf-CQ), and proved a


*PODS'12,* May 21–23, 2012, Scottsdale, Arizona, USA.

dichotomy theorem implying that the views for which the straightforward algorithm is suboptimal are *exactly* those for which deletion propagation is NP-hard. Later, we discuss that dichotomy theorem in more detail.

Still discussing the view Access$(u, f)$ of (1), suppose that now we have the restrictions that each user belongs to at most one group, and each file is associated with at most one group. It is easy to show that then, deletion propagation is in polynomial time, since to delete a tuple (`user`, `file`) from the view, it is enough to consider (and select the best out of) two possible solutions: one obtained by deleting the single fact of the form GroupUser$(g,$ `user`$)$, and the other by deleting the single fact of the form GroupFile$(g,$ `file`$)$. Actually, this example is a special case of the result of Cong et al. [4], stating that when the view (defined by a CQ) preserves keys, deletion propagation is in polynomial time. Now what if each file is associated with one group, but a user may belong to any number of groups (as commonly practiced by operating systems)? It is not difficult to show that, here also, the problem is in polynomial time. Actually, it turns out that *every* nontrivial key constraint suffices for the tractability of deletion propagation in the special case of Access$(u, f)$; moreover, there is one very simple algorithm, which we later refer to as the *unirelation* algorithm, that realizes the polynomial time for each of these key constraints.

The above example illustrates the fact that *functional dependencies* (e.g., key constraints, {conference, year} →chair, etc.), abbreviated *fds*, have an immediate and direct effect on the complexity of deletion propagation. Moreover, this effect is on one direction: if for a certain view definition the problem is in polynomial time when disregarding the fds, it continues to be in polynomial time when fds are taken into account; however, if the problem is hard without the fds, it may no longer be hard when fds are accounted for. Therefore, the class of views that are known to be tractable for deletion propagation would expand as a result of the investigation of the problem in the presence of functional dependencies. This is what we do in this paper.

The formal setting is as follows. The view is defined by an sjf-CQ $Q$ over a schema $\mathbf{S}$ that may have functional dependencies. (Exact definitions of sjf-CQs and schemas are in Section 2.) In the problem MaxDP$\langle \mathbf{S}, Q\rangle$ (where "DP" stands for "Deletion Propagation"), the input consists of a database instance $I$ over $\mathbf{S}$ and a tuple $\mathbf{a} \in Q(I)$. A *solution* is a subinstance $J$ of $I$ (i.e., $J$ is obtained from $I$ by deleting facts) such that $\mathbf{a} \notin Q(J)$. The goal is to find an *optimal solution*, which is a solution that maximizes the number of tuples in $Q(J)$. The decision problem FreeDP$\langle \mathbf{S}, Q\rangle$ has the same input (i.e., $I$ and $\mathbf{a}$), and the goal is to decide whether there is a solution that is side-effect free, that is, a subinstance $J$ of $I$ with $Q(J) = Q(I) \setminus \{\mathbf{a}\}$.

The main result of this paper is a generalization of the dichotomy theorem of Kimelfeld et al. [14] to accommodate functional dependencies. More precisely, in Section 3 we phrase a syntactic condition on (the fds of) the schema $\mathbf{S}$ and the sjf-CQ $Q$ at hand. We call this condition the *tractability condition*. Then, the following is proved (Theorem 6.1).

- If the tractability condition is met, then MaxDP$\langle \mathbf{S}, Q\rangle$ (as well as FreeDP$\langle \mathbf{S}, Q\rangle$) is solvable in polynomial time by a straightforward algorithm. (Section 3.)
- Otherwise, MaxDP$\langle \mathbf{S}, Q\rangle$ is NP-hard, even to approximate better than some constant smaller than one; moreover, FreeDP$\langle \mathbf{S}, Q\rangle$ is NP-complete. (Section 5.)

The straightforward algorithm in the positive side of the dichotomy is called the *unirelation* algorithm, and is described in Section 3. This algorithm is very similar to the "trivial" algorithm used by Kimelfeld et al. [14] and originally observed by Buneman et al. [3]. The notion of approximation used in the negative side of the dichotomy is that of Kimelfeld et al. [14], which differs from that of Buneman et al. [3]. In particular, here the goal is to *maximize* the number of tuples that remain in the view, rather than to *minimize* the side effect [3]. (See [14] for a discussion on the difference between the two notions of approximation.)

To give the flavor of our tractability condition, we illustrate it on the CQ Access$(u, f)$ defined in (1). (More involved examples are in the body of the paper.) In this CQ, $u$ and $f$ are *head variables* while $g$ is an *existential variable*. Suppose first that the underlying schema $\mathbf{S}$ has no fds. In the *existential-connectivity graph* of $Q$, denoted $\mathcal{G}_{\exists}(Q)$, each atom is a node, and two atoms are connected by an edge whenever they share at least one existential variable. The tractability condition in the dichotomy of Kimelfeld et al. [14] says that for each connected component $P$ of $\mathcal{G}_{\exists}(Q)$ there is an atom that contains all head variables of $P$. For $Q =$ Access$(u, f)$, the graph $\mathcal{G}_{\exists}(Q)$ has only one connected component (containing the two atoms). Since no atom contains both $u$ and $f$, the tractability condition of Kimelfeld et al. [14] is violated, thus MaxDP$\langle \mathbf{S}, Q\rangle$ is NP-hard. Now suppose that $\mathbf{S}$ has the fd file → group (i.e., every file belongs to one group). Then the existential variable $g$ is *functionally determined* by the head variables; that is, in a mapping from $Q$ to the database, if we know what the head variables are mapped to, then we also know what $g$ is mapped to. So, we treat $g$ as if it were a head variable. But now, the edge of $\mathcal{G}_{\exists}(Q)$ disappears (as the two atoms no longer share an existential variable), and we get two connected components, each with one node. Then, the tractability condition of Kimelfeld et al. [14] trivially holds, and so, MaxDP$\langle \mathbf{S}, Q\rangle$ is now in polynomial time. Lastly, suppose that instead of file → group, the schema $\mathbf{S}$ contains the fd group → user (i.e., every group has at most one user). Although the atom GroupFile$(g, f)$ does not contain both head variables ($u$ and $f$), it functionally determines both of them. This implies that the tractability condition of this paper is satisfied, and hence, MaxDP$\langle \mathbf{S}, Q\rangle$ is again in polynomial time.

Proving the tractable part of the dichotomy is fairly simple. However, the proof of the hardness part is nontrivial, and entails a fairly intricate construction and case analysis. A central concept in the proof is that of *graph separation*. In Section 5, we review the proof, which is restricted to the NP-completeness of FreeDP$\langle \mathbf{S}, Q\rangle$. In Section 6 we discuss the adaptation of the proof to hardness of approximation for MaxDP$\langle \mathbf{S}, Q\rangle$; the discussion is brief, as this adaptation uses ideas similar to the ones used by Kimelfeld et al. [14].

The work reported here is restricted to a special setting within view update: deletion propagation for sjf-CQs in the presence of fds, with the goal of preserving as many tuples of the view as possible. Other aspects of deletion propagation, and view update in general, have been studied in the literature. For example, deletion propagation was studied with the goal of minimizing the *source side effect* [3–5], namely, finding a solution with a minimal number of missing facts. A significant research effort has been invested on defining the semantics of (and notions of correctness for) general view-update operations (e.g., deletion and insertion

of tuples) [1, 6, 10, 12]. For example, Cosmadakis and Papadimitriou [6] introduced a requirement for a view to keep intact a *complement view*.

Several dichotomies in the complexity of operations involving CQs (and sjf-CQs) were proved in the literature. For example, Dalvi and Suciu [9] considered query evaluation over probabilistic databases with independent tuples, and classified the sjf-CQs into those that can be evaluated in polynomial time and those that are #P-hard. Later, Dalvi et al. [8] extended this result to the class of unions of conjunctive queries (allowing self joins). More recently, Meliou et al. [18] showed a dichotomy (polynomial time vs. NP-completeness) in the complexity of computing the *degree of responsibility* of a database tuple for an answer in the result of an sjf-CQ. Examples of dichotomy theorems that involve CQs and key constraints are within the topic of *consistent query answering*. Kolaitis and Pema [15] proved a dichotomy (polynomial time vs. coNP-hardness) in the complexity of computing the consistent answers of a Boolean sjf-CQ with exactly two atoms. Finally, Maslowski and Wijsen [16] showed a dichotomy (polynomial vs. #P-hardness) in the complexity of counting the database repairs that satisfy a Boolean sjf-CQ. We are not aware of any dichotomy theorem that applies to sjf-CQs and general functional dependencies like the main theorem in this paper (Theorem 6.1).

The classical use case for deletion propagation is database administration: actual translation of a deletion in the view to deletions in the base relations. Kimelfeld et al. [14] discuss further motivation (e.g., information extraction) in the context of database debugging: the undesired answer is a detected error (false positive), and the facts deleted in an (approximately) optimal solution serve as a possible *cause* of the error. That cause is meaningful in the sense that it is focused on the erroneous answer and has a small (if any) effect on the other results. This notion of causality is different from that of Meliou et al. [17], which actually corresponds to deletion propagation with a minimal source side effect.

Due to a lack of space, the proofs are omitted, and presented in the extended version of this paper [13].

# 2. SETTING AND BACKGROUND

This section describes the formal setting for this paper, and provides some needed background.

## 2.1 Schemas and Functional Dependencies

In this paper, a *schema* $\mathbf{S}$ is a pair $(\mathcal{R}, \Delta)$, where $\mathcal{R} = \{R_1, \ldots, R_n\}$ is a finite set of *relation symbols* and $\Delta$ is a set of *functional dependencies*. We abbreviate *functional dependency* as *fd*. Each relation symbol $R_i \in \mathcal{R}$ has an *arity*, denoted $\mathrm{arity}(R_i)$, which is a natural (nonzero) number. An fd $\delta \in \Delta$ has the form $R_i : A \to B$, where $R_i \in \mathcal{R}$ and $A$ and $B$ are subsets of $\{1, \ldots, \mathrm{arity}(R_i)\}$. We use $R_i : A \to j$ and $R_i : j' \to j$ as shorthand notations of $R_i : A \to \{j\}$ and $R_i : \{j'\} \to \{j\}$, respectively.

We assume an infinite set $\mathsf{Const}$ of *constants*. A database over a schema $\mathbf{S}$ is called an *instance* (*over* $\mathbf{S}$), and its values are taken from $\mathsf{Const}$. Formally, an instance $I$ over a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ consists of a finite relation $R_i^I \subseteq \mathsf{Const}^{\mathrm{arity}(R_i)}$ for each relation symbol $R_i \in \mathcal{R}$, such that all the fds of $\mathbf{S}$ are satisfied; that is, for each fd $R_i : A \to B$ in $\Delta$ and tuples $\mathbf{t}$ and $\mathbf{u}$ in $R_i^I$, if $\mathbf{t}$ and $\mathbf{u}$ agree on (i.e., have the same values for) the indices of $A$, then they also agree on the indices of $B$. If $I$ is an instance over $\mathbf{S}$ and $\mathbf{t}$ is a tuple in $R_i^I$, then we say that $R_i(\mathbf{t})$ is a *fact of* $I$. Notationally, we view an instance $I$ as the set of its facts. As an example, $J \subseteq I$ means that $R_i^J \subseteq R_i^I$ for all $R_i \in \mathcal{R}$, in which case we say that $J$ is *subinstance* of $I$. (Note that if $I$ satisfies $\Delta$, then every subinstance of $I$ satisfies $\Delta$ as well.)

## 2.2 Conjunctive Queries

We fix an infinite set $\mathsf{Var}$ of *variables*, disjoint from $\mathsf{Const}$. We usually denote variables by lowercase letters from the end of the Latin alphabet (e.g., $x$, $y$ and $z$). We use the Datalog style for denoting a conjunctive query (abbrev. CQ); that is, a CQ over a schema $\mathbf{S}$ is an expression of the form $Q(\mathbf{y}) :- \varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$, where $\mathbf{x}$ and $\mathbf{y}$ are disjoint tuples of variables (from $\mathsf{Var}$), $\mathbf{c}$ is a tuple of constants (from $\mathsf{Const}$), and $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is a conjunction of atomic formulas $\phi_i(\mathbf{x}, \mathbf{y}, \mathbf{c})$ over $\mathbf{S}$; an atomic formula is also called an *atom*. We may write just $Q(\mathbf{y})$, or even just $Q$, if $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is irrelevant. We denote by $\mathrm{atoms}(Q)$ the set of atoms of $Q$. We usually write $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ by simply listing the atoms of $Q$. We require every variable of $\mathbf{y}$ to occur (at least once) in $\varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$. The *arity of* $Q$, denoted $\mathrm{arity}(Q)$, is the length of the tuple $\mathbf{y}$.

