REPLACE: A Logical Framework for Combining Collective Entity Resolution and Repairing

Meghyn Bienvenu[1] , **Gianluca Cima**[2] and **Víctor Gutiérrez-Basulto**[3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800
[2]Department of Computer, Control and Management Engineering, Sapienza University of Rome
[3]School of Computer Science & Informatics, Cardiff University
meghyn.bienvenu@labri.fr, cima@diag.uniroma1.it, gutierrezbasultov@cardiff.ac.uk

Abstract

This paper considers the problem of querying dirty databases, which may contain both erroneous facts and multiple names for the same entity. While both of these data quality issues have been widely studied in isolation, our contribution is a holistic framework for jointly deduplicating and repairing data. Our REPLACE framework follows a declarative approach, utilizing logical rules to specify under which conditions a pair of entity references can or must be merged and logical constraints to specify consistency requirements. The semantics defines a space of solutions, each consisting of a set of merges to perform and a set of facts to delete, which can be further refined by applying optimality criteria. As there may be multiple optimal solutions, we use classical notions of possible and certain query answers to reason over the alternative solutions, and introduce a novel notion of most informative answer to obtain a more compact presentation of query results. We perform a detailed analysis of the data complexity of the central reasoning tasks of recognizing optimal solutions and (most informative) possible and certain answers, for each of the three notions of optimal solution and for both general and restricted specifications.

1 Introduction

Data quality is one of the most fundamental problems in data management as dirty data can lead to incorrect decisions based on faulty retrieved answers from information systems, and unreliable data analysis, engendering huge costs to the private and public sectors [Fan and Geerts, 2012; Ilyas and Chu, 2019]. It is also a multi-faceted problem, encompassing several distinct issues: multiple representations of the same entity (deduplication), conflicting and/or erroneous information (consistency / accuracy), missing information (completeness), and outdated information (currency) [Fan and Geerts, 2012]. While far from solved, each facet of the data quality problem has given rise to a sizeable literature and increasingly sophisticated methods. We give a brief (and necessarily incomplete) introduction to the issues central to our work: deduplication and consistency.

Deduplication, also called record linkage or entity resolution (ER), was originally formulated as the task of identifying duplicate records in a table, and traditionally handled by comparing attribute values using similarity measures [Newcombe *et al.*, 1959]. Over time, however, variants of the problem have been explored, in which we may identify (match, merge) pairs of entity references or values, rather than whole tuples, and treat multiple tables and/or entity types together (so-called collective entity resolution [Bhattacharya and Getoor, 2007]), e.g. using a match of two authors infer that a pair of paper ids should be merged. Moreover, some recent approaches to collective ER [Deng *et al.*, 2022; Bienvenu *et al.*, 2022] are able to exploit recursive dependencies, e.g. a merge of authors may trigger merges of papers which in turn may trigger new author merges. Diverse techniques have been applied to (collective) ER, including similarity measures, deep learning, probabilistic formalisms, and declarative frameworks based upon logical rules and constraints, see [Christophides *et al.*, 2021] for a recent survey.

There is likewise a vast body of work aimed at identifying and removing conflicting facts to restore consistency. Declarative constraints (such as functional dependencies, or the broader class of denial constraints) are often employed to specify consistency requirements [Chu *et al.*, 2013], and the goal is to produce a consistent version of the data (called a repair) through deletion or modification of facts. However, due to lack of information, one typically cannot definitively identify the 'true' repair, so data cleaning often relies upon heuristics to produce a unique result [Ilyas and Chu, 2019]. By contrast, the well-known consistent query answering (CQA) paradigm [Arenas *et al.*, 1999; Bertossi, 2011] allows for meaningful answers to be obtained without committing to a single repair, by reasoning over the space of all (preferred) repairs and returning those answers which hold w.r.t. every such repair. The approach follows the skeptical mode of reasoning often considered in knowledge representation and reasoning (KR), and it has in turn inspired a line of KR research on inconsistency-tolerant ontology-based data access [Bienvenu, 2020; Lukasiewicz *et al.*, 2022]. While CQA has higher complexity than data cleaning methods, SAT-based implementations show promising results [Dixit and Kolaitis, 2022].

The different facets of the data quality problem have mostly been considered in isolation, whereas in practice, datasets can be expected to suffer from multiple data quality issues. A pipeline approach, which applies different methods in sequence, has the disadvantage that useful synergies may be missed, as noted in [Chu *et al.*, 2013; Fan *et al.*, 2014]. For example, by merging two constants, we may resolve a violation of a functional dependency (FD) without the need to delete facts, while conversely, by deleting incorrect facts, we may enable some desirable merges. The interest of combining ER with repairs has been advocated in [Fan *et al.*, 2014]: "When taken together, record matching and data repairing perform much better than being treated as separate processes". They propose to interleave repair operations (value updates) with merges of values inferred using matching dependencies, and study when this combined process terminates and how to generate a single repair of optimal cost.

We believe that the development of holistic approaches for jointly tackling the ER and repairing tasks, pioneered in [Fan *et al.*, 2014], merits further investigation. Indeed, given the vast number of different ER and repair methods, there are many options for which methods to use and how to integrate them. Moreover, to the best of our knowledge, no work has explored how to reason over a space of alternative solutions (in the spirit of CQA) for the combined task. These considerations motivate us to introduce REPLACE, a logic-based framework for collective entity resolution and repairing.

The REPLACE framework adopts the expressive class of denial constraints (which generalize the conditional FDs considered in [Fan *et al.*, 2014]) and subset repairs (obtained via deletion of facts, rather than updates), the most commonly considered repair notion in the CQA literature. The ER mechanism in REPLACE is based on the recently proposed LACE framework [Bienvenu *et al.*, 2022], which employs hard and soft rules to define mandatory and possible merges of constants. Differently from [Fan *et al.*, 2014] and other works using matching dependencies [Fan *et al.*, 2009; Bertossi *et al.*, 2013], the semantics is global in the sense that we merge *all* occurrences of the matched constants [Arasu *et al.*, 2009; Burdick *et al.*, 2016], rather than only those constant occurrences used in deriving the match. Such a semantics is geared towards merging of constants that are entity references (e.g. authors, publications), whereas the local one is more appropriate for merging attribute values (e.g. titles and addresses) (see [Bienvenu *et al.*, 2022] for a detailed discussion). REPLACE's semantics further follows the standard desiderata of maximizing merges and minimizing deletions. As these two criteria may conflict, REPLACE implements three natural ways to compare solutions: give priority to the maximization of merges (MER), give priority to the minimization of deletions (DEL), or adopt the Pareto principle (PAR).

Aside from introducing the new framework, our main contribution is the investigation of the data complexity of the main reasoning tasks associated with REPLACE. First, we show that the problem of recognizing optimal solutions is coNP-complete, for all three optimality notions. Next, we consider how to query the space of optimal solutions and determine the complexity of recognizing *certain* and *possible* query answers, i.e. those answers which hold in all or some optimal solution, respectively. The certain answer tasks are Π_2^p-complete for the three optimality notions, while for possible answers, the recognition problem Σ_2^p-complete for MER and DEL, but NP-complete for PAR. We further consider a restricted setting in which inequality atoms are disallowed in denial constraints. This restriction does not yield better complexity for these problems, if we consider the MER preorder, whereas for DEL and PAR, the complexity improves in almost all cases. As a further contribution, we introduce a novel notion of most informative answer to obtain a more compact presentation of query results and show that the improved format leads to a slight increase in the complexity of certain and possible answer recognition tasks. We conclude the paper with some directions for future work.

2 Preliminaries

A *(relational) schema* $\mathcal{S}$ is a finite set of relation symbols, with each $R \in \mathcal{S}$ having an associated arity and list of attributes. As is standard, we use R/k and $R(A_1, \dots, A_k)$ to indicate, respectively, that R has arity k and that its attributes are $A_1, \dots, A_k$. A *database instance over a schema* $\mathcal{S}$ (or *($\mathcal{S}$-)database* for short) assigns to each k-ary relation symbol $R \in \mathcal{S}$ a finite k-ary relation over a fixed, denumerable set of constants. Equivalently, we view an $\mathcal{S}$-database D as a finite set of *facts* of the form $R(c_1, \dots, c_k)$, where $(c_1, \dots, c_k)$ is a tuple of constants of the same arity as R. We use the notations $R(c_1, \dots, c_k) \in D$ and $D \subseteq D'$ with their obvious meanings. The *active domain* of a database D, denoted by $\mathsf{dom}(D)$, is the set of constants occurring in D.

When we speak of queries in this paper, unless otherwise stated, we mean a *conjunctive query (CQ)*. Recall that a *CQ over a schema* $\mathcal{S}$ takes the form $q(\mathbf{x}) = \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are disjoint lists of variables, and φ is a finite conjunction of relational atoms over $\mathcal{S}$, i.e. atoms of the form $R(t_1, \dots, t_k)$ with $R \in \mathcal{S}$ and each t_i is either a constant or a variable from $\mathbf{x} \cup \mathbf{y}$. The *arity* of a query $q(\mathbf{x})$ is the arity of $\mathbf{x}$, and a query with arity 0 is called *Boolean*. Given an n-ary query $q(x_1, \dots, x_n)$ and n-tuple of constants $\mathbf{c} = (c_1, \dots, c_n)$, we denote by $q[\mathbf{c}]$ the Boolean query obtained by replacing each x_i by c_i. The *answers to an n-ary query $q(\mathbf{x})$ over a database D* is defined as the set of n-tuples of constants $\mathbf{c}$ from $\mathsf{dom}(D)$ such that the Boolean CQ $q[\mathbf{c}]$ holds in D. We use $q(D)$ to denote the answers to q over D.

When formulating entity resolution rules, we will consider queries that may also contain atoms built from a set of externally defined binary *similarity predicates*. The preceding definitions and notations extend to such queries, the only difference being that similarity predicates have a fixed meaning (typically defined by applying a similarity metric, e.g. edit distance, and keeping those pairs of values whose score exceeds a given threshold).

Our framework will also make use of *denial constraints* [Bertossi, 2011; Fan and Geerts, 2012]. Recall that a *denial constraint over a schema* $\mathcal{S}$ takes the form $\forall \mathbf{x}. \neg(\phi(\mathbf{x}))$, where $\phi(\mathbf{x})$ is a finite conjunction of relational atoms over $\mathcal{S}$ and inequality atoms $t_1 \neq t_2$.

3 Existing LACE Framework

In this section, we recall the salient features and definitions of the LACE framework [Bienvenu *et al.*, 2022] for collective entity resolution, as it will form the basis for our new REPLACE framework, presented in Section 4.

Entity resolution consists in determining pairs of database constants that refer to the same entity and can thus be identified. We will use the term *merge* to speak about such pairs. The LACE framework employs hard and soft rules to indicate, respectively, required or potential merges. A *hard rule (w.r.t. a schema $\mathcal{S}$)* takes the form $q(x, y) \Rightarrow \mathsf{EQ}(x, y)$, where $q(x, y)$ is a CQ, whose atoms may use relation symbols in $\mathcal{S}$ as well as similarity predicates, and EQ is a special relation symbol (not in $\mathcal{S}$) used to store merges. Intuitively, such a rule states that (c_1, c_2) being an answer to q is sufficient to conclude that c_1 and c_2 refer to the same entity. A *soft rule* has a similar form: $q(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, but states instead that (c_1, c_2) being an answer to q provides reasonable evidence for c_1 and c_2 denoting the same entity. Soft rules suggest potential (but not mandatory) merges of constants. In what follows, we use the notation $q(x, y) \to \mathsf{EQ}(x, y)$ for a generic (hard or soft) rule, and shall omit quantifiers in rule bodies for brevity.

In addition to rules for generating merges, the LACE framework employs denial constraints to define consistency requirements. Together they form a specification:

Definition 1 ([Bienvenu *et al.*, 2022])**.** *A* data quality (DQ) specification Σ *over a schema $\mathcal{S}$ takes the form $\Sigma = \langle \Gamma, \Delta \rangle$, where $\Gamma = \Gamma_h \cup \Gamma_s$ is a finite set of hard and soft rules over $\mathcal{S}$, and Δ is a finite set of denial constraints over $\mathcal{S}$.*

Example 1. *Figure 1 introduces the schema $\mathcal{S}_{\mathsf{ex}}$, database D_{ex}, and DQ specification $\Sigma_{\mathsf{ex}} = \langle \Gamma_{\mathsf{ex}}, \Delta_{\mathsf{ex}} \rangle$ of our running example. Informally, the hard rule ρ_1 states that paper identifiers with similar titles, same year, same venue, same first author, and same conference chair must refer to the same paper. The soft rule σ_1 states that author ids associated with similar emails and the same institution likely refer to the same person. Finally, the denial constraint δ_1 enforces that there is a single chair for a given venue and year, while δ_2 states that the first author of a paper cannot be the same as the chair of the event where the paper was published.*

The semantics of LACE is based upon solutions, which take the form of equivalence relations over the constants, with the meaning that all constants from the same equivalence class are deemed to be references to the same entity. Solutions equate constants, rather than occurrences of constants, because LACE focuses specifically on merging constants that are entity references (e.g. paper and author ids). Intuitively, each solution is obtained by 'deriving' new merges via rule applications and closure operations. Importantly, rule bodies are evaluated on the database induced by previously derived merges, which makes it possible for new rules to become applicable, i.e. merges can enable additional merges.