Let $Q(\mathbf{y}) :- \varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ be a CQ. A variable in $\mathbf{x}$ is called an *existential* variable, and a variable in $\mathbf{y}$ is called a *head* variable. We use $\mathsf{Var}_\exists(Q)$ and $\mathsf{Var}_\mathsf{h}(Q)$ to denote the sets of existential variables and head variables of $Q$, respectively. Similarly, if $\phi$ is an atom of $Q$, then $\mathsf{Var}_\exists(\phi)$ and $\mathsf{Var}_\mathsf{h}(\phi)$ denote the sets of existential and head variables, respectively, that occur in $\phi$. We denote by $\mathsf{Var}(Q)$ and $\mathsf{Var}(\phi)$ the unions $\mathsf{Var}_\exists(Q) \cup \mathsf{Var}_\mathsf{h}(Q)$ and $\mathsf{Var}_\exists(\phi) \cup \mathsf{Var}_\mathsf{h}(\phi)$, respectively.

This paper mostly considers CQs that are *self-join free*, which means that each relation symbol occurs at most once in the CQ. We use *sjf-CQ* as an abbreviation of *self-join-free CQ*. If $R$ is a relation symbol that occurs in the sjf-CQ $Q(\mathbf{y}) :- \varphi(\mathbf{x}, \mathbf{y}, \mathbf{c})$, then the unique atom over $R$ is denoted by $\phi_R(\mathbf{x}, \mathbf{y}, \mathbf{c})$ or just $\phi_R$. Similarly, $R_\phi$ denotes the relation symbol of the atom $\phi$.

Example 2.1. An important CQ in this work is the sjf-CQ $Q^\star$ which is the same as the CQ $\mathrm{Access}(u, f)$ defined in (1), up to renaming of relation symbols and variables. The schema is $\mathbf{S}^\star$, and it has two binary relation symbols $R_1$ and $R_2$ and no functional dependencies. The sjf-CQ $Q^\star$ over $\mathbf{S}^\star$ is the following.

$$Q^\star(y_1, y_2) :- R_1(x, y_1), R_2(x, y_2) \tag{2}$$

The atoms of $Q^\star$ are $\phi_{R_1} = R_1(x, y_1)$ and $\phi_{R_2} = R_2(x, y_2)$. In the examples of this paper, $R_i$ and $R_j$ are assumed to be different symbols whenever $i \neq j$. In particular, $R_1 \neq R_2$, so $Q^\star$ is indeed an sjf-CQ. There is only one existential variable in $Q^\star$, namely $x$, and the two head variables are $y_1$ and $y_2$. Hence $\mathsf{Var}_\exists(Q) = \{x\}$ and $\mathsf{Var}_\mathsf{h}(Q) = \{y_1, y_2\}$. Furthermore, $\mathsf{Var}_\exists(\phi_{R_1}) = \{x\}$ and $\mathsf{Var}_\mathsf{h}(\phi_{R_1}) = \{y_1\}$. □

Let $\mathbf{S}$ be a schema and let $Q(\mathbf{y})$ be a CQ over $\mathbf{S}$. An *assignment* for $Q$ is a mapping $\mu : \mathsf{Var}(Q) \to \mathsf{Const}$. For an assignment $\mu$ for $Q$, the tuple $\mu(\mathbf{y})$ is the one obtained from $\mathbf{y}$ by replacing every head variable $y$ with the constant $\mu(y)$. Similarly, for an atom $\phi$ of $Q$, the fact $\mu(\phi)$ is the one obtained from $\phi$ by replacing every variable $z$ with the constant $\mu(z)$. Let $I$ be an instance over $\mathbf{S}$. A *match for* $Q$ *in* $I$ is an assignment $\mu$ for $Q$, such that $\mu(\phi)$ is a fact of $I$ for all $\phi \in \mathrm{atoms}(Q)$. We denote by $\mathcal{M}(Q, I)$ the set of all matches for $Q$ in $I$. If $\mu \in \mathcal{M}(Q, I)$, then $\mu(\mathbf{y})$ is called

an *answer* (*for $Q$ in $I$*). The *result* of evaluating $Q$ over $I$, denoted $Q(I)$, is the set of all the answers for $Q$ in $I$; that is, $Q(I)$ is the set $\{\mu(\mathbf{y}) \mid \mu \in \mathcal{M}(Q, I)\}$.

Over a given schema, let $Q(\mathbf{y})$ be a CQ, let $I$ be an instance, and let $\mathbf{b}$ be an answer. A fact $f \in I$ is $(Q, \mathbf{b})$-*useful* (*in $I$*) if for some $\mu \in \mathcal{M}(Q, I)$ and $\phi \in \text{atoms}(Q)$ we have $\mu(\mathbf{y}) = \mathbf{b}$ and $\mu(\phi) = f$. A fact $f \in I$ is $Q$-*useful* (*in $I$*) if $f$ is $(Q, \mathbf{b})$-useful for some answer $\mathbf{b}$.

EXAMPLE 2.2. Take the schema $\mathbf{S}^\star$ and the CQ $Q^\star$ of Example 2.1. Let $I^\star$ be the instance with the following facts:

$$R_1(\mathtt{0}, \triangleleft) \quad R_1(\mathtt{0}, \mathtt{a}) \quad R_1(\mathtt{1}, \triangleleft) \quad R_1(\mathtt{1}, \mathtt{c})$$
$$R_2(\mathtt{0}, \triangleright) \quad R_2(\mathtt{0}, \mathtt{b}) \quad R_2(\mathtt{1}, \triangleright)$$

The top six cells in the leftmost column of the table in Figure 4 show the matches in $\mathcal{M}(Q^\star, I^\star)$ (the rest of the table will be discussed in Section 5). As a shorthand notation, we use the triple $c, b_1, b_2$ (e.g., $\mathtt{0}, \triangleleft, \mathtt{b}$) to specify the match that maps $x^\star$, $y_1^\star$ and $y_2^\star$ to $c$, $b_1$ and $b_2$, respectively. Note that $Q^\star(I^\star) = \{(\triangleleft, \triangleright), (\triangleleft, \mathtt{b}), (\mathtt{a}, \triangleright), (\mathtt{a}, \mathtt{b}), (\mathtt{c}, \triangleright)\}$. For the answer $\mathbf{a}^\star = (\triangleleft, \triangleright)$, the $(Q^\star, \mathbf{a}^\star)$-useful facts are $R_1(\mathtt{0}, \triangleleft)$, $R_1(\mathtt{1}, \triangleleft)$, $R_2(\mathtt{0}, \triangleright)$ and $R_2(\mathtt{1}, \triangleright)$. Finally, note that every fact of $I^\star$ is $Q^\star$-useful; but if we added the fact $R_1(\mathtt{3}, \mathtt{a})$, it would not be $Q^\star$-useful (since it would have no fact of $R_2$ to join with). □

### 2.2.1 CQs and Functional Dependencies

Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema, and let $Q$ be a CQ over $\mathbf{S}$. If $\mu$ is an assignment for $Q$ and $Z$ is a subset of $\mathsf{Var}(Q)$, then $\mu[Z]$ denotes the restriction of $\mu$ to the variables in $Z$ (i.e., $Z$ is the domain of $\mu[Z]$, and $\mu[Z](z) = \mu(z)$ for all $z \in Z$). Let $Z_1$ and $Z_2$ be subsets of $\mathsf{Var}(Q)$. We denote by $Q : Z_1 \to Z_2$ the functional dependency stating that for every instance $I$ over $\mathbf{S}$ and matches $\mu, \mu' \in \mathcal{M}(Q, I)$ we have:

$$\mu[Z_1] = \mu'[Z_1] \Rightarrow \mu[Z_2] = \mu'[Z_2]$$

Note that when $\Delta$ is empty (e.g., as in the setting of [3,14]), $Q : Z_1 \to Z_2$ is equivalent to $Z_1 \supseteq Z_2$. We may write $Q : Z \to z$ instead of $Q : Z \to \{z\}$. The *image* of $Z$, denoted $img(Z)$, is the set of all the variables $z \in \mathsf{Var}(Q)$ with $Q : Z \to z$. Note that $Z \subseteq img(Z)$. For $\phi \in \text{atoms}(Q)$, we use $img(\phi)$ as a shorthand notation of $img(\mathsf{Var}(\phi))$.

EXAMPLE 2.3. Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be such that $\mathcal{R}$ contains two binary relation symbols $R_1$, $R_2$ and two ternary relation symbols $S$ and $T$. Suppose that $\Delta$ has the following fds:

$$R_1 : 1 \to 2 \quad R_2 : 1 \to 2 \quad S : \{1, 2\} \to 3$$

Let $Q$ be the following CQ.

$$Q(y_1, y_2, y_3) :-$$
$$R_1(x_1, y_1), R_2(x_2, y_2), S(y_1, y_3, x), T(x, x_1, x_2)$$

We have $img(\{x_1\}) = \{x_1, y_1\}$ due to the fd $R_1 : 1 \to 2$, $img(\{x_2\}) = \{x_2, y_2\}$ due to $R_2 : 1 \to 2$, and $img(\{y_1, y_3\}) = \{y_1, y_3, x\}$ due to $S : \{1, 2\} \to 3$. Finally, we have $img(\phi_T) = \{x, x_1, x_2, y_1, y_2\}$ due to the fds $R_1 : 1 \to 2$ and $R_2 : 1 \to 2$. (Recall that $\phi_T$ is the atom $T(x, x_1, x_2)$.) □

## 2.3 Deletion Propagation

Let $\mathbf{S}$ be a schema, and let $Q$ be a CQ. The problem of maximizing the view in deletion propagation, with $Q$ as the view definition, is denoted by $\text{MAXDP}\langle \mathbf{S}, Q \rangle$ and is defined as follows. The input consists of an instance $I$ over $\mathbf{S}$, and a tuple $\mathbf{a} \in Q(I)$. A *solution* (*for $I$ and $\mathbf{a}$*) is a subinstance $J \subseteq I$, such that $\mathbf{a} \notin Q(J)$. The goal is to find an *optimal* solution, which is a solution $J$ that maximizes $|Q(J)|$; that is, $J$ is such that $|Q(J)| \geq |Q(J')|$ for all solutions $J'$. The problem of determining whether there is a *side-effect-free* solution, denoted $\text{FREEDP}\langle \mathbf{S}, Q \rangle$, is that of deciding whether there is a solution in which we lose exactly $\mathbf{a}$; that is, given the input $I$ and $\mathbf{a}$, decide whether there is a subinstance $J \subseteq I$ with $Q(J) = Q(I) \setminus \{\mathbf{a}\}$.

For example, take the schema $\mathbf{S}^\star$ and the CQ $Q^\star$ of Example 2.1. The input for $\text{MAXDP}\langle \mathbf{S}^\star, Q^\star \rangle$, as well as for $\text{FREEDP}\langle \mathbf{S}^\star, Q^\star \rangle$, consists of an instance $I^\star$ over $\mathbf{S}^\star$ and an answer $\mathbf{a}^\star = (a_1, a_2)$. To obtain a solution, we need to delete from $I^\star$ facts, so that for all constants $c$ where both $R_1(c, a_1)$ and $R_2(c, a_2)$ are in $I^\star$, at least one of the two is deleted. In $\text{MAXDP}\langle \mathbf{S}^\star, Q^\star \rangle$, the goal is to have as many as possible pairs remaining in $Q^\star(I^\star)$ after the deletion. In $\text{FREEDP}\langle \mathbf{S}^\star, Q^\star \rangle$, the goal is to determine whether there is a solution that preserves every original answer except for $\mathbf{a}^\star$.

As follows from the presentation, we analyze the complexity of $\text{MAXDP}\langle \mathbf{S}, Q \rangle$ and $\text{FREEDP}\langle \mathbf{S}, Q \rangle$ under the yardstick of *data complexity*: the schema $\mathbf{S}$ and the CQ $Q$ (that parameterize the problems) are fixed, and the input consists of the instance $I$ and the answer $\mathbf{a}$. Buneman et al. [3] proved the following.