In order to formally define solutions, we must first introduce some preliminary notions. Given a set S of pairs of constants from a database D, we denote by $\mathsf{EqRel}(S, D)$ the least equivalence relation $E \supseteq S$ over $\mathsf{dom}(D)$, i.e. we close S under reflexivity, symmetry, and transitivity. We assume that each equivalence relation E is equipped with a function rp_E that maps each element to a representative of its equivalence class in E. Given a database D and an equivalence relation E over $\mathsf{dom}(D)$, the *database induced by D and E*, denoted by D_E, is the database obtained from D by replacing each constant c by $\mathsf{rp}_E(c)$. Moreover, for a tuple $\mathbf{c}$ of constants (resp. query q, denial constraint δ), we denote by $\mathbf{c}_E$ (resp. q_E, δ_E) the tuple of constants (resp. query, denial constraint) obtained by replacing each constant c mentioned also in an equivalence relation E by $\mathsf{rp}_E(c)$. We then define the set $q(D, E)$ of *answers to a query $q(\mathbf{x})$ w.r.t. D and E* as:

$$\mathbf{c} \in q(D, E) \text{ iff } \mathbf{c}_E \in q_E(D_E)$$

A set of denial constraints Δ is satisfied in (D, E), written $(D, E) \models \Delta$, if $D_E \models \delta_E$ for every $\delta \in \Delta$. A rule $\gamma = q(x, y) \to \mathsf{EQ}(x, y) \in \Gamma$ is satisfied in (D, E), written $(D, E) \models \gamma$, if $q(D, E) \subseteq E$, and $(D, E) \models \Gamma'$ if all rules in $\Gamma' \subseteq \Gamma$ are satisfied. We call a pair (c, c') of constants *active in* (D, E) *w.r.t.* Γ if there exists a rule $q(x, y) \to \mathsf{EQ}(x, y) \in \Gamma$ such that $(c, c') \in q(D, E)$.

Remark 1. *There is in fact an additional syntactic condition placed on* LACE *rulesets (and which we shall adopt also in this paper), namely, that attributes that are involved in merges cannot participate in similarity atoms. We refer to [Bienvenu* et al.*, 2022] for a formal definition and discussion, simply noting that this condition ensures an unambiguous evaluation of similarity atoms in induced databases.*

We can now give the formal definition of LACE solutions:

Definition 2 ([Bienvenu *et al.*, 2022])**.** *Given a DQ specification Σ over a schema $\mathcal{S}$ and an $\mathcal{S}$-database D, we say that an equivalence relation E over* $\mathsf{dom}(D)$ *is an* ER candidate solution for (D, Σ) *if it satisfies one of the two conditions:*

(i) $E = \mathsf{EqRel}(\emptyset, D)$*;*

(ii) $E = \mathsf{EqRel}(E' \cup \{\alpha\}, D)$*, where E' is a candidate solution for (D, Σ) and $\alpha = (c_1, c_2)$ is active in (D, E') w.r.t. Γ.*

An ER solution *for (D, Σ) is a candidate solution E that further satisfies (a) $(D, E) \models \Gamma_h$ and (b) $(D, E) \models \Delta$. We denote by $\mathsf{ERSol}(D, \Sigma)$ the set of ER solutions for (D, Σ).*

Notice that each pair of constants that is deemed equivalent by the ER solution is obtained by a sequence of rule applications and closure operations. Moreover, solutions must be coherent in the sense that all of the hard rules and denial constraints have to be satisfied w.r.t. the induced database.

Example 2. *Continuing our running example, let D'_{ex} be the $\mathcal{S}_{\mathsf{ex}}$-database obtained from D_{ex} by removing the tuples regarding papers p_1, p_2, p_3, and p_4. Due to the tuples involving p_6, p_7, and p_8, we have $(D'_{\mathsf{ex}}, E_0) \not\models \delta_1$, for the initial relation $E_0 = \mathsf{EqRel}(\emptyset, D'_{\mathsf{ex}})$. However, we can resolve this violation by merging authors a_4 and a_5. Indeed, one can verify that $\epsilon = (a_4, a_5)$ is active in (D'_{ex}, E_0) w.r.t. Γ_{ex} due to σ_1. Also $\alpha = (a_1, a_2)$ and $\beta = (a_2, a_3)$ are active due to σ_1.*

However, we cannot include both α and β, otherwise by transitivity we would have $a_1 = a_3$, implying that the first author of paper p_5 would be the same as the chair, in violation of δ_2. Now, consider $E_1 = \mathsf{EqRel}(\{\beta, \epsilon\}, D'_{\mathsf{ex}})$ and $E_2 = \mathsf{EqRel}(\{\alpha, \epsilon\}, D'_{\mathsf{ex}})$. While $E_1 \in \mathsf{ERSol}(D'_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$, we have $E_2 \notin \mathsf{ERSol}(D'_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$. This is because $(D'_{\mathsf{ex}}, E_2) \not\models \rho_1$ since $\zeta = (p_6, p_7)$ is now active in (D'_{ex}, E_2) w.r.t. Γ_{ex}. One can verify that E_1 and $E_3 = \mathsf{EqRel}(\{\alpha, \epsilon, \zeta\}, D'_{\mathsf{ex}})$ are the

Author(aid, email, inst)

aid	email	inst
a_1	wtaka@gm.com	Tokyo
a_2	wtaka@tku.jp	Tokyo
a_3	takaw@tku.jp	Tokyo
a_4	mnk@ox.uk	NYU
a_5	mnk@gm.com	NYU
a_6	ropaolo@sap.it	Sap

Paper(pid, title, fid, year, venue, cid)

pid	title	fid	year	venue	cid
p_1	Computational Complexity of CQA	a_6	2009	IJCAI	a_1
p_2	CQA: Computational Complexity	a_6	2009	IJCAI	a_2
p_3	A Framework for Collective ER	a_1	2010	PODS	a_2
p_4	A Logical Framework for Collective ER	a_1	2010	PODS	a_2
p_5	Answering CQs over DL Ontologies	a_1	2012	KR	a_3
p_6	AI Techniques for ER	a_2	2023	AAAI	a_5
p_7	AI Techniques for Collective ER	a_1	2023	AAAI	a_4
p_8	Logical Techniques for Collective ER	a_3	2023	AAAI	a_5

$\delta_1 = \neg(\exists p, t, f, y, v, c, p', t', f', c'.\text{Paper}(p, t, f, y, v, c) \wedge \text{Paper}(p', t', f', y, v, c') \wedge c \neq c')$
$\delta_2 = \neg(\exists p, t, a, y, v.\text{Paper}(p, t, a, y, v, a))$
$\rho_1 = \text{Paper}(x, t, f, y, v, c) \wedge \text{Paper}(y, t', f, y, v, c) \wedge t \approx_1 t' \Rightarrow \mathsf{EQ}(x, y)$
$\sigma_1 = \text{Author}(x, e, i) \wedge \text{Author}(y, e', i) \wedge e \approx_2 e' \dashrightarrow \mathsf{EQ}(x, y)$

Figure 1: A schema $\mathcal{S}_{\mathsf{ex}}$, $\mathcal{S}_{\mathsf{ex}}$-database D_{ex}, and DQ specification $\Sigma_{\mathsf{ex}} = \langle \Gamma_{\mathsf{ex}}, \Delta_{\mathsf{ex}} \rangle$ over $\mathcal{S}_{\mathsf{ex}}$ with $\Gamma_{\mathsf{ex}} = \{\rho_1, \sigma_1\}$ and $\Delta_{\mathsf{ex}} = \{\delta_1, \delta_2\}$. The extension of the similarity predicates $\approx_1$ and $\approx_2$ (both restricted to $\mathsf{dom}(D_{\mathsf{ex}})$) are the symmetric and reflexive closures of $\{(e_1, e_2), (e_2, e_3), (e_4, e_5)\}$ and $\{(t_1, t_2), (t_3, t_4), (t_6, t_7), (t_7, t_8)\}$, respectively, where e_i and t_i are the email of author a_i and title of paper p_i, respectively.

only maximal ER solutions *for* $\mathsf{ERSol}(D'_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$*, i.e. they belong to* $\mathsf{ERSol}(D'_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *and there is no other solution in* $\mathsf{ERSol}(D'_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *containing strictly more merges.*

Now reconsider the original database D_{ex}*. One can verify that* $\mathsf{Sol}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \emptyset$*. This is because the tuples with* p_1 *and* p_2 *violate* δ_1*, and, if* a_1 *and* a_2 *are merged to solve this violation, then* δ_2 *is violated due to the tuples with* p_3 *and* p_4*.*

4 REPLACE: Adding Delete Operations

In practice, a given database may suffer from multiple data quality issues. Some constraint violations may result from the use of different constants for the same entity, and thus may be resolved through merging constants. However, other constraint violations stem from the presence of erroneous facts and can only be resolved by removing information. In this section, we introduce a holistic approach to data quality that allows for both merge and fact deletion operations. Our new REPLACE framework can be viewed as the marriage of LACE with the well-known consistent query answering approach.

Extending LACE with fact deletions allows us to obtain meaningful solutions when $\mathsf{ERSol}(D, \Sigma) = \emptyset$, but also to discover merges that were blocked due to constraint violations:

Example 3. *Recall the database* D'_{ex} *from the previous example and observe that by removing the fact with pid* p_5*, we can now include both* $\alpha = (a_1, a_2)$ *and* $\beta = (a_2, a_3)$ *in the set of merges, which will lead to* $\eta = (p_7, p_8)$ *being active.*

The REPLACE framework adopts the DQ specifications from LACE, but redefines what constitutes a solution to a database-specification (D, Σ) pair. In addition to an equivalence relation E that specifies merges, solutions will additionally contain a set R of facts to delete from D. We shall require that (*i*) $E \in \mathsf{ERSol}(D \setminus R, \Sigma)$, i.e. E is an ER solution for $(D \setminus R, \Sigma)$ and (*ii*) if a fact $\varphi \in R$ is equivalent to a fact $\psi \in D$ w.r.t. E, then $\psi \in R$. A fact $\varphi = P(\mathbf{c})$ is said to be *equivalent to* $\psi = P'(\mathbf{c}')$ *w.r.t.* E, denoted $\varphi \equiv_E \psi$, if $P = P'$ and $\mathbf{c}_E = \mathbf{c}'_E$.

We are now ready to formally define the new notion of solutions employed by REPLACE:

Definition 3. *Given a DQ specification* Σ *over a schema* $\mathcal{S}$ *and an* $\mathcal{S}$*-database* D*, we say that a pair* $W = (R, E)$ *is a* solution for (D, Σ) *if* (i) $R \subseteq D$*,* (ii) $E \in \mathsf{ERSol}(D \setminus R, \Sigma)$*, and* (iii) *for all* $\varphi, \psi \in D$ *with* $\varphi \equiv_E \psi$*,* $\varphi \in R$ *iff* $\psi \in R$*. We denote by* $\mathsf{Sol}(D, \Sigma)$ *the set of solutions for* (D, Σ)*.*

Example 4. *Let* $W_1 = (R_1, E_1)$ *and* $W_2 = (R_2, E_2)$ *be such that* R_1 *(resp.* R_2*) consists of the Paper fact with pid* p_1 *(resp.* p_2*),* $E_1 = \mathsf{EqRel}(\{\beta, \epsilon, \theta\}, (D_{\mathsf{ex}} \setminus R_1))$*, and* $E_2 = \mathsf{EqRel}(\{\beta, \epsilon, \theta\}, (D_{\mathsf{ex}} \setminus R_2)))$*, where* $\beta = (a_2, a_3)$*,* $\epsilon = (a_4, a_5)$*, and* $\theta = (p_3, p_4)$*. One can verify that* $W_1 \in \mathsf{Sol}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *and* $W_2 \in \mathsf{Sol}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$*.*

Rather than considering all solutions, it is natural to focus on the 'best' ones. But what makes a solution better than another? Similarly to LACE, we will prefer solutions that contain more merges, since we aim to tackle the ER problem. However, we also want to retain as much information as possible, hence should minimize fact deletions, as is done when defining repairs. These two criteria may conflict, as deleting more facts may enable more merges. This leads us to consider three natural ways to compare solutions: give priority to the maximization of merges (MER), give priority to the minimization of deletions (DEL), or adopt the Pareto principle and accord equal priority to both criteria (PAR). The following definition formalizes the three preorders for comparing solutions and the resulting notions of optimal solution, using set inclusion for comparing the sets of merges and deletions.

Definition 4. *Consider a DQ specification* Σ *over schema* $\mathcal{S}$ *and* $\mathcal{S}$*-database* D*. The preorders* $\prec_{\text{MER}}$*,* $\prec_{\text{DEL}}$*, and* $\prec_{\text{PAR}}$ *over* $\mathsf{Sol}(D, \Sigma)$ *are defined as follows:*

- $(R, E) \prec_{\text{MER}} (R', E')$ *iff either* (i) $E \subset E'$ *or* (ii) $E \subseteq E'$ *and* $R' \subset R$*;*
- $(R, E) \prec_{\text{DEL}} (R', E')$ *iff either* (i) $R' \subset R$ *or* (ii) $R' \subseteq R$ *and* $E \subset E'$*;*
- $(R, E) \prec_{\text{PAR}} (R', E')$ *iff either* (i) $E \subset E'$ *and* $R' \subseteq R$ *or* (ii) $R' \subset R$ *and* $E \subseteq E'$*.*

For $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we call a solution W for (D, Σ) an $\preceq_X$-optimal solution for (D, Σ) if there is no solution W' for (D, Σ) such that $W \prec_X W'$, and denote by $\mathsf{Sol}_X(D, \Sigma)$ the set of $\preceq_X$-optimal solutions for (D, Σ).

It is easy to verify that both $\mathsf{Sol}_{\textsc{Mer}}(D, \Sigma) \subseteq \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ and $\mathsf{Sol}_{\textsc{Del}}(D, \Sigma) \subseteq \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ hold for any database-specification pair (D, Σ). The next example shows that the converse inclusions do not necessarily hold. Furthermore, using analogous arguments, it is not hard to construct a case where $W \in \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ but neither $W \in \mathsf{Sol}_{\textsc{Mer}}(D, \Sigma)$ nor $W \in \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$.

Example 5. *Returning to our running example, it can be verified that W_1 and W_2 from Example 4 both belong to $\mathsf{Sol}_X(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$.*

Next consider $W_3 = (R_3, E_3)$, in which R_3 consists of the tuples with pids p_3 and p_4 and $E_3 = \mathsf{EqRel}(\{\alpha, \mu, \epsilon, \zeta, \}, (D_{\mathsf{ex}} \setminus R_3))$, where $\alpha = (a_1, a_2)$, $\mu = (p_1, p_2)$, $\epsilon = (a_4, a_5)$, and $\zeta = (p_6, p_7)$. One can show that $W_3 \in \mathsf{Sol}_{\textsc{Del}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ (hence, $W_3 \in \mathsf{Sol}_{\textsc{Par}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$) because the violation of δ_1 involving the p_1 and p_2 tuples is resolved by merging a_1 and a_2, rather than via deletion. We claim however that $W_3 \notin \mathsf{Sol}_{\textsc{Mer}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$. To see why, let $W_4 = (R_4, E_4)$ be such that R_4 contains the tuples with pids p_3, p_4, and p_5 and $E_4 = \mathsf{EqRel}(\{\alpha, \mu, \beta, \epsilon, \zeta, \eta\}, (D_{\mathsf{ex}} \setminus R_4))$, where $\beta = (a_2, a_3)$ and $\eta = (p_7, p_8)$. One can verify that $W_4 \in \mathsf{Sol}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ and $W_3 \prec_{\textsc{Mer}} W_4$ (while $W_4 \prec_{\textsc{Del}} W_3$ and W_3 and W_4 are incomparable w.r.t. the $\preceq_{\textsc{Par}}$ preorder).

Overall, we obtain the following: $\mathsf{Sol}_{\textsc{Par}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{W_1, W_2, W_3, W_4\}$, $\mathsf{Sol}_{\textsc{Del}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{W_1, W_2, W_3\}$, and $\mathsf{Sol}_{\textsc{Mer}}(D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{W_1, W_2, W_4\}$.

We conclude this section by situating REPLACE w.r.t. existing frameworks. First, observe that for any database-specification pair (D, Σ), we have $(\emptyset, E) \in \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$ iff $(\emptyset, E) \in \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ iff E is a *maximal ER solution* in the sense of [Bienvenu *et al.*, 2022, Definition 3]. Thus, the maximal solutions considered in LACE can be seen as special case of $\preceq_{\textsc{Del}}$- and $\preceq_{\textsc{Par}}$-optimal solutions. It is not hard to see that an analogous property does not hold for $\preceq_{\textsc{Mer}}$ preorder.

Next we relate REPLACE solutions with the subset repairs employed in consistent query answering. Consider any database-specification pair (D, Σ) such that $\Sigma = \langle \emptyset, \Delta \rangle$. Then, $\mathsf{Sol}_{\textsc{Mer}}(D, \Sigma)$, $\mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$, and $\mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ all coincide and contain only solutions of the form (R, trivE), where $\mathsf{trivE} = \{(c, c) \mid c \in \mathsf{dom}(D \setminus R)\}$. It is readily verified that $(R, \mathsf{trivE}) \in \mathsf{Sol}_{\textsc{Mer}}(D, \Sigma) = \mathsf{Sol}_{\textsc{Del}}(D, \Sigma) = \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ iff $D \setminus R$ is a repair in the sense of [Chomicki and Marcinkowski, 2005, Definition 2.2].

5 Reasoning about Solutions

In this section, we analyze the computational complexity of the central decision problems associated with the REPLACE framework, namely, checking whether a given set of merges and deletions is an (optimal) solution, and whether a candidate tuple is a certain or possible answer w.r.t. the space of optimal solutions. As is common when considering data-centric tasks, we employ the *data complexity* measure [Vardi, 1982], i.e. complexity is measured w.r.t. the size of the database D (and also the pair $W = (R, E)$ when it is part of the input).

Our results, summarized in Table 1, consider the three notions of optimality, as well as the impact of adopting a syntactically restricted form of specification (defined further).

5.1 Solution Recognition

We first consider the solution recognition problem (REC): given Σ, D, and W, decide whether $W \in \mathsf{Sol}(D, \Sigma)$. Tractability easily follows from the P-completeness of the analogous problem for ERSol [Bienvenu *et al.*, 2022]:

Theorem 1. REC *is* P-*complete.*

Next we determine the complexity of the problem X-OPTREC of deciding whether $W \in \mathsf{Sol}_X(D, \Sigma)$, where $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$ is the chosen optimality notion.

Theorem 2. *X-OPTREC is coNP-complete for any $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$.*

The upper bounds employ a guess-and-check approach, exploiting Theorem 1. We transferred an existing coNP lower bound for maximal ER solutions to X-OPTREC when $X \in \{\textsc{Del}, \textsc{Par}\}$, while MER-OPTREC required a new proof.

5.2 Query Answering

In an ideal world, we would determine which solution corresponds to the true data, and query the resulting clean instance. When this is infeasible, due to lack of time or knowledge, a reasonable approach is to query the space of optimal solutions to identify those tuples that are answers w.r.t. every solution (in line with CQA semantics and the *skeptical* mode of inference employed in non-monotonic reasoning) or at least one solution (a form of *credulous / brave* reasoning).

This leads us to define the following notions of certain and possible answers. Note that given a solution $W = (R, E)$ to (D, Σ), we shall use $q(D, W)$ to refer to $q(D \setminus R, E)$.

Definition 5. *Given a DQ specification Σ, database D, and query q, all over schema $\mathcal{S}$, and $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we say that a tuple $\mathbf{c}$ of constants is an X*-certain (resp. X-possible) answer to q on D w.r.t. Σ *if $\mathbf{c} \in q(D, W)$ for every (resp. some) $W \in \mathsf{Sol}_X(D, \Sigma)$. We use X-$\mathsf{certAns}(q, D, \Sigma)$ and X-$\mathsf{possAns}(q, D, \Sigma)$ to denote, respectively, the set of X-certain answers and X-possible answers to q on D w.r.t. Σ.*

Example 6. *First consider the query $q^1_{\mathsf{ex}}(x, y, z) = \exists t, v, e, m.\mathit{Paper}(x, t, y, 2023, v, e) \wedge \mathit{Author}(y, m, z)$, which returns the id of papers written in 2023 along with the institution and the id of its first author. For the tuple $\mathbf{t} = (p_6, a_3, \mathit{Tokyo})$, we have the following:*

- $\mathbf{t} \in \textsc{Mer}$-$\mathsf{certAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$;
- $\mathbf{t} \notin \textsc{Del}$-$\mathsf{certAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$, *as* $\mathbf{t} \notin q^1_{\mathsf{ex}}(D_{\mathsf{ex}}, W_3)$, *hence also* $\mathbf{t} \notin \textsc{Par}$-$\mathsf{certAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$;
- $\mathbf{t} \in X$-$\mathsf{possAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ ($X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$).

Next let $q^2_{\mathsf{ex}}(x, y) = \exists t, f, v, m, i.\ \mathit{Paper}(x, t, f, 2012, v, y) \wedge \mathit{Author}(y, m, i)$ be the query that returns the ids of papers written in 2012 and the venue chair. Observe that X-$\mathsf{certAns}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \emptyset$ for $X \in \{\textsc{Mer}, \textsc{Par}\}$, while $\textsc{Del}$-$\mathsf{certAns}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{(p_5, a_3)\}$. Notice moreover that X-$\mathsf{possAns}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{(p_5, a_2), (p_5, a_3)\}$ for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$.

Specifications	X	X-OPTREC	X-CERTANS	X-POSSANS	X-MICERTANS	X-MIPOSSANS
General	MER/DEL	coNP-c	Π_2^p-c	Σ_2^p-c	DP_2-c	DP_2-c
	PAR	coNP-c	Π_2^p-c	NP-c	DP_2-c	DP-c
Restricted	MER	coNP-c	Π_2^p-c	Σ_2^p-c	DP_2-c	DP_2-c
	DEL/PAR	P-c	coNP-c	NP-c	DP-c	DP-c

Table 1: Data complexity of the decision problems, parameterized by $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$. We use '-c' as an abbreviation for '-complete'.

Our next theorem provides the complexity of the decision problems X-CERTANS and X-POSSANS of checking whether a given tuple of constants belongs to the set of X-certain answers and X-possible answers, respectively. We remind the reader that whenever we speak of queries we refer to CQs.

Theorem 3. *X-CERTANS is Π_2^p-complete for any $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, PAR-POSSANS is NP-complete, and X-POSSANS is Σ_2^p-complete for $X \in \{\text{MER}, \text{DEL}\}$.*

The Π_2^p and Σ_2^p membership proofs involve guessing a potential solution W that contains / omit the query tuple and calling an NP oracle to check that W is indeed an optimal solution. The NP upper bound for PAR-POSSANS relies upon showing that it is sufficient to check that W is a solution, rather than a PAR-optimal solution. This is because $c \in q(D, W)$ implies $c \in q(D, W')$ for any W' such that $W \prec_{\text{PAR}} W'$ (no such property holds for $\prec_{\text{MER}}$ and $\prec_{\text{DEL}}$). While some lower bounds were adapted from analogous results for LACE, others require new ingredients.

5.3 Restricted Specifications

The preceding results show that it is computationally challenging to reason about optimal solutions. Faced with a similar situation, Bienvenu *et al.* (2022) explored *restricted DQ specifications*, in which inequality atoms are disallowed in the denial constraints. While such specifications cannot capture keys and functional dependencies, they do allow for other meaningful forms of constraints, e.g. class and property disjointness statements commonly used for Semantic Web data.

Do restricted DQ specifications yield better complexity in our setting? For REC, MER-OPTREC, MER-CERTANS, MER-POSSANS, and PAR-POSSANS, the answer is no, as the lower bound proofs employ restricted DQ specifications. However, for the remaining decision problems, we do find a drop in complexity (under the usual complexity assumptions).

Theorem 4. *For restricted DQ specifications, we have that:*

- DEL-OPTREC *and* PAR-OPTREC *are* P*-complete;*
- X-CERTANS *is coNP-complete for* $X \in \{\text{DEL}, \text{PAR}\}$ *and* DEL-POSSANS *is NP-complete;*

Intuitively, this lower complexity is due to constraint violations being preserved under improvements, i.e. if δ is a denial constraint without $\neq$-atoms and both $W = (R, E)$ and $W' = (R', E')$ belong to $\mathsf{Sol}(D, E)$, then $(D \setminus R)_E \not\models \delta$ implies $(D \setminus R')_{E'} \not\models \delta$ whenever $E \subseteq E'$ and $R' \subseteq R$.

5.4 Comparison with LACE and CQA

Comparing with LACE, we note that in almost all cases, the addition of delete operations does not affect the complexity of recognizing (maximal / optimal) solutions or certain and possible answers. The main exception is if we consider $\prec_{\text{MER}}$-optimal solutions coupled with restricted specifications, where all problems are one level higher in the polynomial hierarchy than the corresponding problems in LACE.

Adding merges to CQA brings a notable increase in complexity. Indeed, the certain query answering and optimal solution recognition tasks are one level higher than the corresponding CQA and repair checking tasks, if one considers general specifications or restricted specifications with the $\prec_{\text{MER}}$ preorder. An even larger complexity jump is observed for possible query answering, as the analogous task w.r.t. repairs is easily seen to have polynomial data complexity.

6 Most Informative Answers

While our notions of certain and possible answers (and the corresponding notions in [Bienvenu *et al.*, 2022]) provide a natural way of querying the space of optimal solutions, they present one major drawback from an end user's perspective: the query results may contain multiple distinct tuples that are *equivalent w.r.t. the considered solutions*, as illustrated next.

Example 7. *Consider a scenario in which we have the database-specification pair (D, Σ), the database D contains facts $P(c_1, c_2)$ and $P(c_3, c_4)$, and $W = (\emptyset, E)$ with $E = \mathsf{EqRel}(\{(c_1, c_3), (c_2, c_4)\}, D)$ is the only $\preceq_X$-optimal solution for (D, Σ), for every $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$. Then, for the query $q(x_1, x_2) = P(x_1, x_2)$ and for any $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, we have four X-certain/possible answers to q on D w.r.t. Σ, namely: (c_1, c_2), (c_1, c_4), (c_3, c_2), and (c_3, c_4). These tuples could be more concisely presented as a a single tuple of* sets of constants $(\{c_1, c_3\}, \{c_2, c_4\})$.

To address this issue and present query results with as much information (and as little repetition) as possible, we introduce the new notions of *most informative (certain / possible) answers*. The main idea, evoked in the example, that answers to queries now consist of tuples of *sets* of constants, each set comprising constants in the same equivalence relation w.r.t. the solution(s) under consideration.

Definition 6. *Given a solution $W = (R, E)$ for (D, Σ), an n-ary query q, and an n-tuple $\mathbf{C} = (C_1, \ldots, C_n)$ of sets of constants from D, we call $\mathbf{C}$ a* set-answer to q on D w.r.t. W *if the following holds:* (i) *C_i contains constants in the same equivalence class in E, for $1 \leq i \leq n$, and* (ii) *there exists a tuple of constants $\mathbf{c} = (c_1, \ldots, c_n) \in q(D, W)$ such that $c_i \in C_i$ for every $1 \leq i \leq n$. We denote by $\overline{q}(D, W)$ the set of set-answers to q on D w.r.t. W.*

Definition 7. *For $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, we say that a tuple $\mathbf{C}$ of sets of constants is a X-*certain set-answer *(resp. X-*

possible set-answer*)* to q on D w.r.t. Σ *if* $\mathbf{C} \in \overline{q}(D, W)$ *for every (resp. some)* $W \in \mathsf{Sol}_X(D, W)$. *We use* X-$\mathsf{SetCert}(q, D, \Sigma)$ *(resp.* X-$\mathsf{SetPoss}(q, D, \Sigma)$*) for the set of* X*-certain (resp.* X*-possible) set-answers to* q *on* D *w.r.t.* Σ.