THEOREM 2.4. [3] $\text{FREEDP}\langle \mathbf{S}^\star, Q^\star \rangle$ *is NP-complete (and hence,* $\text{MAXDP}\langle \mathbf{S}^\star, Q^\star \rangle$ *is NP-hard).*

Kimelfeld et al. [14] generalized Theorem 2.4 to a complexity dichotomy over the class of all sjf-CQs $Q$; we discuss that in the next section.

Another problem that falls in our setting is that of deletion propagation for *key-preserving views* [4], which is described as follows. Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema. For a relation symbol $R \in \mathcal{R}$, a *primary key* for $R$ is a set $K$ of indices in $\{1, \ldots, \text{arity}(R)\}$, such that $\Delta$ contains the fd $K \to \{1, \ldots, \text{arity}(R)\}$. We say that a CQ $Q$ over $\mathbf{S}$ is *key preserving* if for every $\phi \in \text{atoms}(Q)$ there is a key $K_\phi$ for $R_\phi$, such that $\phi$ has no existential variables in the positions of $K_\phi$. Cong et al. [4] proved the following.

THEOREM 2.5. [4] *Let $\mathbf{S}$ be a schema and let $Q$ be a CQ over $\mathbf{S}$, such that $Q$ is key preserving. Then* $\text{MAXDP}\langle \mathbf{S}, Q \rangle$ *(as well as* $\text{FREEDP}\langle \mathbf{S}, Q \rangle$*) is solvable in polynomial time.*

In Section 3, we will show how Theorem 2.5 is a special case of the main result of this paper in the case where $Q$ is self-join free.

## 2.4 Head Domination and Dichotomy

Let $\mathbf{S}$ be a schema, and let $Q$ be a CQ over a $\mathbf{S}$. The *existential-connectivity graph* of $Q$, denoted $\mathcal{G}_\exists(Q)$, is the undirected graph that has atoms$(Q)$ as the set of nodes, and that has an edge $\{\phi_1, \phi_2\}$ whenever $\phi_1$ and $\phi_2$ have at least one existential variable in common (in notation, $\mathsf{Var}_\exists(\phi_1) \cap \mathsf{Var}_\exists(\phi_2) \neq \emptyset$). Let $P$ be a subset of atoms$(Q)$. We denote by $\mathsf{Var}(P)$ the set of all the variables that occur in $P$, by $\mathsf{Var}_\mathsf{h}(P)$ the set $\mathsf{Var}(P) \cap \mathsf{Var}_\mathsf{h}(Q)$, and by $\mathsf{Var}_\exists(P)$ the set $\mathsf{Var}(P) \cap \mathsf{Var}_\exists(Q)$.

EXAMPLE 2.6. The left side of Figure 1 shows the graph $\mathcal{G}_\exists(Q)$ for the CQ $Q$ of Example 2.3. Observe that there is no

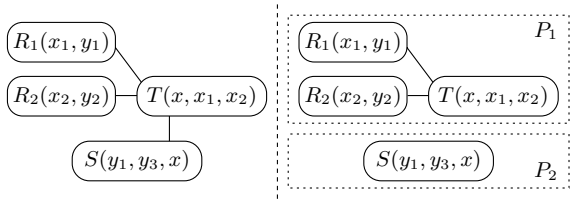


**Figure 1: The graphs $\mathcal{G}_\exists(Q)$ (left) and $\mathcal{G}_\exists(Q^+)$ (right) for the CQ $Q$ of Example 2.3**

edge between $R_1(x_1, y_1)$ and $S(y_1, y_3, x)$ since the two atoms do not share an existential variable (they share $y_1$, which is a head variable). The graph $\mathcal{G}_\exists(Q)$ is connected, and its single connected component $P$ satisfies $\mathsf{Var_h}(P) = \{y_1, y_2, y_3\}$ and $\mathsf{Var}_\exists(P) = \{x, x_1, x_2\}$. □

Kimelfeld et al. [14] defined the *head-domination* property of a CQ.

DEFINITION 2.7. (**Head Domination [14]**) A CQ $Q$ over a schema $\mathbf{S}$ has the *head-domination* property if for every connected component $P$ of $\mathcal{G}_\exists(Q)$ there is an atom $\phi$ of $Q$ such that $\mathsf{Var_h}(P) \subseteq \mathsf{Var}(\phi)$. □

Note that in the definition, the atom $\phi$ satisfying $\mathsf{Var_h}(P) \subseteq \mathsf{Var}(\phi)$ is not necessarily in $P$.

EXAMPLE 2.8. A CQ that has head domination is the following:

$$Q'(y_1, y_2, y_3) :- R_1(x, y_1), R_2(x, y_2), S(y_1, y_2), T(y_2, y_3)$$

Note that $\mathcal{G}_\exists(Q')$ has three connected components: $P_1 = \{\phi_{R_1}, \phi_{R_2}\}$, $P_2 = \{\phi_S\}$, $P_3 = \{\phi_T\}$. Since $\mathsf{Var_h}(P_1) \subseteq \mathsf{Var}(\phi_S)$, $\mathsf{Var_h}(P_2) \subseteq \mathsf{Var}(\phi_S)$ and $\mathsf{Var_h}(P_3) \subseteq \mathsf{Var}(\phi_T)$, we get that, indeed, $Q'$ has head domination.

As a negative example, consider again the CQ $Q$ of Example 2.3. As shown in Example 2.6 (through Figure 1), $\mathcal{G}_\exists(Q)$ has only one connected component. Since no atom contains all the head variables of $Q$, we get that $Q$ does not have head domination. □

Kimelfeld et al. [14] proved the following theorem, which states a dichotomy in the complexities of $\text{MAXDP}\langle\mathbf{S}, Q\rangle$ and $\text{FREEDP}\langle\mathbf{S}, Q\rangle$ in the absence of fds.

THEOREM 2.9. [14] *Let $\mathbf{S} = (\mathcal{R}, \Delta)$ be a schema and let $Q$ be an sjf-CQ over $\mathbf{S}$.*

1. *If $Q$ has head domination, then $\text{MAXDP}\langle\mathbf{S}, Q\rangle$ can be solved in polynomial time.*
2. *If $\Delta = \emptyset$ and $Q$ does not have head domination, then $\text{FREEDP}\langle\mathbf{S}, Q\rangle$ is NP-complete.*

In this paper, we extend the complexity dichotomy of Theorem 2.9 to account for fds (Theorem 6.1). Specifically, in the next section we define a *tractability condition*, which is a syntactic condition on the schema $\mathbf{S}$ and the CQ $Q$ at hand, such that fds are taken into account to weaken the property of head domination. We will show that the tractability condition gives a dichotomy similar to that of Theorem 2.9, that is, its satisfaction implies tractability of $\text{MAXDP}\langle\mathbf{S}, Q\rangle$ and its violation implies NP-completeness of $\text{FREEDP}\langle\mathbf{S}, Q\rangle$. (Of course, the tractability condition is equivalent to head domination in the absence of fds.) As we will show, this extension of Theorem 2.9 requires arguments, concepts and constructions that are significantly different from those used for proving Theorem 2.9.

Actually, the result of Kimelfeld et al. [14] is stronger than Theorem 2.9. In the tractable part (item 1), the problem $\text{MAXDP}\langle\mathbf{S}, Q\rangle$ is solved by an extremely simple algorithm that has been observed already by Buneman et al. [3]. We will show that the tractable side in the current paper is solved by essentially the same algorithm, with a slight modification. In the intractable part of Theorem 2.9 (item 2), Kimelfeld et al. further show some hardness of approximation for $\text{MAXDP}\langle\mathbf{S}, Q\rangle$. To simplify the presentation, we ignore approximation through most of the paper, except for Section 6; there, we will show that a similar hardness of approximation is in the intractable side of this paper's dichotomy as well.

# 3. TRACTABILITY

In this section, we define and discuss the tractability condition on a schema $\mathbf{S}$ and an sjf-CQ $Q$ over $\mathbf{S}$. Also in this section we will state that the condition is indeed sufficient for $\text{MAXDP}\langle\mathbf{S}, Q\rangle$ to be solvable in polynomial time. In Section 5, we will state that the condition is also necessary for this tractability (under standard complexity assumptions). We begin by describing the *unirelation algorithm*, which is a very simple algorithm that produces a (not-necessarily optimal) solution given the input $I$ and $\mathbf{a}$ for $\text{MAXDP}\langle\mathbf{S}, Q\rangle$.

## 3.1 The Unirelation Algorithm

Let $\mathbf{S}$ be a schema, and let $Q$ be a CQ over $\mathbf{S}$. Buneman et al. [3] observed the following simple algorithm (also called *the trivial algorithm* [14]) for producing a solution, given the input $I$ and $\mathbf{a}$ for $\text{MAXDP}\langle\mathbf{S}, Q\rangle$. For each $\phi \in \text{atoms}(Q)$, construct the candidate solution $J_\phi$ by deleting from $I$ every fact that can be obtained from $\phi$ by an assignment that agrees with $\mathbf{a}$ on the head variables; then, select the $J_\phi$ with the maximal $|Q(J_\phi)|$. The *unirelation algorithm*, denoted $\mathsf{UniRel}\langle\mathbf{S}, Q\rangle$ and depicted in Figure 2, is essentially the same algorithm, except that we delete only facts that are $(Q, \mathbf{a})$-useful (that is, the facts we delete are always of the form $\mu(\phi)$ where $\mu \in \mathcal{M}(Q, I)$ is such that $\mu(\mathbf{y}) = \mathbf{a}$). The following straightforward proposition states that the unirelation algorithm is correct and terminates in polynomial time.

PROPOSITION 3.1. $\mathsf{UniRel}\langle\mathbf{S}, Q\rangle(I, \mathbf{a})$ *returns a solution in polynomial time.*

It follows immediately from the results of Kimelfeld et al. [14] that if $Q$ is an sjf-CQ with head domination, then $\mathsf{UniRel}\langle\mathbf{S}, Q\rangle$ produces an optimal solution. It also follows immediately that in the absence of fds, head domination is a necessary condition for the optimality of $\mathsf{UniRel}\langle\mathbf{S}, Q\rangle$; but this is no longer true when fds are present, as the following example shows.

EXAMPLE 3.2. Consider again the CQ $Q^\star$ over $\mathbf{S}^\star$ (Example 2.1), but now let us use the schema $\mathbf{S}^\star_0$, which is the same as $\mathbf{S}^\star$ except that it has the fd $R_1 : 1 \to 2$. We claim that, given input $I^\star$ and $\mathbf{a}^\star = (a_1, a_2)$, the execution of $\mathsf{UniRel}\langle\mathbf{S}^\star_0, Q^\star\rangle(I^\star, \mathbf{a}^\star)$ returns an optimal solution. As proof, we will show that for $\phi = R_2(x, y_2)$ the solution $J_\phi$, constructed by the algorithm, is optimal. Let $J_{\text{opt}}$ be

**Algorithm** $\mathsf{UniRel}\langle \mathbf{S}, Q\rangle(I, \mathbf{a})$

1: $J \leftarrow \emptyset$
2: **for all** $\phi \in \text{atoms}(Q)$ **do**
3: $\quad J_\phi \leftarrow I$
4: $\quad$ Remove from $J_\phi$ every fact over $R_\phi$ that is $(Q, \mathbf{a})$-useful in $I$
5: $\quad$ **if** $|Q(J_\phi)| > |Q(J)|$ **then**
6: $\quad\quad J \leftarrow J_\phi$
7: **return** $J$

**Figure 2: The unirelation algorithm**

an optimal solution. It is enough to show that every fact $R_2(c, a_2)$ that is $(Q^\star, \mathbf{a}^\star)$-useful in $I^\star$ is not $Q$-useful in $J_{\text{opt}}$ (hence, $Q(J_{\text{opt}}) \subseteq Q(J_\phi)$). Indeed, if $J_{\text{opt}}$ contains both $R_1(c, b_1)$ and $R_2(c, a_2)$, then the fd implies that $b_1 = a_1$, since $I^\star$ contains also $R_1(c, a_1)$ (or otherwise $R_2(c, a_2)$ is not $(Q^\star, \mathbf{a}^\star)$-useful in $I^\star$). This means that $Q(J_{\text{opt}})$ contains $\mathbf{a}^\star$, which contradicts the fact that $J_{\text{opt}}$ is a solution. □

## 3.2 Functional Head Domination

The idea that is used in Example 3.2 to show the optimality of $\mathsf{UniRel}\langle \mathbf{S}_0^\star, Q^\star\rangle$ is the following. The head variables of the single connected component of $\mathcal{G}_\exists(Q^\star)$ are $y_1$ and $y_2$. The atom $\phi_{R_2}$ does not contain both $y_1$ and $y_2$, but $img(\phi_{R_2})$ does. So, intuitively, we can treat $\phi_{R_2}$ *as if* it contains both $y_1$ and $y_2$. This idea is formalized next.

Definition 3.3. (**Functional Head Domination**) A CQ $Q$ over a schema $\mathbf{S}$ has the *functional head-domination* property if for every connected component $P$ of $\mathcal{G}_\exists(Q)$ there is an atom $\phi$ of $Q$ such that $\mathsf{Var_h}(P) \subseteq img(\phi)$. □

For example, the CQ $Q^\star$ over $\mathbf{S}_0^\star$ in Example 3.2 has functional head domination, since $img(\phi_{R_2})$ contains both $y_1$ and $y_2$ (as previously explained). Another example follows.

Example 3.4. Consider the schema $\mathbf{S}$ and CQ $Q$ of Example 2.3. Recall that the left side of Figure 1 shows the graph $\mathcal{G}_\exists(Q)$. Since $\mathcal{G}_\exists(Q)$ is connected and no atom $\phi$ of $Q$ is such that $img(\phi)$ contains all the head variables, $Q$ does not have functional head domination.

Now, let $Q'$ be obtained from $Q$ by adding $x$ to the head (so the head variables of $Q'$ are $x$ and the $y_i$). The right side of Figure 1 shows $\mathcal{G}_\exists(Q')$. For the connected component $P_1$ we have $\mathsf{Var_h}(P_1) = \{x, y_1, y_2\} \subseteq img(\phi_T)$, and $\mathsf{Var_h}(P_2) = \{x, y_1, y_3\} \subseteq img(\phi_S)$. Thus, $Q'$ has functional head domination. □

The following theorem generalizes the idea illustrated in Example 3.2: in the absence of self joins, functional head domination is sufficient for $\text{MaxDP}\langle \mathbf{S}, Q\rangle$ to be solvable in polynomial time. This theorem follows from Theorems 3.10 and 3.11 given later in this section.

Theorem 3.5. *Let* $\mathbf{S}$ *be a schema, and let* $Q$ *be an sjf-CQ over* $\mathbf{S}$. *If* $Q$ *has functional head domination, then* $\mathsf{UniRel}\langle \mathbf{S}, Q\rangle$ *optimally solves* $\text{MaxDP}\langle \mathbf{S}, Q\rangle$.

Nevertheless, functional head domination is still not a necessary condition for the optimality of the unirelation algorithm, as the following example shows.

Example 3.6. Consider again the CQ $Q^\star$ over $\mathbf{S}^\star$ (Example 2.1), but now let us use the schema $\mathbf{S}_1^\star$, which is the same as $\mathbf{S}^\star$ except that it has the fd $R_1 : 2 \to 1$ (recall that in Example 3.2 we used the fd $R_1 : 1 \to 2$). We again claim that, given input $I^\star$ and $\mathbf{a}^\star = (a_1, a_2)$, the execution of $\mathsf{UniRel}\langle \mathbf{S}_0^\star, Q^\star\rangle(I^\star, \mathbf{a}^\star)$ returns an optimal solution. To see that, let $J_{\text{opt}}$ be an optimal solution. If none of the facts $R_2(c, a_2)$ that are $(Q^\star, \mathbf{a}^\star)$-useful in $I^\star$ are $Q$-useful in $J_{\text{opt}}$, we are done as in Example 3.2. Otherwise, take such a fact $R_2(c, a_2)$, and we will show that none of the facts $R_1(c', a_1)$ that are $(Q^\star, \mathbf{a}^\star)$-useful in $I^\star$ are $Q^\star$-useful in $J_{\text{opt}}$ (which means that $Q(J_{\text{opt}}) \subseteq Q(J_\phi)$ for $\phi = R_1(x, y_1)$). Indeed, if $R_1(c', a_1)$ is a counterexample, then $R_1 : 2 \to 1$ implies $c' = c$, since we know that $I^\star$ contains $R_1(c, a_1)$ (or otherwise $R_2(c, a_2)$ is not $(Q^\star, \mathbf{a}^\star)$-useful in $I^\star$), and then $\mathbf{a}^\star \in Q^\star(J_{\text{opt}})$, which is of course a contradiction. □

The optimality of the unirelation algorithm in Example 3.6 is due to the fact that the existential variable $x$ is functionally dependent on the head variables (i.e., $x \in img(\{y_1, y_2\})$), which means that given an answer in $Q(I^\star)$, every realizing assignment maps $x$ to the same value. In effect, this means that $x$ can be treated as a head (rather than existential) variable, which may change the structure of $\mathcal{G}_\exists(Q)$ and can potentially lead to functional head domination. Indeed, if $x$ becomes a head variable in $Q^\star$, then no existential variables remain in $Q^\star$, and then (functional) head domination trivially holds. Next, we formalize this intuition.

For a CQ $Q$ over a schema $\mathbf{S}$, we denote by $Q^+$ the CQ that is obtained by appending (in some order) all the existential variables in $\mathsf{Var}_\exists(Q) \cap img(\mathsf{Var_h}(Q))$ to the head of $Q$. As an example, for the CQ $Q = Q^\star$ over the schema $\mathbf{S}_1^\star$ of Example 3.6, the CQ $Q^+$ is:

$$Q^+(y_1, y_2, x) :- R_1(x, y_1), R_2(x, y_2)$$

The following lemma is straightforward.

Lemma 3.7. *Let* $\mathbf{S}$ *be a schema, and let* $Q$ *be a CQ over* $\mathbf{S}$. *Given an instance* $I$ *over* $\mathbf{S}$, *there is a function* $\beta : Q(I) \to Q^+(I)$, *such that* $\beta$ *is a bijection from* $Q(J)$ *to* $Q^+(J)$ *for every* $J \subseteq I$; *moreover,* $\beta$ *is computable in polynomial time in the size of* $I$.

As a conclusion of Lemma 3.7, we get the following.

Lemma 3.8. *Let* $Q$ *be a CQ over a schema* $\mathbf{S}$.

- *There is a polynomial-time reduction from the problem* $\text{FreeDP}\langle \mathbf{S}, Q\rangle$ *to the problem* $\text{FreeDP}\langle \mathbf{S}, Q^+\rangle$ *and vice versa.*
- *There is a polynomial-time, approximation-preserving reduction from* $\text{MaxDP}\langle \mathbf{S}, Q\rangle$ *to* $\text{MaxDP}\langle \mathbf{S}, Q^+\rangle$ *and vice versa.*

## 3.3 Tractability Condition

We can now define our tractability condition.

Definition 3.9. (**Tractability Condition**) For an sjf-CQ $Q$ over a schema $\mathbf{S}$, the *tractability condition* is that $Q^+$ has functional head domination. □

Later on, we show that the tractability condition indeed implies tractability. But first, the following theorem states that this condition is weaker than functional head domination, which in turn is weaker than head domination. The proof uses Theorem 4.2 (specifically, Part 3) that we present and discuss in the next section.

THEOREM 3.10. *Let $Q$ be a CQ over a schema $\mathbf{S}$.*
1. *If $Q$ has head domination, then $Q$ has functional head domination.*
2. *If $Q$ has functional head domination, then $Q^+$ has functional head domination.*

*Furthermore, none of the two reverse implications hold.*

Following is the main result for this section, and it says that when the tractability condition is satisfied (and in the absence of self joins), the unirelation algorithm is optimal.

THEOREM 3.11. *Let $Q$ be an sjf-CQ over a schema $\mathbf{S}$. If $Q^+$ has functional head domination, then $\mathsf{UniRel}\langle\mathbf{S}, Q\rangle$ optimally solves $\textsc{MaxDP}\langle\mathbf{S}, Q\rangle$ (in polynomial time).*

Next, we illustrate applications of Theorem 3.11.

EXAMPLE 3.12. Consider again the schema $\mathbf{S}$ and CQ $Q$ of Example 2.3, where it is stated that the existential variable $x$ is in $img(\{y_1, y_2\})$. So $x \in img(\mathsf{Var_h}(Q))$. The reader can verify that none of $x_1$ and $x_2$ are in $img(\mathsf{Var_h}(Q))$. Therefore, the CQ $Q^+$ is the following:

$$Q^+(y_1, y_2, y_3, x) :-$$
$$R_1(x_1, y_1), R_2(x_2, y_2), S(y_1, y_3, x), T(x, x_1, x_2)$$

Hence, $Q^+$ is actually the CQ $Q'$ of Example 3.4, where it is shown to have functional head domination. Therefore, the tractability condition is satisfied, and hence, Theorem 3.11 says that $\textsc{MaxDP}\langle\mathbf{S}, Q\rangle$ is optimally solvable in polynomial time by the unirelation algorithm. □

As another example, we will show how Theorem 2.5 (by Cong et al. [4]) follows from Theorem 3.11 for the class of sjf-CQs. Let $\mathbf{S}$ be a schema and let $Q$ be an sjf-CQ over $\mathbf{S}$, such that $Q$ is key preserving. It is straightforward to show that $Q$ being key preserving implies that $Q^+$ has no existential variables. Hence, $\mathcal{G}_\exists(Q)$ has no edges, and so $Q^+$ has (functional) head domination. Then, Theorem 3.11 says that $\textsc{MaxDP}\langle\mathbf{S}, Q\rangle$ is in polynomial time.

# 4. FD-SEPARATION

Before we proceed to proving the hardness side of our complexity dichotomy, we introduce the concept of *fd-separation*, which plays an important role in our proofs.

Let $Q$ be a CQ over a schema $\mathbf{S}$. The *variable co-occurrence graph* of $Q$, denoted $\mathcal{G}_\mathrm{v}(Q)$, is the graph that has $\mathsf{Var}(Q)$ as the set of nodes, and an edge $\{z_1, z_2\}$ whenever $z_1$ and $z_2$ co-occur in the same atom of $Q$ (i.e., $\{z_1, z_2\} \subseteq \mathsf{Var}(\phi)$ for some $\phi \in \text{atoms}(Q)$). Consider two variables $z_1, z_2 \in \mathsf{Var}(Q)$ and a subset $Z$ of $\mathsf{Var}(Q)$. We say that $Z$ *separates $z_1$ from $z_2$* if every path of $\mathcal{G}_\mathrm{v}(Q)$ connecting $z_1$ and $z_2$ visits one or more nodes in $Z$. Observe that as a special case, if $z_1 \in Z$ then $Z$ separates $z_1$ from every other variable $z_2$. Moreover, if $Z$ separates $z_1$ from itself then $z_1$ is necessarily in $Z$.

Let $Z$ be a subset of $\mathsf{Var}(Q)$, and let $z_1, z_2 \in \mathsf{Var}(Q)$ be two variables. We say that $Z$ *fd-separates $z_1$ from $z_2$* if $img(Z)$ separates $z_1$ from $z_2$. We also say that an atom $\phi$ of $Q$ *fd-separates $z_1$ from $z_2$* if $\mathsf{Var}(\phi)$ fd-separates $z_1$ from $z_2$. We denote by $fdSep(Z, z)$ (respectively, $fdSep(\phi, z)$) the set of all the variables $z'$ of $Q$, such that $Z$ (respectively, $\phi$) fd-separates $z$ from $z'$.