Example 8. *Recall the queries* q^1_{ex} *and* q^2_{ex} *from Example 6. The tuple* $\mathbf{T} = (\{p_6\}, \{a_2, a_3\}, \{Tokyo\}) \in$ MER-$\mathsf{SetCert}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *while* $\mathbf{T} \notin X$-$\mathsf{SetCert}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *for both* $X =$ DEL *and* $X =$ PAR.

As another example, we have that $(\{p_5\}, \{a_2, a_3\}) \in X$-$\mathsf{SetPoss}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *for each* $X \in \{$MER, DEL, PAR$\}$.

Among the X-certain and X-possible set-answers, we are interested in presenting the *most informative* ones. More formally, for two n-tuples $\mathbf{C} = \langle C_1, \ldots, C_n \rangle$ and $\mathbf{C}' = \langle C'_1, \ldots, C'_n \rangle$ of sets of constants, we say that $\mathbf{C}'$ is *strictly more informative* than $\mathbf{C}$ if (*i*) $C_i \subseteq C'_i$ for every $1 \leq i \leq n$, and (*ii*) $C_i \subset C'_i$ for some $1 \leq i \leq n$. Given a set S of n-tuples of sets of constants, we say that $\mathbf{C} \in S$ is *most informative in* S if there is no $\mathbf{C}' \in S$ that is strictly more informative than $\mathbf{C}$. With these notions in hand, we can now formally define most informative certain and possible answers.

Definition 8. *Given a DQ specification* Σ*, database* D*, and query* q*, all over schema* $\mathcal{S}$*, and* $X \in \{$MER, DEL, PAR$\}$*, we say that a tuple* $\mathbf{C}$ *of sets of constants from* D *is a* most informative X-certain answer *(resp.* most informative X-possible answer*) to* q *on* D *w.r.t.* Σ *if* $\mathbf{C}$ *is most informative in* X-$\mathsf{SetCert}(q, D, \Sigma)$ *(resp. in* X-$\mathsf{SetPoss}(q, D, \Sigma)$*). We denote by* X-$\mathsf{MIcertAns}(q, D, \Sigma)$ *(resp.* X-$\mathsf{MIpossAns}(q, D, \Sigma)$*) the set of most informative* X*-certain (resp.* X*-possible) answers to* q *on* D *w.r.t.* Σ.

Example 9. *Observe that* $\mathbf{T_1} = (\{p_6\}, \{a_2\}, \{Tokyo\}) \in X$-$\mathsf{MIcertAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *for* $X \in \{$DEL, PAR$\}$*, while* $\mathbf{T_1} \notin$ MER-$\mathsf{MIcertAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *because the tuple* $\mathbf{T}_0 = (\{p_6\}, \{a_2, a_3\}, \{Tokyo\})$ *in* MER-$\mathsf{SetCert}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *is strictly more informative than* $\mathbf{T_1}$*. In fact,* $\mathbf{T}_0 \in$ MER-$\mathsf{MIcertAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$*. Analogously, one can see that* $\mathbf{T_2} = (\{p_6, p_7\}, \{a_1, a_2\}, \{Tokyo\})$ *is such that* $\mathbf{T_2} \in X$-$\mathsf{MIpossAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *for both* $X =$ DEL *and* $X =$ PAR*, while* $\mathbf{T_2} \notin$ MER-$\mathsf{MIcertAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *because* $\mathbf{T_3} = (\{p_6, p_7, p_8\}, \{a_1, a_2, a_3\}, \{Tokyo\})$ *occurs in* MER-$\mathsf{SetPoss}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$ *and is strictly more informative than* $\mathbf{T_2}$*. In fact,* $\mathbf{T_3} \in$ MER-$\mathsf{MIpossAns}(q^1_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}})$.

For query q^2_{ex}*,* X-$\mathsf{MIcertAns}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \emptyset$ *for* $X \in \{$MER, PAR$\}$ *while* DEL-$\mathsf{MIcertAns}(q^2_{\mathsf{ex}}, D_{\mathsf{ex}}, \Sigma_{\mathsf{ex}}) = \{(\{p_5\}, \{a_3\})\}$*. As for possible answers,* X-$\mathsf{MIpossAns} = \{(\{p_5\}, \{a_2, a_3\})\}$ *for each* $X \in \{$MER, DEL, PAR$\}$.

While X-$\mathsf{certAns}(q, D, \Sigma) \subseteq X$-$\mathsf{possAns}(q, D, \Sigma)$, the inclusion X-$\mathsf{MIcertAns}(q, D, \Sigma) \subseteq X$-$\mathsf{MIpossAns}(q, D, \Sigma)$ does not hold in general. However, we have the following related property: if $\mathbf{C} \in X$-$\mathsf{MIcertAns}(q, D, \Sigma)$, then either $\mathbf{C} \in X$-$\mathsf{MIpossAns}(q, D, \Sigma)$ or there exists $\mathbf{C}' \in X$-$\mathsf{MIpossAns}(q, D, \Sigma)$ that is strictly more informative than $\mathbf{C}$.

We now consider the decision problems X-MIPOSSANS and X-MIPOSSANS of checking whether a given tuple of sets of constants is a most informative X-certain (respectively, X-possible) answer. We find that adopting the most informative notions of answers leads to higher complexity compared to the (plain) notions of certain and possible answers.

Theorem 5. X-MICERTANS *is* DP_2*-complete*[1] *for any* $X \in \{$MER, DEL, PAR$\}$*,* X-MIPOSSANS *is* DP_2*-complete for* $X \in \{$MER, DEL$\}$*, and* PAR-MIPOSSANS *is DP-complete.*

The upper bounds rely on the fact that the set of yes-instances for X-MICERTANS is precisely the intersection of the yes-instances of X-SETCERT (decide whether $\mathbf{C} \in X$-$\mathsf{SetCert}(q, D, \Sigma)$) and X-NOBETTERCERT (decides whether there is no $\mathbf{C}' \in \mathsf{SetCert}(q, D, \Sigma)$ strictly more informative than $\mathbf{C}$). The latter can be solved by guessing a solution W_i for each possible 'minimal improvement' $\mathbf{C}_i$ of $\mathbf{C}$ and verifying that $\mathbf{C}_i \notin q(D, W_i)$. Similar considerations apply for X-MIPOSSANS. Lower bounds rely on new reductions.

We conclude this section by considering the impact of adopting restricted specifications. While MER-MICERTANS, PAR-MIPOSSANS, and MER-MICERTANS retain their original complexity, the other problems enjoy lower complexity.

Theorem 6. *For restricted DQ specifications, the decision problems* DEL-MICERTANS*,* PAR-MICERTANS*, and* DEL-MIPOSSANS *are DP-complete.*

7 Conclusion and Future Work

We presented REPLACE, a new holistic framework for (possibly recursive) collective entity resolution and repairing, which employs denial constraints coupled with (hard and soft) logical rules to infer merges. The semantics, based upon solutions that take the form of (coherent) sets of merges and deletions, generalizes both LACE and (subset) repairs. In the spirit of CQA, we studied how to query the space of (optimal) solutions. Our complexity analysis shows that while certain and possible answer recognition is harder than the analogous tasks for repairs, it is for the most part on par with existing results for LACE. We also explored an important question (not considered in LACE) of how to present the query results, which is non-trivial due to the merged constants, leading us to propose novel notions of most informative answers.

We view this work as a starting point, with many interesting questions left to explore. First, it could be natural to consider other reasoning tasks, such as identifying certain and possible merges and deletions, which could help guide users towards a unique solution. While some results can be transferred from query answering, other cases require further study. Next, we believe it would be interesting to explore extensions of REPLACE with quantitive information (weight or scores) associated to rules and facts, in particular, so that approximate weighted solutions could be generated. Finally, we would like to develop an efficient prototype based on logic-based technologies, such as answer set programming (ASP) [Gebser *et al.*, 2012]. To this end, we could use LACE's ASP encoding as a steppingstone, but new insights will be needed to handle most informative certain answers, whose DP_2 complexity goes beyond what is traditionally supported by ASP. It would also be interesting to use more informative similarity measures by adding ML predicates, in the style of [Deng *et al.*, 2022].

[1]We recall that the complexity classes DP (a.k.a. BH(2)) and DP_2 (a.k.a. $BH_3(2)$) are the second level of the Boolean hierarchy of NP sets and of Σ^p_2 sets, respectively [Chang and Kadin, 1996].

Acknowledgements

This work has been supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), by MUR under the PNRR project FAIR (PE0000013), and by the Royal Society (IES\R3\193236).

References

[Arasu *et al.*, 2009] Arvind Arasu, Christopher Ré, and Dan Suciu. Large-scale deduplication with constraints using dedupalog. In *Proceedings of the Twenty-Fifth International Conference on Data Engineering (ICDE 2009)*, pages 952–963, 2009.

[Arenas *et al.*, 1999] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1999)*, pages 68–79, 1999.

[Bertossi *et al.*, 2013] Leopoldo E. Bertossi, Solmaz Kolahi, and Laks V. S. Lakshmanan. Data cleaning and query answering with matching dependencies and matching functions. *Theory of Computing Systems*, 52(3):441–482, 2013.

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

[Bhattacharya and Getoor, 2007] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1):5, 2007.

[Bienvenu *et al.*, 2022] Meghyn Bienvenu, Gianluca Cima, and Víctor Gutiérrez-Basulto. LACE: A logical approach to collective entity resolution. In *Proceedings of the Forty-First International Conference on Management of Data (PODS 2022)*, pages 379–391, 2022.

[Bienvenu, 2020] Meghyn Bienvenu. A short survey on inconsistency handling in ontology-mediated query answering. *Künstliche Intelligenz*, 34(4):443–451, 2020.

[Burdick *et al.*, 2016] Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang-Chiew Tan. A declarative framework for linking entities. *ACM Transactions on Database Systems*, 41(3):17:1–17:38, 2016.

[Chang and Kadin, 1996] Richard Chang and Jim Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340–354, 1996.

[Chomicki and Marcinkowski, 2005] Jan Chomicki and Jerzy Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, 197(1-2):90–121, 2005.

[Christophides *et al.*, 2021] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys*, 53(6):127:1–127:42, 2021.

[Chu *et al.*, 2013] Xu Chu, Ihab F. Ilyas, and Paolo Papotti. Holistic data cleaning: Putting violations into context. In *Proceedings of the Twenty-Ninth IEEE International Conference on Data Engineering (ICDE 2013)*, pages 458–469, 2013.

[Deng *et al.*, 2022] Ting Deng, Wenfei Fan, Ping Lu, Xiaomeng Luo, Xiaoke Zhu, and Wanhe An. Deep and collective entity resolution in parallel. In *Proceedings of the Thirty-Eighth IEEE International Conference on Data Engineering (ICDE 2022)*, pages 2060–2072, 2022.

[Dixit and Kolaitis, 2022] Akhil A. Dixit and Phokion G. Kolaitis. Consistent answers of aggregation queries via SAT. In *Proceedings of the Thirty-Eighth IEEE International Conference on Data Engineering (ICDE 2022)*, pages 924–937, 2022.

[Fan and Geerts, 2012] Wenfei Fan and Floris Geerts. *Foundations of Data Quality Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012.

[Fan *et al.*, 2009] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. Reasoning about record matching rules. *Proceedings of the VLDB Endowment*, 2(1):407–418, 2009.

[Fan *et al.*, 2014] Wenfei Fan, Shuai Ma, Nan Tang, and Wenyuan Yu. Interaction between record matching and data repairing. *Journal of Data and Information Quality*, 4(4):16:1–16:38, 2014.

[Gebser *et al.*, 2012] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[Ilyas and Chu, 2019] Ihab F. Ilyas and Xu Chu. *Data Cleaning*. ACM, 2019.

[Lukasiewicz *et al.*, 2022] Thomas Lukasiewicz, Enrico Malizia, Maria Vanina Martinez, Cristian Molinaro, Andreas Pieris, and Gerardo I. Simari. Inconsistency-tolerant query answering for existential rules. *Artificial Intelligence*, 307:103685, 2022.

[Newcombe *et al.*, 1959] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959.

[Vardi, 1982] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth ACM Symposium on Theory of Computing (STOC 1982)*, pages 137–146, 1982.

Theorem 1. REC *is* P*-complete.*

Proof. **Upper Bound:** Given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, and a pair $W = (R, E)$, we now show how to check whether $W \in \mathsf{Sol}(D, \Sigma)$ in polynomial time in the size of D and W. First, following Definition 3, we know that $W \in \mathsf{Sol}(D, \Sigma)$ if and only if (*i*) $R \subseteq D$, (*ii*) $E \in \mathsf{ERSol}(D \setminus R, \Sigma)$, and (*iii*) for all $\varphi, \psi \in D$ with $\varphi \equiv_E \psi$, $\varphi \in R$ iff $\psi \in R$.

Condition (*i*) can be clearly checked in polynomial time in the size of D and W. Also condition (*iii*) can be checked in polynomial time in the size of D and W because checking whether $\varphi \equiv_E \psi$ can be trivially done in polynomial time. Finally, to verify condition (*ii*), we can first materialize the $\mathcal{S}$-database $D' = D \setminus R$ and then check whether $E \in \mathsf{ERSol}(D', \Sigma)$. Since due to [Bienvenu *et al.*, 2022, Theorem 1] this last step can be done in polynomial time in the size of D' and E, also condition (*ii*) can be checked in polynomial time in the size of D and W. Thus, we immediately get an overall procedure for checking whether $W \in \mathsf{Sol}(D, \Sigma)$ that runs in polynomial time in the size of D and W.