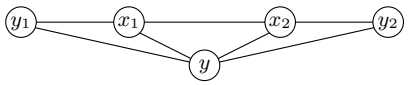


**Figure 3: $\mathcal{G}_\mathrm{v}(Q)$ in Example 4.1**

EXAMPLE 4.1. Let $\mathbf{S}$ be a schema with three ternary relation symbols $R_1$, $R_2$ and $S$, and the fd $S : \{1, 3\} \to 2$. Consider the following CQ over $\mathbf{S}$:

$$Q(y_1, y_2, y) :- R_1(y_1, y, x_1), S(x_1, y, x_2), R_2(x_2, y, y_2)$$

The graph $\mathcal{G}_\mathrm{v}(Q)$ is depicted in Figure 3. The singleton $\{x_1\}$ does not fd-separate $y_1$ from $y_2$, since the path $y_1 — y — y_2$ does not contain any node in $img(\{x_1\}) = \{x_1\}$. Similarly, $\{x_2\}$ does not fd-separate $y_1$ from $y_2$. However, $\{x_1, x_2\}$ fd-separates $y_1$ from $y_2$, since every path between $y_1$ and $y_2$ visits a node in $img(\{x_1, x_2\}) = \{x_1, x_2, y\}$. This also shows that $\phi_S$ fd-separates $y_1$ from $y_2$, or in other words, $y_2 \in fdSep(\phi_S, y_1)$. □

The following theorem states some facts on fd-separation. This theorem is important for this work, as it is used repeatedly in the proofs for the other results in this paper.

THEOREM 4.2. *Let $Q$ be a CQ over a schema $\mathbf{S}$, let $Z \subseteq \mathsf{Var}(Q)$ be a set of variables, and let $z \in \mathsf{Var}(Q)$ be a variable. The following hold.*
1. *$fdSep(Z, z)$ is closed under fds; that is:*
$$img(fdSep(Z, z)) = fdSep(Z, z)$$
2. *$fdSep(Z, z)$ is closed under fd-separation by subsets; that is, for all $Z' \subseteq \mathsf{Var}(Q)$ we have:*
$$Z' \subseteq fdSep(Z, z) \;\Rightarrow\; fdSep(Z', z) \subseteq fdSep(Z, z)$$
3. *$Z$ determines $z$ whenever $Z$ separates from $z$ a set that determines $z$; that is, for all $Z' \subseteq \mathsf{Var}(Q)$ we have:*
$$\big(Q : Z' \to z \wedge Z' \subseteq fdSep(Z, z)\big) \;\Rightarrow\; Q : Z \to z$$

# 5. HARDNESS

In this section, we discuss the proof of the hardness side of our complexity dichotomy: for an sjf-CQ $Q$ over a schema $\mathbf{S}$, if the tractability condition is violated, then $\textsc{FreeDP}\langle\mathbf{S}, Q\rangle$ is NP-complete. In Section 6 we discuss hardness of approximation for $\textsc{MaxDP}\langle\mathbf{S}, Q\rangle$. We will explain the main steps involved in the proof; actual proofs and intermediate steps are in the extended version of the paper [13].

Similarly to the proof of Theorem 2.9 by Kimelfeld et al. [14], the proof of hardness here is by a reduction from $\textsc{FreeDP}\langle\mathbf{S}^\star, Q^\star\rangle$; recall that $\mathbf{S}^\star$ and $Q^\star$ are defined in Example 2.1. But the reduction here is quite different from that of Theorem 2.9, due to the accommodation of functional dependencies. We first set some notation and assumptions that we will hold for the remainder of this section.

## 5.1 Notation and Assumptions

Throughout this section, we fix a schema $\mathbf{S} = (\mathcal{R}, \Delta)$ and an sjf-CQ $Q$ over $\mathbf{S}$, such that the tractability condition is violated: $Q^+$ does not have functional head domination. Membership of $\textsc{FreeDP}\langle\mathbf{S}, Q\rangle$ in NP is straightforward (nondeterministically choose a subinstance and test whether it is a side-effect-free solution). The goal is to prove that $\textsc{FreeDP}\langle\mathbf{S}, Q\rangle$ is NP-hard. By Lemma 3.8, it suffices

to prove that FREEDP$\langle \mathbf{S}, Q^+ \rangle$ is NP-hard. Therefore, to avoid carrying the plus superscript, we will simply assume that $Q = Q^+$. We fix a component $P$ of $\mathcal{G}_\exists(Q)$, such that none of the atoms $\phi$ of $Q$ satisfy $\mathsf{Var_h}(P) \subseteq img(\phi)$; such $P$ exists, since $Q$ does not have functional head domination.

For the sake of presentation, we rephrase $\mathbf{S}^\star$ and $Q^\star$ for a clear distinction from $\mathbf{S}$ and $Q$. Specifically, we will assume that the two relation symbols of $\mathbf{S}^\star$ are $R_1^\star$ and $R_2^\star$ (rather than $R_1$ and $R_2$ as in Example 2.1), and that the CQ $Q^\star$ over $\mathbf{S}^\star$ is the following sjf-CQ:

$$Q^\star(y_1^\star, y_2^\star) :- R_1^\star(x^\star, y_1^\star), R_2^\star(x^\star, y_2^\star)$$

It is known that FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$ is NP-complete [3]. Recall that $\mathbf{S}^\star$ has no fds. We will reduce FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$ to FREEDP$\langle \mathbf{S}, Q \rangle$. For the remainder of this section we fix the input $I^\star$ and $\mathbf{a}^\star$ for FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$. We assume that $\mathbf{a}^\star$ is the pair $(\lhd, \rhd)$. Our reduction will construct the input $I$ and $\mathbf{a}$ for the problem FREEDP$\langle \mathbf{S}, Q \rangle$.

EXAMPLE 5.1. A running example in this section is over the following CQ.

$$Q(y_1, y_2, y_3) :- S_1(y_1, x_1), R_1(x_1, x), R_2(x, x_2), S_2(x_2, y_2), T(x_1, x', x_2), U(x, x'', y_3), V(x_3, y_2, y_3)$$

The schema $\mathbf{S}$ has the relation symbols of $Q$ (with the corresponding arities). In addition, $\mathbf{S}$ has the following fds.

$$S_1 : 2 \to 1 \qquad S_2 : 1 \to 2 \qquad U : 1 \to 3$$

Observe that $Q = Q^+$ since no existential variable belongs to $img(\{y_1, y_2, y_3\})$. The graph $\mathcal{G}_\exists(Q)$ has two connected components, where one consists of just $V(x_3, y_2, y_3)$ and the other, which we fix here as $P$, consists of all remaining atoms. It is easy to verify that no atom $\phi$ of $Q$ is such that $img(\phi)$ contains all of $y_1$, $y_2$ and $y_3$ (i.e., the head variables of $P$); hence, the tractability condition is violated.

As example input for FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$, we use the instance $I^\star$ of Example 2.2. Recall that $\mathcal{M}(Q^\star, I^\star)$ is shown in the leftmost column of the table in Figure 4. For later use, the matches that produce $\mathbf{a}^\star = (\lhd, \rhd)$ are repeated in the bottom of the column. The reader can verify that there is no side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$. □

## 5.2 Template Construction

To reduce FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$ to FREEDP$\langle \mathbf{S}, Q \rangle$, we first define a *template construction*, which produces an instance $I_\Theta$ over $\mathbf{S}$. Intuitively, $I_\Theta$ *simulates* $I^\star$ in the sense that there is a direct correspondence between $Q^\star(I^\star)$ and $Q(I_\Theta)$, and between $\mathcal{M}(Q^\star, I^\star)$ and $\mathcal{M}(Q, I_\Theta)$. The challenge is to assure that *(1)* $I_\Theta$ satisfies the fds of $\mathbf{S}$ (Lemma 5.4), *and (2)* correspondence is achieved (Lemma 5.5). Next, we show how it is done. Still, $I_\Theta$ will not be the final $I$ of the reduction, since FREEDP$\langle \mathbf{S}, Q \rangle$ may be "too easy" on $I$. But $I_\Theta$ is a crucial step in the construction of $I$, and in fact, $I_\Theta$ will be a subinstance of $I$.

Recall that we are given the input $I^\star$ and $\mathbf{a}^\star = (\lhd, \rhd)$ for FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$. Also recall that $Q^\star$ uses three variables: $x^\star$, $y_1^\star$, $y_2^\star$. We fix a constant $\diamond$ that does not appear in $I^\star$. We first define a *template assignment* $\Theta$, which maps every variable $z \in \mathsf{Var}(Q)$ to a triple $\Theta(z) = \langle w, w_1, w_2 \rangle$, where $w$ is either the variable $x^\star$ or the constant $\diamond$, $w_1$ is either the variable $y_1^\star$ or the constant $\lhd$, and $w_2$ is either the variable $y_2^\star$ or the constant $\rhd$.

The template construction is parameterized by two variables $y_1, y_2 \in \mathsf{Var_h}(P)$ and two atoms $\phi_1, \phi_2 \in P$. Intuitively, each $y_i$ simulates $y_i^\star$, and the existential variables of $P$ simulate $x^\star$. Roughly speaking, $\phi_i$ "guards" $y_i$ from functional dependency on the variables that simulate the other two variables of $Q^\star$. In the next section, we discuss in detail the way these parameters are chosen (which is critical for the correctness of the reduction); for now, assume that they are already present. Given the parameters $y_1$, $y_2$, $\phi_1$ and $\phi_2$, the triple $\Theta(z)$, where $z \in \mathsf{Var}(Q)$, is constructed as follows.

- Begin with $\Theta(z) = \langle x^\star, y_1^\star, y_2^\star \rangle$.
- If $z \notin \mathsf{Var}_\exists(P)$, then replace $x^\star$ with $\diamond$.
- If $z \in fdSep(\phi_2, y_1)$, then replace $y_1^\star$ with $\lhd$.
- If $z \in fdSep(\phi_1, y_2)$, then replace $y_2^\star$ with $\rhd$.

Recall from Section 4 that $fdSep(\phi, y)$ is the set of all variables $z \in \mathsf{Var}(Q)$, such that $\phi$ fd-separates $y$ from $z$ (i.e., $img(\phi)$ separates $y$ from $z$ in the graph $\mathcal{G}_\mathrm{v}(Q)$).

EXAMPLE 5.2. We continue with our running example, which is introduced in Example 5.1. Figure 5 shows the graph $\mathcal{G}_\mathrm{v}(Q)$. The grey shadow should be ignored for now; we discuss it later in this section. Recall that $P$ consists of all the atoms of $Q$, except for $V(x_3, y_2, y_3)$.

Suppose that the parameters of the template construction are $y_1$, $y_2$, $\phi_1 = R_1(x_1, x)$ and $\phi_2 = R_2(x, x_2)$. (Later, we discuss the manner in which these specific parameters are chosen.) The sets $img(\phi_1)$ and $img(\phi_2)$ are surrounded by boxes with dotted edges. For each variable $z$ of $Q$, the corresponding node contains $\Theta(z)$ below $z$. As an example, consider the variable $x'$. As shown in the figure, $\Theta(x') = \langle x^\star, y_1^\star, y_2^\star \rangle$. The occurrence of $x^\star$ is due to the fact that $x'$ is an existential variable in $P$; that of $y_1^\star$ is due to the fact that $\phi_2$ does not fd-separate $x'$ from $y_1$, as evidenced by the path $x' — x_1 — y_1$, and similar is the explanation for $y_2^\star$.

As another example, note that $\Theta(x'') = \langle x^\star, \lhd, \rhd \rangle$, since $\phi_2$ (resp., $\phi_1$) fd-separates $x''$ from $y_1$ (resp., $y_2$), as the edges that are incident to $x''$ connect $x''$ to variables in the intersection $img(\phi_1) \cap img(\phi_2)$. Finally, for $x_3$ we have $\Theta(x_3) = \langle \diamond, \lhd, y_2^\star \rangle$, where the reason for the occurrence of $\diamond$ is that $x_3$ is not in $\mathsf{Var}_\exists(P)$; besides that, the explanations for $\lhd$ and $y_2^\star$ are like those for $x''$ and $x'$, respectively. □

Let $\mu$ be a match in $\mathcal{M}(Q^\star, I^\star)$. The assignment $\Theta_\mu$ is the same as $\Theta$, except that in the image, every occurrence of $x^\star$, $y_1^\star$ and $y_2^\star$ is replaced with $\mu(x^\star)$, $\mu(y_1^\star)$ and $\mu(y_2^\star)$, respectively. We call $\Theta_\mu$ an *instantiation* of $\Theta$. The *template construction* builds the instance $I_\Theta$ that comprises of all the $\Theta_\mu(\phi)$ over all $\mu \in \mathcal{M}(Q^\star, I^\star)$ and $\phi \in \mathrm{atoms}(Q)$.

EXAMPLE 5.3. We continue with our running example. For presentation sake, in our examples we may write just $cb_1b_2$ instead of $\langle c, b_1, b_2 \rangle$ (e.g., $\diamond\lhd\rhd$ instead of $\langle \diamond, \lhd, \rhd \rangle$).

Recall that Figure 5 shows the values that $\Theta$ assigns the variables to. Each of the top six rows in the table of Figure 4 corresponds to one of the six matches $\mu \in \mathcal{M}(Q^\star, I^\star)$. (We later discuss the bottom two rows.) The leftmost element (under "$\mathcal{M}(Q^\star, I^\star)$") is $\mu$, as explained in Example 5.1. The cells under the atom $\phi$ represent the facts $\Theta_\mu(\phi)$. As an example, if $\mu$ is the match represented by $\mathtt{0}, \lhd, \mathtt{b}$ and $\phi$ is $S_1(y_1, x_1)$, then $\Theta_\mu(\phi) = S(\diamond\lhd\rhd, \mathtt{0}\lhd\rhd)$. So, the top six rows in the table depict $I_\Theta$ (with some facts repeated). □

Next, we discuss some basic properties of the template construction, which hold regardless of the choice of the parameters $\phi_i$ and $y_i$. The following lemma states that $I_\Theta$ is

| $\mathcal{M}(Q^\star, I^\star)$ | $S_1(y_1, x_1)$ | $R_1(x_1, x)$ | $R_2(x, x_2)$ | $S_2(x_2, y_2)$ | $T(x_1, x', x_2)$ | $U(x, x'', y_3)$ | $V(x_3, y_2, y_3)$ | $Q(I)$ |
|---|---|---|---|---|---|---|---|---|
| $0, \lhd, \rhd$ | $\diamond\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, \diamond\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ |
| $0, \lhd, \mathtt{b}$ | $\diamond\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\mathtt{b}$ | $0\lhd\mathtt{b}, \diamond\lhd\mathtt{b}$ | $0\lhd\rhd, 0\lhd\mathtt{b}, 0\lhd\mathtt{b}$ | $0\lhd\rhd, 0\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\mathtt{b}, \diamond\lhd\mathtt{b}, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\mathtt{b}, \diamond\lhd\rhd$ |
| $0, \mathtt{a}, \rhd$ | $\diamond\mathtt{a}\rhd, 0\mathtt{a}\rhd$ | $0\mathtt{a}\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, \diamond\lhd\rhd$ | $0\mathtt{a}\rhd, 0\mathtt{a}\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\mathtt{a}\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ |
| $0, \mathtt{a}, \mathtt{b}$ | $\diamond\mathtt{a}\rhd, 0\mathtt{a}\rhd$ | $0\mathtt{a}\rhd, 0\lhd\rhd$ | $0\lhd\rhd, 0\lhd\mathtt{b}$ | $0\lhd\mathtt{b}, \diamond\lhd\mathtt{b}$ | $0\mathtt{a}\rhd, 0\mathtt{ab}, 0\lhd\mathtt{b}$ | $0\lhd\rhd, 0\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\mathtt{b}, \diamond\lhd\mathtt{b}, \diamond\lhd\rhd$ | $\diamond\mathtt{a}\rhd, \diamond\lhd\mathtt{b}, \diamond\lhd\rhd$ |
| $1, \lhd, \rhd$ | $\diamond\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, \diamond\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ |
| $1, \mathtt{c}, \rhd$ | $\diamond\mathtt{c}\rhd, 1\mathtt{c}\rhd$ | $1\mathtt{c}\rhd, 1\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, \diamond\lhd\rhd$ | $1\mathtt{c}\rhd, 1\mathtt{c}\rhd, 1\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ | $\diamond\mathtt{c}\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd$ |
| $0, \lhd, \rhd$ | $\diamond\lhd\rhd, 0\lhd\rhd$ | $0\lhd\rhd, \heartsuit_0^{\phi_T}$ | $\heartsuit_0^{\phi_T}, 0\lhd\rhd$ | $0\lhd\rhd, \diamond\lhd\rhd$ | $0\lhd\rhd, 0\lhd\rhd, 0\lhd\rhd$ | $\heartsuit_0^{\phi_T}, \heartsuit_0^{\phi_T}, \heartsuit_0^{\phi_T}$ | $\heartsuit_0^{\phi_T}, \diamond\lhd\rhd, \heartsuit_0^{\phi_T}$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \heartsuit_0^{\phi_T}$ |
| $1, \lhd, \rhd$ | $\diamond\lhd\rhd, 1\lhd\rhd$ | $1\lhd\rhd, \heartsuit_1^{\phi_T}$ | $\heartsuit_1^{\phi_T}, 1\lhd\rhd$ | $1\lhd\rhd, \diamond\lhd\rhd$ | $1\lhd\rhd, 1\lhd\rhd, 1\lhd\rhd$ | $\heartsuit_1^{\phi_T}, \heartsuit_1^{\phi_T}, \heartsuit_1^{\phi_T}$ | $\heartsuit_1^{\phi_T}, \diamond\lhd\rhd, \heartsuit_1^{\phi_T}$ | $\diamond\lhd\rhd, \diamond\lhd\rhd, \heartsuit_1^{\phi_T}$ |

**Figure 4: The instance $I$ constructed in the running example (introduced in Example 5.1)**

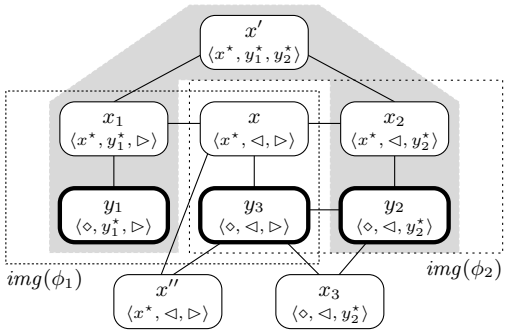


**Figure 5: $\mathcal{G}_{\mathrm{v}}(Q)$ in the running example (introduced in Example 5.1)**

an instance of $\mathbf{S}$ (i.e., all fds are satisfied). The proof uses Part 1 of Theorem 4.2.

LEMMA 5.4. *$I_\Theta$ satisfies every fd of $\Delta$.*

The next lemma, which is central to the correctness of our construction, states that there are no matches for $Q$ in $I_\Theta$ except for the instantiations of $\Theta$ (used for constructing $I_\Theta$).

LEMMA 5.5. $\mathcal{M}(Q, I_\Theta) = \{\Theta_\mu \mid \mu \in \mathcal{M}(Q^\star, I^\star)\}$.

The tuple $\mathbf{a}$ (in the input for FREEDP$\langle \mathbf{S}, Q\rangle$) is the one obtained from $\mathbf{y}$ by replacing each variable $y$ with the triple $\langle \diamond, \lhd, \rhd\rangle$. As as example, with our shorthand notation the tuple $\mathbf{a}$ in our running example is $(\diamond\lhd\rhd, \diamond\lhd\rhd, \diamond\lhd\rhd)$.

Up to now, we showed how to construct $I_\Theta$ and $\mathbf{a}$ from the input $I^\star$ and $\mathbf{a}^\star$ for FREEDP$\langle \mathbf{S}^\star, Q^\star\rangle$, given parameters for the template construction. For our reduction, can we can use $I_\Theta$ and $\mathbf{a}$ directly as input for FREEDP$\langle \mathbf{S}, Q\rangle$? As stated later in Lemma 5.7, the existence of a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ implies the existence of a side-effect-free solution for $I_\Theta$ and $\mathbf{a}$. But the other direction is not necessarily true: there may be a side-effect-free solution for $I_\Theta$ and $\mathbf{a}$ even if no such a solution exists for $I^\star$ and $\mathbf{a}^\star$.

EXAMPLE 5.6. Consider again the instances $I^\star$ and $I_\Theta$, and the tuples $\mathbf{a}^\star$ and $\mathbf{a}$, constructed in our running example. Recall that $I_\Theta$ is depicted in the top part of the table in Figure 4. Recall from Example 5.1 that there is no side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$. But *there is* a side-effect-free solution for $I$ and $\mathbf{a}$. To see that, observe (in Figure 4) that the facts $f_0 = T(0\lhd\rhd, 0\lhd\rhd, 0\lhd\rhd)$ and $f_1 = T(1\lhd\rhd, 1\lhd\rhd, 1\lhd\rhd)$ are used exactly in those matches that produce $\mathbf{a}$. Lemma 5.5 implies that Figure 4 has all the answers for $Q$ in $I_\Theta$. So, by removing $f_1$ and $f_2$ from $I_\Theta$ we get a solution that is side-effect free. □

The "problem" illustrated in Example 5.6 is that one of the atoms $\phi$, specifically $T(x_1, x', x_2)$, is such that $\Theta(\phi)$ contains both $y_1^\star$ and $y_2^\star$. Indeed, in the example both of $y_1^\star$ and $y_2^\star$ appear in $\Theta(x')$. An atom $\phi$ of $Q$ such that $\Theta(\phi)$ contains both $y_1^\star$ and $y_2^\star$ (not necessarily in the image of a single variable) is called an *encounter*. Following is a key lemma that will later guide us in the further definition of the reduction. This lemma shows that, under some condition, the reduction thus far is "correct" up to the issue of Example 5.6. That is, the existence of a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ implies the existence of a side-effect-free solution for $I_\Theta$ and $\mathbf{a}$, and on the other direction, the existence of a side-effect-free solution $J_\Theta$ for $I_\Theta$ and $\mathbf{a}$ implies the existence of a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ *provided that* $J_\Theta$ does not miss any of the facts over the encounters.

LEMMA 5.7. *Suppose that $P$ has two non-encounters $\gamma_1$ and $\gamma_2$, such that $\Theta(\gamma_1)$ contains $y_1^\star$ and $\Theta(\gamma_2)$ contains $y_2^\star$. Then, the following are equivalent.*

1. *There is a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$.*
2. *There is a side-effect-free solution $J_\Theta$ for $I_\Theta$ and $\mathbf{a}$, such that $R_\phi^{J_\Theta} = R_\phi^{I_\Theta}$ for all encounters $\phi$.*

In the next section, we branch into cases, based on properties of $\mathbf{S}$ and $Q$, and show how we build on Lemma 5.7 to complete the reduction in each of these cases.

## 5.3 Cases

We proceed by considering different cases, where in each case we specify the parameters for the template construction. We use the following terminology. A set $Z$ of variables is *simultaneously determined* if there is an atom $\phi$ of $Q$, such that $Z \subseteq img(\phi)$. Two variables $z_1$ and $z_2$ are *fd-separable* if there is an atom $\phi$ of $Q$ with the following properties:

- $img(\phi)$ contains neither $z_1$ nor $z_2$.
- $\phi$ fd-separates $z_1$ from $z_2$.

**Case 1:** *$P$ has two head variables $y_1$ and $y_2$ that are not simultaneously determined.*

The variables $y_1$ and $y_2$ will act as parameters for the template construction. With $y_1$ and $y_2$ chosen, we branch into two sub-cases, where in each sub-case we choose the parameters $\phi_1$ and $\phi_2$ differently.

**Case 1.a:** *$y_1$ and $y_2$ are fd-separable.*

In Case 1.a, we choose an atom $\phi$, such that $\phi$ fd-separates $y_1$ from $y_2$ and $img(\phi)$ contains neither $y_1$ nor $y_2$.

LEMMA 5.8. *In Case 1.a, $\phi \in P$.*

Now, both parameters $\phi_1$ and $\phi_2$ are chosen to be $\phi$. Recall that $\phi_1$ and $\phi_2$ are required to be in $P$, so this choice is justified by Lemma 5.8.