Lower Bound: The proof can be straightforwardly obtained from [Bienvenu *et al.*, 2022, Theorem 1]. Specifically, we know that there exists a fixed DQ specification Σ_{REC} over a fixed schema $\mathcal{S}_{\text{REC}}$ consisting only of a hard rule such that, given an $\mathcal{S}_{\text{REC}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, the problem of deciding whether $E \in \mathsf{ERSol}(D, \Sigma_{\text{REC}})$ is P-hard. The reduction from the above problem is as follows: given an $\mathcal{S}_{\text{REC}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, we construct in LOGSPACE a pair $W_E = (R, E)$, where $R = \emptyset$. Since $E \in \mathsf{ERSol}(D, \Sigma)$ if and only if $W = (\emptyset, E) \in \mathsf{Sol}(D, \Sigma)$ trivially holds for any database-specification pair (D, Σ) and equivalence relation E over $\mathsf{dom}(D)$, we derive that $E \in \mathsf{ERSol}(D, \Sigma_{\text{REC}})$ if and only if $W_E \in \mathsf{Sol}(D, \Sigma_{\text{REC}})$, thus obtaining the claimed lower bound. □

As announced in the main text of the paper, we now exhibit a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a DQ specification Σ over $\mathcal{S}$, and a $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$ but $W \notin \mathsf{Sol}_{\text{MER}}(D, \Sigma)$ and $W \notin \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$.

Example 10. *Consider the schema $\mathcal{S} = \{P/2, T/2\}$, the $\mathcal{S}$-database $D = \{P(a_1, a_2), P(a_3, a_4), T(a_1, a_2), T(a_3, a_4)\}$, and the DQ specification $\Sigma = \langle \Gamma, \Delta \rangle$ over $\mathcal{S}$, where $\Gamma = \{P(x, y) \dashrightarrow \mathsf{EQ}(x, y)\}$ and $\Delta = \{\neg(\exists y. T(y, y))\}$. Furthermore, consider $W = (R, E)$, where $R = \{T(a_1, a_2)\}$ and $E = \mathsf{EqRel}(\{\alpha\}, (D \setminus R))$ with $\alpha = (a_1, a_2)$.*

One can easily verify that $W \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$. However, we have that $W \notin \mathsf{Sol}_{\text{MER}}(D, \Sigma)$ and $W \notin \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$. The former because $W' = (R', E')$ with $R' = \{T(a_1, a_2), T(a_3, a_4)\}$ and $E' = \mathsf{EqRel}(\{(a_1, a_2), (a_3, a_4)\}, (D \setminus R'))$ is such that $W' \in \mathsf{Sol}(D, \Sigma)$ and $W \prec_{\text{MER}} W'$ (in fact, $\mathsf{Sol}_{\text{MER}}(D, \Sigma) = \{W'\}$). The latter because $W'' = (\emptyset, E'')$ with $E'' = \mathsf{trivE} = \{(c, c) \mid c \in \mathsf{dom}(D)\}$ is such that $W'' \in \mathsf{Sol}(D, \Sigma)$ and $W \prec_{\text{DEL}} W''$ (in fact, $\mathsf{Sol}_{\text{DEL}}(D, \Sigma) = \{W''\}$).

As announced in the main text of the paper, while $(\emptyset, E) \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$ iff $(\emptyset, E) \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$ iff E is a maximal ER solution in the sense of [Bienvenu *et al.*, 2022, Definition 3] holds for any database-specification pair (D, Σ), we now exhibit a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a DQ specification specification Σ over $\mathcal{S}$, and an equivalence relation E'' over $\mathsf{dom}(D)$ such that E'' is a maximal ER solution in the sense of [Bienvenu *et al.*, 2022, Definition 3] but $(\emptyset, E'') \notin \mathsf{Sol}_{\text{MER}}(D, \Sigma)$.

Example 11. *Recall Example 10. While E'' is a maximal ER solution in the sense of [Bienvenu* et al.*, 2022, Definition 3], we have that $W'' = (\emptyset, E'') \notin \mathsf{Sol}_{\text{MER}}(D, \Sigma)$. The latter holds because $W'' \prec_{\text{MER}} W'$ and $W' \in \mathsf{Sol}(D, \Sigma)$.*

Theorem 2. X-OPTREC *is coNP-complete for any* $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$.

Proof. **Upper Bound:** Given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, and a pair $W = (R, E)$, for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, we now show how to check whether $W \notin \mathsf{Sol}_X(D, \Sigma)$ in NP in the size of D and W. First, following Definition 4, we have that $W \notin \mathsf{Sol}_X(D, \Sigma)$ if and only if either (*i*) $W \notin \mathsf{Sol}(D, \Sigma)$ or (*ii*) there exists $W' = (R', E')$ such that $W' \in \mathsf{Sol}(D, \Sigma)$ and $W \prec_X W'$.

So, we first guess a pair $W' = (R', E')$, where $R' \subseteq D$ and E' is an equivalence relation over $\mathsf{dom}(D \setminus R')$. We then check conditions (*i*) and (*ii*). If either condition (*i*) or condition (*ii*) holds, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $W \notin \mathsf{Sol}_X(D, \Sigma)$ is trivial. As for its running time, we observe that W' is polynomially related to D. Furthermore, due to Theorem 1, checking whether $W \in \mathsf{Sol}(D, \Sigma)$ (resp. $W' \in \mathsf{Sol}(D, \Sigma)$) can be done in polynomial time in the size of D and W (resp. W'). Finally, also checking whether $W \prec_X W'$ can be trivially done in polynomial time in the size of W and W' for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$. So, overall, checking whether $W \notin \mathsf{Sol}_X(D, \Sigma)$ can be done in NP in the size of D and W.

Lower Bound for X = DEL and X = PAR: The proof can be straightforwardly obtained from [Bienvenu *et al.*, 2022, Theorem 3]. Specifically, from [Bienvenu *et al.*, 2022, Theorem 3] we know that there exists a fixed DQ specification $\Sigma^{\text{DEL,PAR}}_{\text{OPTREC}}$ over a fixed schema $\mathcal{S}^{\text{DEL,PAR}}_{\text{OPTREC}}$ such that, given an $\mathcal{S}^{\text{DEL,PAR}}_{\text{OPTREC}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, it is coNP-hard to decide whether E is a maximal ER solution for $(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$ in the sense of [Bienvenu *et al.*, 2022, Definition 3], i.e. $E \in \mathsf{ERSol}(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$ and there is no $E' \in \mathsf{ERSol}(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$ such that $E \subset E'$. The reduction from the above problem is as follows: given an $\mathcal{S}^{\text{DEL,PAR}}_{\text{OPTREC}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, we construct in LOGSPACE a pair $W_E = (R, E)$, where $R = \emptyset$. Since, as already observed in the paper, for any database-specification pair (D, Σ) and equivalence relation E over $\mathsf{dom}(D)$, we have that E is a maximal ER solution for (D, Σ) if and only if $W = (\emptyset, E) \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$ (resp. $W = (\emptyset, E) \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$), we derive that E is a maximal ER solution for $(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$ if and only if $W_E \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$ (resp. $W_E \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma^{\text{DEL,PAR}}_{\text{OPTREC}})$), thus obtaining the claimed lower bound.

Lower Bound for $X = $ MER: The proof is by a LOGSPACE reduction from the complement of the 3SAT problem. 3SAT is the prototypical NP-complete problem [?] of deciding, given a formula of the form $\phi = \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_m$ such that $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ is a clause of three literals (each literal being either a variable in $\mathbf{x} = (x_1, \ldots, x_n)$ or its negated) for each $i = 1, \ldots, m$, whether ϕ is `true`. For a clause $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$, we will say that $l_{i,1}$ (resp. $l_{i,2}$, $l_{i,3}$) is the first (resp. second, third) literal of c_i, and we will denote by $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) the variable $x \in \mathbf{x}$ of the literal $l_{i,1}$ (resp. $l_{i,2}$, $l_{i,3}$).

We first define the fixed schema $\mathcal{S}_{3\text{SAT}}$ and DQ specification $\Sigma_{3\text{SAT}}$ over $\mathcal{S}_{3\text{SAT}}$ as follows. We have the schema $\mathcal{S}_{3\text{SAT}} = \{R_{\mathit{fff}}/4,\ R_{\mathit{fft}}/4,\ R_{\mathit{ftf}}/4,\ R_{\mathit{ftt}}/4,\ R_{\mathit{tff}}/4,\ R_{\mathit{tft}}/4,\ R_{\mathit{ttf}}/4,\ R_{\mathit{ttt}}/4,\ P/4,\ T_X/1,\ F_X/1,\ O/2\}$. Informally, both T_X and F_X store (the constants representing) the variables $\mathbf{x}$, O simply stores the pair (o_1, o_2) of constants, and the R predicates are used to store the clauses of ϕ. For instance, a clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$ occurring in a 3SAT instance ϕ will be represented as $R_{\mathit{tft}}(c_5, x_2, x_4, x_1)$. Finally, consider again clause c_5. The predicate P will store three quadruples of the form $(c_5, x_2, a^{c_5}_{x_2}, b^{c_5}_{x_2})$, $(c_5, x_4, a^{c_5}_{x_4}, b^{c_5}_{x_4})$, and $(c_5, x_1, a^{c_5}_{x_1}, b^{c_5}_{x_1})$, where $a^{c_5}_{x_2}$ and $b^{c_5}_{x_2}$ (resp. $a^{c_5}_{x_4}$ and $b^{c_5}_{x_4}$, $a^{c_5}_{x_1}$ and $b^{c_5}_{x_1}$) are constants representing the fact that variables x_2, x_4, and x_1 occur in clause c_5. The DQ specification $\Sigma_{3\text{SAT}} = \langle \Gamma_{3\text{SAT}}, \Delta_{3\text{SAT}} \rangle$ over $\mathcal{S}_{3\text{SAT}}$ is such that $\Gamma_{3\text{SAT}}$ contains the following soft rules over $\mathcal{S}_{3\text{SAT}}$:

- $\sigma_O = O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_1 with constant o_2 in the presence of $O(o_1, o_2)$
- For every $I \in \{\mathit{fff},\ \mathit{fft},\ \mathit{ftf},\ \mathit{ftt},\ \mathit{tff},\ \mathit{tft},\ \mathit{ttf},\ \mathit{ttt}\}$, there are soft rules:
  - $\sigma^t_{I,1} = \exists c, v_1, v_2, v_3. P(c, v_1, x, y) \wedge T_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,1} = \exists c, v_1, v_2, v_3. P(c, v_1, x, y) \wedge F_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^t_{I,2} = \exists c, v_1, v_2, v_3. P(c, v_2, x, y) \wedge T_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,2} = \exists c, v_1, v_2, v_3. P(c, v_2, x, y) \wedge F_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^t_{I,3} = \exists c, v_1, v_2, v_3. P(c, v_3, x, y) \wedge T_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,3} = \exists c, v_1, v_2, v_3. P(c, v_3, x, y) \wedge F_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$

  Informally, consider again clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$. The presence of $P(c_5, x_2, a^{c_5}_{x_2}, b^{c_5}_{x_2})$ and $R_{\mathit{tft}}(c_5, x_2, x_4, x_1)$, together with the presence of at least one among $T_X(x_2)$ and $F_X(x_2)$, allows the merge of the constant $a^{c_5}_{x_2}$ with the constant $b^{c_5}_{x_2}$ thanks to the soft rules $\sigma^t_{\mathit{tft},1}$ and $\sigma^f_{\mathit{tft},1}$ (analogous considerations apply for the mergings of $a^{c_5}_{x_4}$ with $b^{c_5}_{x_4}$ and of $a^{c_5}_{x_1}$ with $b^{c_5}_{x_1}$).

and $\Delta_{3\text{SAT}}$ comprises the following denial constraints over $\mathcal{S}_{3\text{SAT}}$:

- $\delta_{\mathit{fff}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{fff}}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta_{\mathit{fft}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{fft}}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta_{\mathit{ftf}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{ftf}}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta_{\mathit{ftt}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{ftt}}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
- $\delta_{\mathit{tff}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{tff}}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta_{\mathit{tft}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{tft}}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta_{\mathit{ttf}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{ttf}}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta_{\mathit{ttt}} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O(z_1, z_2) \wedge R_{\mathit{ttt}}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$

Informally, consider a clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$. The denial δ_{tft} avoids the simultaneous presence of $O(o_1, o_2)$, of $R_{\mathit{tft}}(c_5, x_2, x_4, x_1)$, and of $F_X(x_2)$, $T_X(x_4)$, and $F_X(x_1)$.

Given an instance $\phi = \exists x_1, \ldots, x_n. c_1 \wedge \ldots \wedge c_m$ of the 3SAT problem, we construct an $\mathcal{S}_{3\text{SAT}}$-database D_ϕ and a pair $W_\phi = (R_\phi, E_\phi)$ as follows:

- D_ϕ contains the fact $O(o_1, o_2)$, the facts $T_X(x_i)$ and $F_X(x_i)$ for each $i = 1, \ldots, n$, and the facts $P(c_i, v_{i,1}, a^{c_i}_{v_{i,1}}, b^{c_i}_{v_{i,1}})$, $P(c_i, v_{i,2}, a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$, and $P(c_i, v_{i,3}, a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$ for each $i = 1, \ldots, m$, where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable of the first (resp. second, third) literal of clause c_i. Furthermore, for each $i = 1, \ldots, m$, if clause c_i is of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), then D_ϕ contains the fact $R_{\mathit{fff}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{\mathit{fft}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{ftf}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{ftt}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{tff}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{tft}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{ttf}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{\mathit{ttt}}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$), where, again, $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable of the first (resp. second, third) literal of clause c_i;
- R_ϕ contains only the fact $O(o_1, o_2)$;
- E_ϕ is the symmetric and transitive closure of the set containing the pair (c, c) for each $c \in \mathsf{dom}(D_\phi \setminus R_\phi)$ and the pairs $(a^{c_i}_{v_{i,1}}, b^{c_i}_{v_{i,1}})$, $(a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$, and $(a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$ for each $i = 1, \ldots, m$, where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ, R_ϕ, and E_ϕ can be constructed in LOGSPACE from an input 3SAT instance ϕ. To conclude the proof of the claimed lower bound, we now show that ϕ is $\texttt{true}$ if and only if $W_\phi \notin \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma_{3\text{SAT}})$.