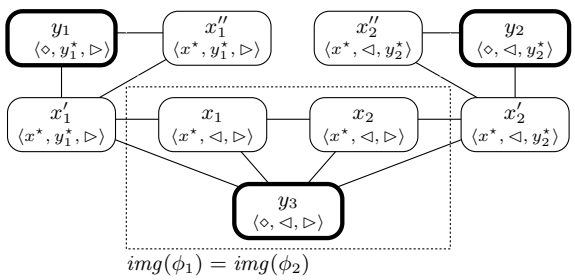


**Figure 6: Case 1.a—$\mathcal{G}_{\mathrm{v}}(Q)$ in Example 5.9**

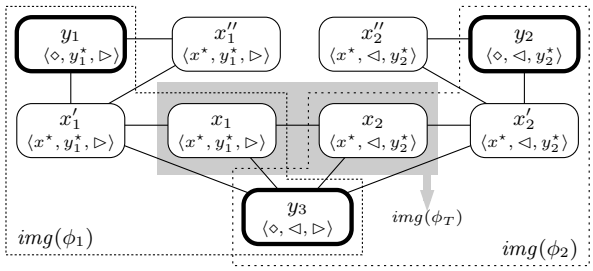


**Figure 7: Case 1.b—$\mathcal{G}_{\mathrm{v}}(Q)$ in Example 5.12**

Example 5.9. Consider the following CQ.

$$Q(y_1, y_2, y_3) :- S_1(x_1'', y_1, x_1'), R_1(x_1', y_3, x_1),$$
$$T(x_1, x_2), R_2(x_2, y_3, x_2'), S_2(x_2', y_2, x_2'')$$

The schema $\mathbf{S}$ has the relation symbols of $Q$ (with the corresponding arities). In addition, $\mathbf{S}$ has the following fds.

$$R_1 : 3 \to 2 \qquad S_1 : 3 \to 2 \qquad S_2 : 1 \to 2$$

It is easy to verify that $\mathcal{G}_{\exists}(Q)$ is connected (i.e., has just one connected component), and that no atom $\phi$ satisfies $\{y_1, y_2\} \subseteq img(\phi)$. Therefore, $y_1$ and $y_2$ are not simultaneously determined, and we are in Case 1. Figure 6 shows the graph $\mathcal{G}_{\mathrm{v}}(Q)$. Surrounded by a rectangle (with dotted edges) is the set $img(\phi_T)$, where $\phi_T$ is the atom $T(x_1, x_2)$. As can be seen in the figure, neither $y_1$ nor $y_2$ appears in $img(\phi_T)$, and $img(\phi_T)$ separates $y_1$ from $y_2$. Therefore, $y_1$ and $y_2$ are fd-separable, and we are in Case 1.a. As $\phi_1$ and $\phi_2$ we select the atom $\phi_T$ (actually, $\phi_T$ is the only possible choice). Finally, having chosen all the parameters of the template construction, the box of each variable $z$ in Figure 6 contains the triple $\Theta(z)$. □

Observe the following for the CQ $Q$ over the schema $\mathbf{S}$ in Example 5.9 (with $\mathcal{G}_{\mathrm{v}}(Q)$ depicted in Figure 6). First, none of the atoms are encounters. Second, $\Theta(y_1)$ contains $y_1^\star$ and $\Theta(y_2)$ contains $y_2^\star$. The following lemma shows that these observations always hold in Case 1.a. This lemma is useful, since it suffices for completing the reduction in Case 1.a.

Lemma 5.10. *In Case 1.a, $\Theta(y_i)$ contains $y_i^\star$ for $i = 1, 2$, and there are no encounters.*

So, for the instance $I$ of our reduction from the problem FreeDP$\langle \mathbf{S}^\star, Q^\star \rangle$ to FreeDP$\langle \mathbf{S}, Q \rangle$ we simply choose $I = I_\Theta$. Based on Lemma 5.10, we can now apply Lemma 5.7 with $\gamma_1$ and $\gamma_2$ being any atoms in $P$ that contain $y_1$ and $y_2$, respectively.

Corollary 5.11. *In Case 1.a, there is a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ if and only if there is a side-effect-free solution for $I$ and $\mathbf{a}$.*

Still within Case 1, the next case we consider is the complement of Case 1.a.

**Case 1.b:** *$y_1$ and $y_2$ are not fd-separable.*

In Case 1.b, we choose the atoms $\phi_1, \phi_2 \in P$ (used as parameters of the template construction) as follows. The atom $\phi_1$ is such that $y_1 \in img(\phi_1)$ (hence, $y_2 \notin img(\phi_1)$ since we are in Case 1) and $fdSep(\phi_1, y_2)$ has the maximal number of elements among the atoms $\phi \in P$ that satisfy $y_1 \in img(\phi)$. (Actually, it is enough for $fdSep(\phi_1, y_2)$ not to be strictly contained in any $fdSep(\phi, y_2)$ for those atoms $\phi$.) The atom $\phi_2$ is selected similarly, except that $y_1$ and $y_2$ swap roles. This manner of selecting $\phi_1$ and $\phi_2$ is crucial for the correctness of our reduction.

Example 5.12. Consider again the CQ $Q$ and schema $\mathbf{S}$ of Example 5.9, except that now we assume that the fds of $\mathbf{S}$ are the following.

$$R_1 : \{3, 2\} \to 1 \quad R_2 : \{1, 2\} \to 3 \quad S_1 : 3 \to 2 \quad S_2 : 1 \to 2$$

Like in Example 5.9, $y_1$ and $y_2$ are not simultaneously determined, which means that we are in Case 1. Figure 7 shows the graph $\mathcal{G}_{\mathrm{v}}(Q)$. Unlike Example 5.9, now the set $img(\phi_T)$ contains just $x_1$ and $x_2$, and as can be seen in Figure 7 the atom $\phi_T$ no longer fd-separates $y_1$ from $y_2$. Actually, it is easy to see that for each atom $\phi$ of $Q$ it holds that $img(\phi)$ either contains one of $y_1$ and $y_2$ or does not separate $y_1$ from $y_2$. We are therefore in Case 1.b.

The atoms $\phi$ with $y_1 \in img(\phi)$ are $\phi_{R_1}$ and $\phi_{S_1}$, namely, $R_1(x_1', y_3, x_1)$ and $S_1(x_1'', y_1, x_1')$, respectively. We have:

$$fdSep(\phi_{R_1}, y_2) = \{x_1'', y_1, x_1', x_1, y_3\}$$
$$fdSep(\phi_{S_1}, y_2) = \{x_1'', y_1, x_1'\}$$

Since $|fdSep(\phi_{R_1}, y_2)| > |fdSep(\phi_{S_1}, y_2)|$, we choose $\phi_1 = \phi_{R_1}$. We similarly choose $\phi_2 = \phi_{R_2}$. In Figure 7, the sets $img(\phi_1)$ and $img(\phi_2)$ are surrounded by polygons with dotted edges. Finally, with all parameters chosen, the box of each variable $z$ in Figure 7 contains the triple $\Theta(z)$. □

Note that in Example 5.12, the atom $\phi_T$ (i.e., $T(x_1, x_2)$) is an encounter, since $\Theta(x_1)$ contains $y_1^\star$ and $\Theta(x_2)$ contains $y_2^\star$. In particular, we cannot just set $I = I_\Theta$ as in Case 1.a: similarly to Example 5.6 it can be shown that there is always a side-effect-free solution for $I_\Theta$ and $\mathbf{a}$, no matter what $I^\star$ and $\mathbf{a}^\star$ are. In the following section, we deal with encounters for both this case and Case 2, which we discuss next.

**Case 2:** *Every two head variables of $P$ are simultaneously determined.*

In Case 2, the selection of the parameters for the template construction is more intricate than in the previous cases. For lack of space, here we do not provide the justification for this selection, and refer the reader to the full proof [13]. We first look at the sets $Z \subseteq \mathsf{Var_h}(P)$ that are simultaneously determined by an atom of $P$, and among those we select a $Z_1$ with a maximal cardinality. Note that $Z_1$ is a strict subset of $\mathsf{Var_h}(P)$, due to our assumption that $P$ violates

the tractability condition. Also note that $Z_1$ may be a strict subset of $\mathsf{Var_h}(\phi)$ for some atom $\phi \notin P$. Next, among those sets $Z$ that are *not* contained in $Z_1$, we again select a set $Z_2$ with a maximal cardinality. Note that neither $Z_1 \subseteq Z_2$ nor $Z_2 \subseteq Z_1$ holds. So, as $y_1$ we select an element of $Z_1 \setminus Z_2$, and as $y_2$ we select an element of $Z_2 \setminus Z_1$. Finally, we consider the atoms $\phi \in P$ with $img(\phi) \cap \mathsf{Var_h}(P) = Z_1$ and select $\phi_1$ to be a $\phi$ with a maximal $|fdSep(\phi, y_2)|$. Similarly, we consider the atoms $\phi \in P$ with $img(\phi) \cap \mathsf{Var_h}(P) = Z_2$, and select $\phi_2$ to be a $\phi$ with a maximal $|fdSep(\phi, y_1)|$. We now have all four parameters needed for the template construction.

EXAMPLE 5.13. An example of Case 2 is the CQ $Q$ over the schema $\mathbf{S}$ in our running example (introduced in Example 5.1). Indeed, $y_1$ and $y_2$ are simultaneously determined (by $\phi_T$), and so are $y_1$ and $y_3$ (by $\phi_{R_1}$) as well as $y_2$ and $y_3$ (by $\phi_{R_2}$). In a possible parameter setting, $Z_1$ is $\{y_1, y_3\}$, $Z_2$ is $\{y_2, y_3\}$, and the parameters are $y_1$, $y_2$, $\phi_{R_1}$ and $\phi_{R_2}$. Recall that Figure 5 shows $\mathcal{G}_{\mathrm{v}}(Q)$ along with $\Theta(z)$ for each $z \in \mathsf{Var}(Q)$. The sets $img(\phi_1)$ and $img(\phi_2)$ are surrounded by boxes with dotted edges. Observe that $T(x_1, x', x_2)$ is an encounter (since $\Theta(x')$ contains both $y_1^\star$ and $y_2^\star$). □

As shown in Example 5.13 (and previously in Example 5.6), in Case 2 there may be encounters, and we deal with those in the next section. Still, the following lemma states that for this case, as well as Case 1.b, we can use $\phi_1$ and $\phi_2$ as $\gamma_1$ and $\gamma_2$, respectively, in Lemma 5.7. Again, Part 1 of Theorem 4.2 is used in the proof.

LEMMA 5.14. *In Cases 1.b and 2, each $\phi_i$ ($i = 1, 2$) is a non-encounter, and $\Theta(\phi_i)$ contains $y_i^\star$.*

## 5.4 Handling Encounters

Our approach to handling encounters (in Cases 1.b and 2) is as follows. Let $\phi$ is an encounter. Denote by $\mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$ the set of all matches $\mu \in \mathcal{M}(Q^\star, I^\star)$ with $\mu(y_1^\star, y_2^\star) = \mathbf{a}^\star$. As illustrated in Example 5.6, we can always obtain a side-effect-free solution for $I_\Theta$ and $\mathbf{a}$ by deleting from $I_\Theta$ all the facts of the form $\Theta_\mu(\phi)$ where $\mu \in \mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$. The idea is to make each such $\Theta_\mu(\phi)$ necessary for a side-effect-free solution. This is done by adding to $I_\Theta$ facts so that in the resulting instance $I$, there are new answers for which every $\Theta_\mu(\phi)$ is needed. An idea in this spirit was used by Kimelfeld et al. [14], but details greatly differ as they did not handle fds. The addition of facts is such that if there is a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ then there is still a side-effect-free solution for $I$ and $\mathbf{a}$. Moreover, if $J$ is a side-effect-free solution for $I$ and $\mathbf{a}$, then $J$ necessarily contains all the facts $\Theta_\mu(\phi)$ where $\mu \in \mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$; we will then be able to assume that $J_\Theta = J \cap I_\Theta$ contains $R_\phi^{I_\Theta}$, and finally apply Lemma 5.7 to conclude the existence of a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$. Next, we give the details.

Let $\phi$ be an encounter, and let $\mu_c \in \mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$ be a match with $\mu_c(x^\star) = c$. We define the assignment $\nu_c^\phi$ for $Q$, as follows. In Case 1.b:

$$\nu_c^\phi(z) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mu_c}(z) & \text{if } z \in fdSep(\phi, y_1) \cap fdSep(\phi, y_2); \\ \heartsuit_c^\phi & \text{otherwise.} \end{cases}$$

And in Case 2:

$$\nu_c^\phi(z) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mu_c}(z) & \text{if } z \in \bigcap_{y \in \mathsf{Var_h}(P)} fdSep(\phi, y); \\ \heartsuit_c^\phi & \text{otherwise.} \end{cases}$$

Here, $\heartsuit_c^\phi$ is a fresh new constant that is unique to $\phi$ and $c$. Note that $z \in \bigcap_{y \in \mathsf{Var_h}(p)} fdSep(\phi, y)$ means that in $\mathcal{G}_{\mathrm{v}}(Q)$, the set $img(\phi)$ separates $z$ from every head variable $y$ of $P$.