Claim 1. *ϕ is $\texttt{true}$ if and only if $W_\phi \notin \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma_{3\text{SAT}})$.*

Proof. First, we have that $W_\phi \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$. To see this, observe that $E_\phi \in \mathsf{ERSol}(D_\phi \setminus R_\phi, \Sigma_{3\text{SAT}})$ due to the fact that (*i*) all the pairs in E_ϕ of the form (a_x^c, b_x^c) can be derived thanks to the soft rules in $\Gamma_{3\text{SAT}}$ and (*ii*) since R_ϕ contains the fact $O(o_1, o_2)$, no denial constraint can be violated by D'_{E_ϕ}, where $D' = D_\phi \setminus R_\phi$, i.e. $D'_{E_\phi} \models \Delta$.

Second, consider any pair $W = (R, E)$ such that $W \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$ and R contains either a fact $R_I(c_i, v_{i,1}, v_{i,2}, v_{i,3})$ representing a clause (for some $I \in \{\mathit{fff},\ \mathit{fft},\ \mathit{ftf},\ \mathit{ftt},\ \mathit{tff},\ \mathit{tft},\ \mathit{ttf},\ \mathit{ttt}\}$) or both facts $T_X(x_j)$ and $F_X(x_j)$ for some (constant representing the) variable $x_j \in \mathbf{x}$. Then, we can immediately get that $W_\phi \not\prec_{\text{MER}} W$. The reason is that there is at least one pair of the form $\alpha = (a_x^c, b_x^c)$ ($c = c_i$ if R contains a fact of the form $R_I(c_i, v_{i,1}, v_{i,2}, v_{i,3})$; and $x = x_j$ if R contains facts of the form $T_X(x_j)$ and $F_X(x_j)$) such that $\alpha \in E_\phi$ and $\alpha \notin E$ because α cannot be activated by the rules in Γ when removing from the $\mathcal{S}_{3\text{SAT}}$-database D_ϕ the above fact(s) in R.

It follows that the only way for a pair $W = (R, E) \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$ to be such that $W_\phi \prec_{\text{MER}} W$ is to satisfy the following three conditions: (*i*) R does not have any of the R_I-facts representing the clauses of ϕ, (*ii*) for each $x \in \mathbf{x}$, R does not have both $T_X(x)$ and $F_X(x)$, (*iii*) differently from R_ϕ, R does not have $O(o_1, o_2)$. While conditions (*i*) and (*ii*) ensure that all the pairs in E_ϕ also occur in E, due to the previous discussions, condition (*iii*) is the only way to ensure that $W_\phi \prec_{\text{MER}} W$ because allows for the merging of o_1 and o_2 due to the soft rule σ_O, thus obtaining $E_\phi \subset E$ because $(o_1, o_2) \in E$ and $(o_1, o_2) \notin E_\phi$. We now prove that a $W = (R, E) \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$ satisfying conditions (*i*), (*ii*), and (*iii*) exists if and only if ϕ is $\texttt{true}$, thus concluding the proof of the claim. Indeed, following Definition 4, we have that $W_\phi \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma_{3\text{SAT}})$ if and only if no W exists such that $W \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$ and $W_\phi \prec_{\text{MER}} W$.

Suppose that ϕ is not $\texttt{true}$, and consider any pair $W = (R, E)$ satisfying conditions (*i*), (*ii*), and (*iii*). We now prove that $D'_E \not\models \Delta_{3\text{SAT}}$, where $D' = D_\phi \setminus R$, and therefore $W \notin \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$. Due to condition (*ii*), we have that the $\mathcal{S}_{3\text{SAT}}$-database $D' = (D_\phi \setminus R)$ contains at least one among $T_X(x_i)$ and $F_X(x_i)$ for each $i = 1, \ldots, n$. This, however, can be seen as an assignment to the $\mathbf{x}$ variables, where, for each $i = 1, \ldots, n$, if $T_X(x_i) \in D'$, then we say that to x_i is assigned $\texttt{true}$; otherwise (i.e. $T_X(x_i) \notin D'$, and therefore $F_X(x_i) \in D'$), we say that to x_i is assigned $\texttt{false}$. Furthermore, due to conditions (*i*) and (*iii*), D' contains, respectively, all the R_I-facts representing the clauses of ϕ and $O(o_1, o_2)$. By construction of the denial constraints in $\Delta_{3\text{SAT}}$, since by assumption ϕ is not $\texttt{true}$, it follows that there is at least an $I \in \{\mathit{fff},\ \mathit{fft},\ \mathit{ftf},\ \mathit{ftt},\ \mathit{tff},\ \mathit{tft},\ \mathit{ttf},\ \mathit{ttt}\}$ such that $D' \not\models \delta_I$, which in turn implies that $D'_E \not\models \Delta_{3\text{SAT}}$, as required.

Suppose that ϕ is $\texttt{true}$, and let $f(\cdot)$ be the function assigning $\texttt{true}$ or $\texttt{false}$ to each variable $x \in \mathbf{x}$ that witnesses the truth of ϕ. Consider now R to be such that, for each $i = 1, \ldots, n$, $T_X(x_i) \in R$ if and only if $f(x_i) = \texttt{false}$ and $F_X(x_i) \in R$ if and only if $f(x_i) = \texttt{true}$. No other fact is included in R. So, the $\mathcal{S}_{3\text{SAT}}$-database $D' = (D_\phi \setminus R)$ contains, for each $i = 1, \ldots, n$, $T_X(x_i)$ if and only if $f(x_i) = \texttt{true}$ and $F_X(x_i)$ if and only if $f(x_i) = \texttt{false}$. Due to the fact that ϕ evaluates to $\texttt{true}$ under the assignment given by $f(\cdot)$, by construction of the denials in $\Delta_{3\text{SAT}}$, we have that $D' \models \Delta_{3\text{SAT}}$. Finally, let $E = E_\phi \cup \{(o_1, o_1), (o_1, o_2), (o_2, o_1), (o_2, o_2)\}$. It can be readily seen that $W = (R, E)$ is such that $W \in \mathsf{Sol}(D_\phi, \Sigma_{3\text{SAT}})$ and W satisfies conditions (*i*), (*ii*), and (*iii*), as required. □

We now introduce a property that is crucial to prove the results claimed in Theorems 3 and 5 regarding possible answers for the PAR preorder.

Lemma 1. *Let Σ be a DQ specification over a schema $\mathcal{S}$, D be an $\mathcal{S}$-database, q be an n-ary CQ over $\mathcal{S}$, and $\mathbf{c} = (c_1, \ldots, c_n)$ be an n-tuple of constants. We have that $\mathbf{c} \in$ PAR-$\mathsf{possAns}(q, D, \Sigma)$ if and only if $\mathbf{c} \in q(D, W)$ for some $W \in \mathsf{Sol}(D, \Sigma)$.*

Proof. First, suppose that $\mathbf{c} \notin q(D, W)$ for every $W \in \mathsf{Sol}(D, \Sigma)$. Then, following Definition 5, we have that $\mathbf{c} \notin$ PAR-$\mathsf{possAns}(q, D, \Sigma)$.

Now, suppose that $\mathbf{c} \in q(D, W)$ for some $W \in \mathsf{Sol}(D, \Sigma)$, where $W = (R, E)$. Since $W \in \mathsf{Sol}(D, \Sigma)$, following Definition 4, we have that either $W \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$ or there exists at least one pair $W' = (R', E')$ such that $W' \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$ and $W \prec_{\text{PAR}} W'$. In the former case, following Definition 5, we immediately get that $\mathbf{c} \in$ PAR-$\mathsf{possAns}(q, D, \Sigma)$ and we are done. Consider now the latter case. By definition, $W \prec_{\text{PAR}} W'$ implies that either $E \subset E'$ and $R' \subseteq R$ or (*ii*) $R' \subset R$ and $E \subseteq E'$ hold. Let D_W be the $\mathcal{S}$-database $D_W = (D \setminus R)_E$ and $D_{W'}$ be the $\mathcal{S}$-database $D_{W'} = (D \setminus R')_{E'}$. Since either (*i*) or (*ii*) holds, one can easily see that there is a *homomorphism* h from D_W to $D_{W'}$[2] such that $h(c) = \mathsf{rp}_{E'}(c)$ for each $c \in \mathsf{dom}(D_W)$. Thus, since $\mathbf{c} \in q(D, W)$, i.e. $\mathbf{c}_E \in q_E(D_W)$, and since CQs are *preserved under homomorphisms* [**?**], we soon derive that $\mathbf{c}_{E'} \in q_{E'}(D_{W'})$, and therefore $\mathbf{c} \in q(D, W')$. Thus, since $W' \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma)$ and $\mathbf{c} \in q(D, W')$, following Definition 5, we get that $\mathbf{c} \in$ PAR-$\mathsf{possAns}(q, D, \Sigma)$, as required. □

Another property that will be used in the subsequent results is the following.

[2] A homomorphism from an $\mathcal{S}$-database D to an $\mathcal{S}$-database D' is a function h from $\mathsf{dom}(D)$ to $\mathsf{dom}(D')$ such that $\alpha \in D$ implies $h(\alpha) \in D'$. As usual, for a fact α of the form $P(c_1, \ldots, c_n)$, $h(\alpha)$ denotes the fact $P(h(c_1), \ldots, h(c_n))$.

Lemma 2. *Let $\mathcal{S}$ be a schema, D be an $\mathcal{S}$-database, q be an n-ary CQ, $\mathbf{c} = (c_1, \ldots, c_n)$ be an n-tuple of constants, and $W = (R, E)$ be a pair such that $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. Then, checking whether $\mathbf{c} \in q(D, W)$ can be done in polynomial time in the size of D and W.*

Proof. By definition, it is enough to compute the $\mathcal{S}$-database $D' = (D \setminus R)_E$, the query q_E, the n-tuple $\mathbf{c}_E$ of constants, and finally check whether $q_E[\mathbf{c}_E]$ holds in D', i.e. $D' \models q_E[\mathbf{c}_E]$. If this is the case, then we return `true`; otherwise, we return `false`. Correctness is trivial. As for the running time of the above procedure, we observe that the first three steps are clearly feasible in polynomial time, whereas the last step is feasible even in AC^0 in the size of D' [**?**]. □

Theorem 3. *X-CERTANS is Π_2^p-complete for any $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, PAR-POSSANS is NP-complete, and X-POSSANS is Σ_2^p-complete for $X \in \{\text{MER}, \text{DEL}\}$.*

Proof. We start by proving that X-CERTANS is Π_2^p-complete for any $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, we then prove that X-POSSANS is Σ_2^p-complete for $X \in \{\text{MER}, \text{DEL}\}$, and finally we prove that PAR-POSSANS is NP-complete.

<u>X-CERTANS is Π_2^p-complete for any $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$.</u>

Upper Bound: Given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{c}$ of constants, for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, we now show how to check whether $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ in Σ_2^p in the size of D, thus obtaining the claimed upper bound. First, following Definition 5, we have that $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ if and only if there exists a W such that $W \in \mathsf{Sol}_X(D, \Sigma)$ and $\mathbf{c} \notin q(D, W)$.

So, we first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{c} \notin q(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ is trivial. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 2, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 2, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ can be done in Σ_2^p in the size of D for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$.

Lower Bound for X = MER: The proof is by a LOGSPACE reduction from the $\forall\exists$3CNF problem, a well-known Π_2^p-complete problem [**?**]. $\forall\exists$3CNF is the problem of deciding, given a quantified Boolean formula of the form $\phi = \forall \mathbf{y}.\exists \mathbf{x}.c_1 \wedge \ldots \wedge c_k$ such that $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ is a clause of three literals (each literal being either a variable in $\mathbf{x} \cup \mathbf{y}$ or its negation) for each $i = 1, \ldots, k$, whether ϕ is `true`, i.e. whether for each possible assignment to the universally quantified variables $\mathbf{y} = (y_1, \ldots, y_m)$ there exists an assignment to the existentially quantified variables $\mathbf{x} = (x_1, \ldots, x_n)$ that satisfy ϕ. For a clause $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$, we will say that $l_{i,1}$ (resp. $l_{i,2}$, $l_{i,3}$) is the first (resp. second, third) literal of c_i, and we will denote by $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) the variable in $\mathbf{x} \cup \mathbf{y}$ of the literal $l_{i,1}$ (resp. $l_{i,2}$, $l_{i,3}$). Without loss of generality, we assume that each clause c_i contains at least an existentially quantified variable $x \in \mathbf{x}$ (if not, then deciding whether ϕ is `true` is clearly not a Π_2^p-hard problem because ϕ would be trivially `false`). Moreover, again without loss of generality, given a clause $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ with only one occurrence of a universal variable $y \in \mathbf{y}$, we assume that y is the variable of the literal $l_{i,1}$. Analogously, given a clause $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ with two occurrences of (not necessarily distinct) universal variables in $\mathbf{y}$, we assume that they are variable(s) of the literal $l_{i,1}$ and of the literal $l_{i,2}$.

We define the fixed schema $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, DQ specification $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, and CQ $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$. We have $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \{R_{\mathit{fff}}/4,\ R_{\mathit{fft}}/4,\ R_{\mathit{ftf}}/4,\ R_{\mathit{ftt}}/4,\ R_{\mathit{tff}}/4,\ R_{\mathit{tft}}/4,\ R_{\mathit{ttf}}/4,\ R_{\mathit{ttt}}/4,\ V_Y/1,\ T/1,\ F/1,\ P/4,\ T_X/1,\ F_X/1,\ O/2\}$. Informally, V_Y stores (the constants representing) the universally quantified variables $\mathbf{y}$, T and F store the constants t (which stands for `true`) and f (which stands for `false`), respectively. Then, the other predicates play the same role as in the lower bound proof for X = MER of Theorem 2. Specifically, both T_X and F_X store (the constants representing) the existentially quantified variables $\mathbf{x}$, O simply stores the pair (o_1, o_2) of constants, and the R predicates are used to store the clauses of ϕ. For instance, a clause $c_5 = (y_2 \vee \overline{x_4} \vee x_1)$ occurring in a $\forall\exists$3CNF instance ϕ will be represented as $R_{\mathit{tft}}(c_5, y_2, x_4, x_1)$. Finally, consider again clause c_5. The predicate P will store two quadruples of the form $(c_5, x_4, a^{c_5}_{x_4}, b^{c_5}_{x_4})$ and $(c_5, x_1, a^{c_5}_{x_1}, b^{c_5}_{x_1})$, where $a^{c_5}_{x_4}$ and $b^{c_5}_{x_4}$ (resp. $a^{c_5}_{x_1}$ and $b^{c_5}_{x_1}$) are constants representing the fact that existentially quantified variables x_4 and x_1 occur in clause c_5.

The DQ specification $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}, \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \rangle$ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ is such that $\Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ contains the following soft rules over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$:

- $\sigma_Y^T = V_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}$ with the constant t
- $\sigma_Y^F = V_Y(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x, y)$, which which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}$ with the constant f
- $\sigma_O = O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_1 with constant o_2 in the presence of $O(o_1, o_2)$

- For every $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, there are soft rules:
  - $\sigma^t_{I,1} = \exists c, v_1, v_2, v_3 . P(c, v_1, x, y) \wedge T_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,1} = \exists c, v_1, v_2, v_3 . P(c, v_1, x, y) \wedge F_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^t_{I,2} = \exists c, v_1, v_2, v_3 . P(c, v_2, x, y) \wedge T_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,2} = \exists c, v_1, v_2, v_3 . P(c, v_2, x, y) \wedge F_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^t_{I,3} = \exists c, v_1, v_2, v_3 . P(c, v_3, x, y) \wedge T_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^f_{I,3} = \exists c, v_1, v_2, v_3 . P(c, v_3, x, y) \wedge F_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$

  Informally, consider again clause $c_5 = (y_2 \vee \overline{x_4} \vee x_1)$. The presence of $P(c_5, x_4, a^{c_5}_{x_4}, b^{c_5}_{x_4})$ and $R_{tft}(c_5, y_2, x_4, x_1)$, together with the presence of at least one among $T_X(x_4)$ or $F_X(x_4)$, allows the merge of the constants $a^{c_5}_{x_4}$ and $b^{c_5}_{x_4}$ thanks to the soft rules $\sigma^t_{tft,2}$ and $\sigma^f_{tft,2}$ (an analogous consideration applies for the merge of constants $a^{c_5}_{x_1}$ and $b^{c_5}_{x_1}$).

Note that σ_O and $\sigma^t_{I,1}, \sigma^f_{I,1}, \sigma^t_{I,2}, \sigma^f_{I,2}, \sigma^t_{I,3}, \sigma^f_{I,3}$ for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$ are the same soft rules used in the lower bound proof for $X =$ MER of Theorem 2.

Then, $\Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ comprises the following denial constraints over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$:

- $\delta_{TF} = \neg(\exists y . T(y) \wedge F(y))$, which prevents the merge between the constants t and f. This means that every (constant representing a) universally quantified variable in $\mathbf{y}$ can be merged with either the constant t or the constant f, but not both
- $\delta^0_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta^1_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta^2_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T_X(y_3))$
- $\delta^0_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta^1_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta^2_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F_X(y_3))$
- $\delta^0_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta^1_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta^2_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T_X(y_3))$
- $\delta^0_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
- $\delta^1_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
- $\delta^2_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F_X(y_3))$
- $\delta^0_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta^1_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
- $\delta^2_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T_X(y_3))$
- $\delta^0_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta^1_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
- $\delta^2_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F_X(y_3))$
- $\delta^0_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta^1_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
- $\delta^2_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T_X(y_3))$
- $\delta^0_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
- $\delta^1_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
- $\delta^2_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3 . O(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F_X(y_3))$

Informally, consider a clause $c_5 = (y_2 \vee \overline{x_4} \vee x_1)$. The denial δ^1_{tft} avoids that the (constant representing the) variable y_1 is merged with f and that $O(o_1, o_2)$, $R_{tft}(c_5, y_2 = f, x_4, x_1)$, $F(f = y_2)$, $T_X(x_4)$, and $F_X(x_1)$ occur in the database.

Finally, the fixed Boolean CQ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ is $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \exists y.O(y, y)$, asking whether constants o_1 and o_2 have been merged.

Given an instance $\phi = \forall \mathbf{y}.\exists \mathbf{x}.c_1 \wedge \ldots \wedge c_k$ of the ∀∃3CNF problem, where $\mathbf{y} = (y_1, \ldots, y_m)$ and $\mathbf{x} = (x_1, \ldots, x_n)$, we construct an $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$-database D_ϕ as follows:

- D_ϕ contains the facts $T(t)$, $F(f)$, and $V_Y(y_i)$ for each $i = 1, \ldots, m$;
- D_ϕ contains the fact $O(o_1, o_2)$, and the facts $T_X(x_i)$ and $F_X(x_i)$ for each $i = 1, \ldots, n$;
- for each clause c_i (i ranges from 1 to k) with no occurrences of universally quantified variables in $\mathbf{y}$, D_ϕ contains the facts $P(c_i, v_{i,1}, a^{c_i}_{v_{i,1}}, b^{c_i}_{v_{i,1}})$, $P(c_i, v_{i,2}, a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$, and $P(c_i, v_{i,3}, a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the existentially quantified variable of the first (resp. second, third) literal of clause c_i;
- for each clause c_i (i ranges from 1 to k) with exactly one occurrence of a universally quantified variable in $\mathbf{y}$, D_ϕ contains the facts $P(c_i, v_{i,2}, a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$ and $P(c_i, v_{i,3}, a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,2}$ and $v_{i,3}$ denote the existentially quantified variables of the second and the third, respectively, literal of clause c_i;
- for each clause c_i (i ranges from 1 to k) with exactly two occurrences of (not necessarily distinct) universally quantified variable(s) in $\mathbf{y}$, D_ϕ contains the fact $P(c_i, v_{i,3}, a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,3}$ denotes the existentially quantified variable of the third literal of clause c_i;
- Finally, for each $i = 1, \ldots, k$, if clause c_i is of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), then D_ϕ contains the fact $R_{fff}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{fft}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftf}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftt}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tff}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tft}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttf}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttt}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$), where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable in $\mathbf{x} \cup \mathbf{y}$ of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ can be constructed in LOGSPACE from an input ∀∃3CNF instance ϕ. To conclude the proof of the claimed lower bound, we now show that ϕ is `true` if and only if () is a MER-certain answer to $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$.

Claim 2. *ϕ is `true` if and only if* $() \in \text{MER-certAns}(q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}, D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$.

Proof. First, observe that every $W = (R, E)$ such that $W \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ must satisfy $(t, f) \notin E$ (i.e. t and f cannot be merged). Indeed, in the case that either $T(t)$ or $F(f)$ occur in R, we have that one of the two constants do not occur anymore in $D_\phi \setminus R$, and so t cannot be merged with f. In the case that both $T(t)$ and $F(f)$ occur in $D_\phi \setminus R$, the merge of t with f would cause the violation of the denial constraint δ_{TF}, and so, again, t cannot be merged with f.

Suppose that ϕ is not `true`, i.e. there exists an assignment $h_Y(\cdot)$ to the universally quantified variables $\mathbf{y}$ such that $\phi' = \exists \mathbf{x}.c'_1 \wedge \ldots \wedge c'_k$ is `false`, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with `true` if $h_Y(y) =$ `true` and with `false` otherwise ($h_Y(y) =$ `false`). Consider $W = (R, E)$ to be such that $R = \{O(o_1, o_2)\}$ and E is the symmetric and transitive closure of the following set S:

- S contains the pair (c, c) for each $c \in \mathsf{dom}(D_\phi \setminus R)$;
- for each $i = 1, \ldots, m$, if $h_Y(y_i) =$ `true`, then S contains the pair (y_i, t); otherwise (i.e. $h_Y(y_i) =$ `false`), S contains the pair (y_i, f). Observe that both (y_i, t) and (y_i, f) can be included thanks to the soft rules σ^T_Y and σ^F_Y, respectively;
- for each clause $i = 1, \ldots, k$, if clause c_i contains zero occurrences of universally quantified variables in $\mathbf{y}$, then S contains the pairs $(a^{c_i}_{v_{i,1}}, b^{c_i}_{v_{i,1}})$, $(a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$, and $(a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the existentially quantified variable of the first (resp. second, third) literal of clause c_i;
- for each clause $i = 1, \ldots, k$, if clause c_i contains exactly one occurrence of a universally quantified variable in $\mathbf{y}$, then S contains the pairs $(a^{c_i}_{v_{i,2}}, b^{c_i}_{v_{i,2}})$ and $(a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,2}$ and $v_{i,3}$ denote the existentially quantified variables of the second and the third, respectively, literal of clause c_i;
- for each clause $i = 1, \ldots, k$, if clause c_i contains two occurrences of (not necessarily distinct) universally quantified variable(s) in $\mathbf{y}$, then S contains the pair $(a^{c_i}_{v_{i,3}}, b^{c_i}_{v_{i,3}})$, where $v_{i,3}$ denotes the existentially quantified variables of the third literal of clause c_i;
- no other pair is in S.

Clearly, $() \notin q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}(D_\phi, W)$ holds. Furthermore, with analogous considerations as the ones used in the proof of Claim 1, it can be immediately verified that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$. Since $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ and $() \notin q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}(D_\phi, W)$, following Definition 5, we have that () is not a MER-certain answer to $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, as required.

Assume that ϕ is $\texttt{true}$. Based on this assumption, we now prove that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ must satisfy $\alpha \in E$, where $\alpha = (o_1, o_2)$, clearly implying that () is a MER-certain answer to $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$. Consider any $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ and suppose, for the sake of contradiction, that $\alpha \notin E$. Since $\alpha \notin E$, we derive that $O(o_1, o_2) \in R$, otherwise we would trivially have that $W' = (R, E')$, where $E' = E \cup \{(o_1, o_1), (o_1, o_2), (o_2, o_1), (o_2, o_2)\}$, is such that $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ and $W \prec_{\text{MER}} W'$, thus immediately deriving a contradiction to the fact that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$. Consider the assignment $h_Y(\cdot)$ such that, for each $i = 1, \ldots, m$, we have $h_Y(y_i) = \texttt{true}$ if $(y_i, t) \in E$, and $h_Y(y_i) = \texttt{false}$ otherwise (as observed at the beginning of the proof, we cannot have $(t, f) \in E$, which implies that, for no $i = 1, \ldots, m$, we have both $(y_i, t) \in E$ and $(y_i, f) \in E$).

Now, since by assumption ϕ is $\texttt{true}$, we have that there exists at least an assignment $h_X(\cdot)$ to the existentially quantified variables $\mathbf{x}$ that satisfies $\phi' = \exists \mathbf{x}. c'_1 \wedge \ldots \wedge c'_k$, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with $\texttt{true}$ if $h_Y(y) = \texttt{true}$ and with $\texttt{false}$ otherwise ($h_Y(y) = \texttt{false}$). Consider now $W' = (R', E')$ to be such that $E' = E \cup \{(o_1, o_1), (o_1, o_2), (o_2, o_1), (o_2, o_2)\}$ and, for each $i = 1, \ldots, n$, we have $T_X(x_i) \in R'$ if and only if $h_X(x_i) = \texttt{false}$ and $F_X(x_i) \in R'$ if and only if $h_X(x_i) = \texttt{true}$. No other fact is included in R'. With analogous considerations as the ones used in the proof of Claim 1, it can be immediately verified that $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ and $W \prec_{\text{MER}} W'$, which is a contradiction to the fact that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$, as required. □

Lower Bound for $X = $ DEL and $X = $ PAR: The proof is similar to [Bienvenu *et al.*, 2022, Theorem 6], and it is again by a LOGSPACE reduction from the $\forall\exists$3CNF problem. As in the previous lower bound proof, without loss of generality, we assume that each clause c_i in a $\forall\exists$3CNF instance ϕ contains at least an existentially quantified variable $x \in \mathbf{x}$ (if not, then deciding whether ϕ is $\texttt{true}$ is clearly not a Π^p_2-hard problem because ϕ would be trivially $\texttt{false}$).