As an example, consider $Q$ and $\mathbf{S}$ from Example 5.12, which fall in Case 1.b. There is a single encounter, namely $\phi_T = T(x_1, x_2)$. In Figure 7, the set $img(\phi_T)$ is shaded by a grey rectangle. Since every variable $z \notin \{x_1, x_2\}$ is reachable from $y_1$ (and $y_2$) by a path that does not visit $img(\phi_T)$, we have $\nu_c^{\phi_T}(z) = \heartsuit_c^{\phi_T}$ for $z \notin \{x_1, x_2\}$, while $\nu_c^{\phi_T}(z) = \Theta_{\mu_c}(z)$ for $z \in \{x_1, x_2\}$. Example 5.16 will discuss Case 2.

Finally, to construct $I$ we start with $I = I_\Theta$, and for each encounter $\phi$, assignment $\nu_c^\phi$ and atom $\gamma$ of $Q$, we add to $I$ the fact $\nu_c^\phi(\gamma)$ (unless it is already in $I$). The following lemma states that $I$ is a legal instance over $\mathbf{S}$. Again, the proof uses Part 1 of Theorem 4.2.

LEMMA 5.15. *In Cases 1.b and 2, $I$ satisfies each fd of $\mathbf{S}$.*

EXAMPLE 5.16. We continue with our running example, which falls in Case 2. There is a single encounter: $\phi_T = T(x_1, x', x_2)$. As shown in Figure 4, there are two matches in $\mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$: the one of the first row maps $x^\star$ to $\mathtt{0}$, and the one of the fifth row maps $x^\star$ to $\mathtt{1}$; let us call these matches $\mu_0$ and $\mu_1$, respectively. In Figure 5, the set $img(\phi_T)$ is shaded by a grey polytope. Since $img(\phi_T)$ does not separate any node from $y_3$, except for the nodes in $img(\phi_T)$ itself, we have $\nu_0^{\phi_T}(z) = \heartsuit_0^{\phi_T}$ and $\nu_1^{\phi_T}(z) = \heartsuit_1^{\phi_T}$ for $z \notin img(\phi_T)$, and $\nu_0^{\phi_T}(z) = \Theta_{\mu_0}(z)$ and $\nu_1^{\phi_T}(z) = \Theta_{\mu_1}(z)$ for $z \in img(\phi_T)$. The facts that are added to construct $I$ are in the bottom two rows of Figure 4. □

For proving the correctness of the reduction, we show that for all encounters $\phi$ and matches $\mu_c \in \mathcal{M}_{\mathbf{a}^\star}(Q^\star, I^\star)$, the mapping $\nu_c^\phi$ maps $\phi$ to $\Theta_{\mu_c}(\phi)$, and moreover, gives rise to a unique new answer that contains at least one occurrence of $\heartsuit_c^\phi$. (See Figure 4 for illustration.) In particular, $\Theta_{\mu_c}(\phi)$ is necessary for a side-effect-free solution. Based on that, we prove the following.

LEMMA 5.17. *In Cases 1.b and 2, these are equivalent:*
1. *There is a side-effect-free solution for $I$ and $\mathbf{a}$.*
2. *There is a side-effect-free solution $J_\Theta$ for $I_\Theta$ and $\mathbf{a}$, such that $R_\phi^{J_\Theta} = R_\phi^{I_\Theta}$ for all encounters $\phi$.*

The proof of Lemma 5.17 is fairly involved; moreover, Theorem 4.2 is used here again, now Parts 1 and 2. Finally, we get the correctness of the reduction (in Cases 1.b and 2) by combining Lemmas 5.7, 5.14, 5.15 and 5.17.

COROLLARY 5.18. *In Cases 1.b and 2, there is a side-effect-free solution for $I^\star$ and $\mathbf{a}^\star$ if and only if there is a side-effect-free solution for $I$ and $\mathbf{a}$.*

## 5.5 Summary and Conclusion

To summarize this section, we fixed $\mathbf{S}$ and $Q$ that violate the tractability condition, and showed a reduction from FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$ to FREEDP$\langle \mathbf{S}, Q \rangle$, taking the input $I^\star$ and $\mathbf{a}^\star$ for FREEDP$\langle \mathbf{S}^\star, Q^\star \rangle$ and constructing the input $I$ and $\mathbf{a}$ for FREEDP$\langle \mathbf{S}, Q \rangle$. We started with $I = I_\Theta$, which we built by means of the assignments $\Theta_\mu$, using the template assignment $\Theta$. To handle encounters, we added tuples to $I$ using the assignments $\nu_c^\phi$. The correctness of the reduction was stated in Corollaries 5.11 and 5.18. Combined with the observation that the construction of $I$ and $\mathbf{a}$ takes polynomial time, we get the main result of this section.

THEOREM 5.19. *If $Q^+$ does not have functional head domination, then* FREEDP$\langle \mathbf{S}, Q\rangle$ *is NP-complete.*

# 6. MAIN RESULT

In this section, we state the complexity dichotomy established in the previous sections, and further strengthen it by considering (hardness of) approximation.

## 6.1 Dichotomy

Combining Theorems 3.11 and 5.19, we get the main result of this paper, stating that for an sjf-CQ $Q$ over a schema $\mathbf{S}$ the complexities of MAXDP$\langle \mathbf{S}, Q\rangle$ and FREEDP$\langle \mathbf{S}, Q\rangle$ are precisely determined by the satisfaction of the tractability condition (i.e., $Q^+$ has functional head domination).

THEOREM 6.1. *Let $Q$ be an sjf-CQ over a schema $\mathbf{S}$.*

- *If $Q^+$ has functional head domination, then* UNIREL$\langle \mathbf{S}, Q\rangle$ *optimally solves* MAXDP$\langle \mathbf{S}, Q\rangle$ *(in polynomial-time).*
- *If $Q^+$ does not have functional head domination, then* FREEDP$\langle \mathbf{S}, Q\rangle$ *is NP-complete.*

## 6.2 Hardness of Approximation

For a number $\alpha \in [0, 1]$, a solution $J$ is *$\alpha$-optimal* if $|Q(J)| \geq \alpha \cdot |Q(K)|$ for all solutions $K$. Kimelfeld et al. [14] showed that for an sjf-CQ $Q$ over a schema $\mathbf{S}$ without fds, when MAXDP$\langle \mathbf{S}, Q\rangle$ is NP-hard, it cannot be approximated better than some constant $\alpha_Q \in (0, 1)$ unless P = NP. For that, they showed a PTAS reduction from MAXDP$\langle \mathbf{S}^\star, Q^\star\rangle$ to MAXDP$\langle \mathbf{S}, Q\rangle$ (i.e., the reduction is such that for all $\alpha$ there is $\alpha_Q$, such that an $\alpha_Q$-approximation for the generated MAXDP$\langle \mathbf{S}, Q\rangle$ instance gives an $\alpha$-approximation for the original MAXDP$\langle \mathbf{S}^\star, Q^\star\rangle$ instance), rather than a reduction from FREEDP$\langle \mathbf{S}^\star, Q^\star\rangle$ to FREEDP$\langle \mathbf{S}, Q\rangle$ as given here. Nevertheless, using ideas very similar to those in the construction of Kimelfeld et al. [14] (specifically, using multiple copies of the constant $\heartsuit_c^\phi$), we can adjust the reduction of Section 5 to a PTAS reduction.[1] As a result, we get the following addition to the hardness part of Theorem 6.1.

THEOREM 6.2. *Let $Q$ be an sjf-CQ over a schema $\mathbf{S}$. If $Q^+$ does not have functional head domination, then there is a constant $\alpha_Q$ such that it is NP-hard to $\alpha_Q$-approximate* MAXDP$\langle \mathbf{S}, Q\rangle$.

Combining Theorems 6.1 and 6.2, we conclude the following about MAXDP$\langle \mathbf{S}, Q\rangle$ for an sjf-CQ $Q$ over a schema $\mathbf{S}$. Either the problem is straightforward, or our ability to even approximate it is limited.

The dependency of the constant $\alpha_Q$ on $Q$ is already known to be unavoidable [14]: for each constant $\alpha \in (0, 1)$ there is an sjf-CQ $Q_\alpha$ over a schema $\mathbf{S}$ without fds, such that $Q_\alpha$ does not have (functional) head domination, and yet, MAXDP$\langle \mathbf{S}, Q\rangle$ is $\alpha$-approximable in polynomial time (and by the unirelation algorithm). In other words, there is no global $\alpha$ that can replace all the $\alpha_Q$ in Theorem 6.2.

[1]We gave a reduction from FREEDP$\langle \mathbf{S}^\star, Q^\star\rangle$ to FREEDP$\langle \mathbf{S}, Q\rangle$, rather than a PTAS one from MAXDP$\langle \mathbf{S}^\star, Q^\star\rangle$ to MAXDP$\langle \mathbf{S}, Q\rangle$, since it greatly simplifies the presentation, yet still, entails essentially all of the novelty.

# 7. CONCLUSIONS

The complexity dichotomy established in this paper shows how exactly functional dependencies extend the class of sjf-CQs where maximum deletion propagation is in polynomial time. On the negative side, it shows that if optimal solutions (or arbitrarily good approximations, e.g., PTAS) are sought, then even with functional dependencies there are no tractable sjf-CQs beyond those that the straightforward unirelation algorithm handles. Many natural directions are left for future work. Among them are exploring CQs with self joins (with or without fds), utilizing fds to generalize and improve upon approximation algorithms [14], handling additional constraints (like general equality-generating dependencies [2] and foreign keys), and extending our results to deletion of multiple tuples (a.k.a. group deletion [4]).


## Acknowledgments

The author is grateful to Jan Vondrák for helpful and insightful discussions, and to Phokion G. Kolaitis for being a source of education and inspiration for this work.


# 8. REFERENCES


[1] F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Trans. Database Syst.*, 6(4):557–575, 1981.

[2] C. Beeri and M. Y. Vardi. A proof procedure for data dependencies. *J. ACM*, 31(4):718–741, 1984.

[3] P. Buneman, S. Khanna, and W. C. Tan. On propagation of deletions and annotations through views. In *PODS*, pages 150–158, 2002.

[4] G. Cong, W. Fan, and F. Geerts. Annotation propagation revisited for key preserving views. In *CIKM*, pages 632–641, 2006.

[5] G. Cong, W. Fan, F. Geerts, J. Li, and J. Luo. On the complexity of view update analysis and its application to annotation propagation. *IEEE Trans. Knowl. Data Eng.*, 24(3):506–519, 2012.

[6] S. S. Cosmadakis and C. H. Papadimitriou. Updates of relational views. *J. ACM*, 31(4):742–760, 1984.

[7] Y. Cui and J. Widom. Run-time translation of view tuple deletions using data lineage. Technical report, Stanford University, 2001. `http://dbpubs.stanford.edu:8090/pub/2001-24`.

[8] N. N. Dalvi, K. Schnaitter, and D. Suciu. Computing query probability with incidence algebras. In *PODS*, pages 203–214, 2010.

[9] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[10] U. Dayal and P. A. Bernstein. On the correct translation of update operations on relational views. *ACM Trans. Database Syst.*, 7(3):381–416, 1982.

[11] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *PODS*, pages 352–365. ACM, 1983.

[12] A. M. Keller. Algorithms for translating view updates to database updates for views involving selections, projections, and joins. In *PODS*, pages 154–163. ACM, 1985.

[13] B. Kimelfeld. A dichotomy in the complexity of deletion propagation with functional dependencies (extended version). Accessible from the author's home page, 2012.

[14] B. Kimelfeld, J. Vondrák, and R. Williams. Maximizing conjunctive views in deletion propagation. In *PODS*, pages 187–198, 2011.

[15] P. G. Kolaitis and E. Pema. A dichotomy in the complexity of consistent query answering for queries with two atoms. In press, 2011.

[16] D. Maslowski and J. Wijsen. On counting database repairs. In *LID*, pages 15–22, 2011.

[17] A. Meliou, W. Gatterbauer, J. Y. Halpern, C. Koch, K. F. Moore, and D. Suciu. Causality in databases. *IEEE Data Eng. Bull.*, 33(3):59–67, 2010.

[18] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. The complexity of causality and responsibility for query answers and non-answers. *PVLDB*, 4(1):34–45, 2010.