Let us first define the fixed schema $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$, DQ specification $\Sigma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$, and CQ $q^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$. We have $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}} = \{R_{\mathit{fff}}/3,\ R_{\mathit{fft}}/3,\ R_{\mathit{ftf}}/3,\ R_{\mathit{ftt}}/3,\ R_{\mathit{tff}}/3,\ R_{\mathit{tft}}/3,\ R_{\mathit{ttf}}/3,\ R_{\mathit{ttt}}/3,\ V_Y/1,\ FV_X/1,\ LV_X/1,\ Prec_X/2,\ F/1,\ T/1,\ L/1,\ C/2,\ C'/2\}$. Informally, the R's predicates and the predicates V_Y, F, and T play a similar role as in the previous lower bound proof, while FV_X and LV_X store (the constants representing) the first and the last existential variable, respectively, and $Prec_X$ stores pairs of the form (x_i, x_{i+1}) of existential variables indicating that variable x_{i+1} comes soon after variable x_i. Finally, L stores the constants t (which stands for $\texttt{true}$) and f (which stands for $\texttt{false}$), and C and C' only store the pair of constants (c_1, c_2) and (c, c'), respectively.

The DQ specification $\Sigma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}, \Delta^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}} \rangle$ over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$ is such that $\Gamma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$ contains the following soft rules over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$:

- $\sigma^T_Y = V_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of (the constants representing) the universally quantified variables $\mathbf{y}$ with the constant t
- $\sigma^F_Y = V_Y(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of (the constants representing) the universally quantified variables $\mathbf{y}$ with the constant f
- $\sigma_{C,C'} = C'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge between constants c and c'
- $\sigma_{FV} = \exists z. FV_X(x) \wedge L(y) \wedge C'(z, z) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge of (the constant representing) the first existentially quantified variable x_1 with both constants t and f but only if constants c and c' have been previously merged
- $\sigma_{Prec} = \exists z_p. L(z_p) \wedge Prec(z_p, x) \wedge L(y) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge of (the constant representing) the existential variable x_i ($2 \leq i \leq n$) with both constants t and f but only if the existential variable x_{i-1} has been previously merged with either constant t or constant f
- $\sigma_{C_1,C_2} = \exists z. C(x, y) \wedge LV_X(z) \wedge L(z) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge between constants c_1 and c_2 but only if the last (constant representing the) existentially quantified variable x_n has been previously merged with either constant t or constant f

Then, $\Delta^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$ comprises the following ten denial constraints over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$:

- $\delta_{TF} = \neg(\exists y. T(y) \wedge F(y))$, which prevents the merge between the constants t and f. This means that every (constant representing a) variable in $\mathbf{x} \cup \mathbf{y}$ can be merged with either the constant t or the constant f, but not both
- $\delta_C = \neg(\exists y, y_1, y_2. C'(y, y) \wedge C(y_1, y_2) \wedge y_1 \neq y_2)$, which is violated if the constants c and c' are merged while the constants c_1 and c_2 are not merged
- $\delta_{\mathit{fff}} = \neg(\exists y_1, y_2, y_3. R_{\mathit{fff}}(y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta_{\mathit{fft}} = \neg(\exists y_1, y_2, y_3. R_{\mathit{fft}}(y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta_{\mathit{ftf}} = \neg(\exists y_1, y_2, y_3. R_{\mathit{ftf}}(y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta_{\mathit{ftt}} = \neg(\exists y_1, y_2, y_3. R_{\mathit{ftt}}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta_{\mathit{tff}} = \neg(\exists y_1, y_2, y_3. R_{\mathit{tff}}(y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T(y_3))$

- $\delta_{tft} = \neg(\exists y_1, y_2, y_3 . R_{tft}(y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta_{ttf} = \neg(\exists y_1, y_2, y_3 . R_{ttf}(y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta_{ttt} = \neg(\exists y_1, y_2, y_3 . R_{ttt}(y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F(y_3))$

Informally, consider a clause $c_5 = (y_2 \vee \overline{x_4} \vee x_1)$. In the presence of $R_{tft}(y_2, x_4, x_1)$, the denial δ_{tft} avoids that, at the same time, variables y_2 and x_1 are merged with the constant f and variable x_4 is merged with the constant t. In other words, once given an assignment to all the variables in $\mathbf{x} \cup \mathbf{y}$, no clause can be unsatisfied.

Finally, the fixed Boolean CQ over $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$ is $q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}} = \exists y . C'(y, y)$, asking whether constants c and c' have been merged.

Given an instance $\phi = \forall \mathbf{y} . \exists \mathbf{x} . c_1 \wedge \ldots \wedge c_k$ of the $\forall\exists$3CNF problem, where $\mathbf{y} = (y_1, \ldots, y_m)$ and $\mathbf{x} = (x_1, \ldots, x_n)$, we construct an $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$-database D_ϕ as follows:

- D_ϕ contains the fact $V_Y(y_i)$ for each $i = 1, \ldots, m$, the fact $FV_X(x_1)$, the fact $Prec_X(x_i, x_{i+1})$ for each $i = 1, \ldots, n-1$, the fact $LV_X(x_n)$, the facts $F(f)$, $T(t)$, $L(f)$ and $L(t)$, and the two facts $C'(c, c')$ and $C(c_1, c_2)$;
- for each clause c_i of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), D_ϕ contains the fact $R_{fff}(v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{fft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftt}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tff}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttt}(v_{i,1}, v_{i,2}, v_{i,3})$), where, $v_{i,1}$ (resp. $v_{i,2}, v_{i,3}$) denotes the variable in $\mathbf{x} \cup \mathbf{y}$ of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ can be constructed in LOGSPACE from an input $\forall\exists$3CNF instance ϕ. To conclude the proof of the claimed lower bound, we now show that, for both $X = \text{DEL}$ and $X = \text{PAR}$, ϕ is `true` if and only if $()$ is an X-certain answer to $q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$.

Claim 3. *For both $X = \text{DEL}$ and $X = \text{PAR}$, ϕ is `true` if and only if $() \in X\text{-certAns}(q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}, D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$.*

Proof. Suppose that ϕ is `false`, i.e. there exists an assignment $h_Y(\cdot)$ to the universally quantified variables $\mathbf{y}$ such that $\phi' = \exists \mathbf{x} . c'_1 \wedge \ldots \wedge c'_k$ is `false`, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with `true` if $h_Y(y) = \texttt{true}$ and with `false` otherwise ($h_Y(y) = \texttt{false}$). Consider $W = (R, E)$ to be such that $R = \emptyset$ and E is the symmetric and transitive closure of the following set S:

- S contains the pair (c, c) for each $c \in \mathsf{dom}(D_\phi)$;
- for each $i = 1, \ldots, m$, if $h_Y(y_i) = \texttt{true}$, then S contains the pair (y_i, t); otherwise (i.e. $h_Y(y_i) = \texttt{false}$), S contains the pair (y_i, f). Observe that both (y_i, t) and (y_i, f) can be included thanks to the soft rules σ_Y^T and σ_Y^F, respectively;
- no other pair is in S.

Clearly, $() \notin q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}(D_\phi, W)$ holds because $(c, c') \notin E$. We now show that $W \in \mathsf{Sol}_X(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$, thus implying $() \notin X\text{-certAns}(q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}, D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$ as per Definition 5. Since $R = \emptyset$, by definition, for both $X = \text{DEL}$ and $X = \text{PAR}$, the only way for a $W' = (R', E')$ to be such that $W \prec_X W'$ is that $R' = \emptyset$ and $E \subset E'$. So, consider any $W' = (R', E')$ with $R' = \emptyset$ and $E \subset E'$. Observe that $\alpha = (c, c')$ (resp. (c', c)) is the only pair active in (D_ϕ, E) w.r.t. $\Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$, and therefore $\alpha \in E'$ due to the fact that $E \subset E'$ (otherwise, we immediately get that $W' \notin \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$). Merging c with c' causes the violation of δ_C because $c_1 \neq c_2$. By construction of the soft rules, with a trivial inductive argument, it is easy to see that this violation can be solved only by first merging each (constant representing a) existentially quantified variables $\mathbf{x}$ with either t or f, and then merging c_1 with c_2 thanks to σ_{C_1, C_2}. Thus, all such merges must occur in E' (otherwise, we immediately get that $W' \notin \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$). Since ϕ is `false`, however, whatever is the combination of merges applied to the existentially quantified variable, by construction of the denial constraints in $\Delta^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$, it is easy to see that there must be at least one $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$ such that $D_{\phi_{E'}} \not\models \delta_I$, and therefore $W' \notin \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$. It follows that there can be no W' satisfying $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$ and $W \prec_X W'$, and therefore $W \in \mathsf{Sol}_X(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$, as required.

Assume now that ϕ is `true`. Based on this assumption, we now prove that every $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$ must satisfy $\alpha \in E$, where $\alpha = (c, c')$, clearly implying that $()$ is an X-certain answer to $q^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$. Consider any $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}})$ and suppose, for the sake of contradiction, that $\alpha \notin E$. Since $\alpha \notin E$, we derive that no existentially quantified variable $x \in \mathbf{x}$ has been merged with either t or f because σ_{FV} cannot activate the merge between x_1 and t or f and, consequently, σ_{Prec} cannot activate the merge between the other existentially quantified variables with either t or f. Consider the assignment $h_Y(\cdot)$ such that, for each $i = 1, \ldots, m$, we have $h_Y(y_i) = \texttt{true}$ if $(y_i, t) \in E$, and $h_Y(y_i) = \texttt{false}$ otherwise (by construction of the soft rules σ_Y^T and σ_Y^F, due to the denial constraint δ_{TF}, for no $i = 1, \ldots, m$ we can have both $(y_i, t) \in E$ and $(y_i, f) \in E$).

Since by assumption ϕ is `true`, we have that there exists at least an assignment $h_X(\cdot)$ to the existentially quantified variables $\mathbf{x}$ that satisfies $\phi' = \exists \mathbf{x} . c'_1 \wedge \ldots \wedge c'_k$, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with `true` if $h_Y(y) = \texttt{true}$ and with `false` otherwise ($h_Y(y) = \texttt{false}$). So, consider now $W' = (R', E')$ to be such that $R' = \emptyset$ and

$E' = E \cup S$, where S is the symmetric and transitive closure of the set containing the pairs α, (x_i, t) if $h_X(x_i) = \mathtt{true}$ and (x_i, f) if $h_X(x_i) = \mathtt{false}$, for each $i = 1, \ldots, n$, and (c_1, c_2). It is clear that $W \prec_X W'$ because $E \subset E'$ (indeed $\alpha \notin E$ and $\alpha \in E'$). Furthermore, due to the fact that ϕ evaluates to $\mathtt{true}$ under the assignment given by $h_Y(\cdot) \cup h_X(\cdot)$, by construction of the denials in $\Delta^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$, we have that $D_{\phi_{E'}} \models \Delta^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}}$, and therefore $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}})$. So, as required, we have a contradiction to the fact that $W \in \mathsf{Sol}_X(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}})$ because $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{CERT,D/C}}_{\forall\exists\text{3CNF}})$ and $W \prec_X W'$. □

X-POSSANS is Σ^p_2-complete for $X \in \{\text{MER}, \text{DEL}\}$.

Upper Bound: Given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{c}$ of constants, for both $X = \text{MER}$ and $X = \text{DEL}$, we now show how to check whether $\mathbf{c} \in X\text{-}\mathsf{possAns}(q, D, \Sigma)$ in Σ^p_2 in the size of D. First, following Definition 5, we have that $\mathbf{c} \in X\text{-}\mathsf{possAns}(q, D, \Sigma)$ if and only if there exists a W such that $W \in \mathsf{Sol}_X(D, \Sigma)$ and $\mathbf{c} \in q(D, W)$.

So, we first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{c} \in q(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return $\mathtt{true}$; otherwise, we return $\mathtt{false}$. Correctness of the above procedure for checking $\mathbf{c} \in X\text{-}\mathsf{possAns}(q, D, \Sigma)$ is trivial. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 2, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 2, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{c} \in X\text{-}\mathsf{possAns}(q, D, \Sigma)$ can be done in Σ^p_2 in the size of D for both $X = \text{MER}$ and $X = \text{DEL}$.

Lower Bound for $X = \text{MER}$: The proof is by a LOGSPACE reduction from the complement of the ∀∃3CNF problem, and it can be obtained with a slight modification of the above lower bound proof for MER-CERTANS. More specifically, recall the fixed schema $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, DQ specification $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}, \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \rangle$ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, and query $q^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ used in the lower bound proof for MER-CERTANS. Consider now the fixed schema $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \mathcal{S}^{\text{POSS,M}}_{\forall\exists\text{3CNF}} \cup \{H/1\}$, DQ specification $\Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{POSS,M}}_{\forall\exists\text{3CNF}}, \Delta^{\text{POSS,M}}_{\forall\exists\text{3CNF}} \rangle$ over $\mathcal{S}^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$, and query $q^{\text{POSS,M}}_{\forall\exists\text{3CNF}} = \exists y. H(y)$ over $\mathcal{S}^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$, where:

- $\Gamma^{\text{POSS,M}}_{\forall\exists\text{3CNF}} = \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$
- $\Delta^{\text{POSS,M}}_{\forall\exists\text{3CNF}} = \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \cup \{\delta_{O,H}\}$ with $\delta_{O,H} = \neg(\exists y_1, y_2. O(y_1, y_2) \land H(y_2))$

Given an instance $\phi = \forall \mathbf{y}. \exists \mathbf{x}. c_1 \land \ldots \land c_k$ of the ∀∃3CNF problem, recall the $\mathcal{S}$-database constructed in the reduction used in the lower bound proof for MER-CERTANS, and let $D'_\phi = D_\phi \cup \{H(o_2)\}$. With the correctness of Claim 2 at hand, we can now easily conclude the proof of the claimed lower bound by showing that ϕ is $\mathtt{true}$ if and only if () is not a MER-possible answer to $q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$ on D'_ϕ w.r.t. $\Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$.

Claim 4. *ϕ is $\mathtt{true}$ if and only if* $() \notin \text{MER-}\mathsf{possAns}(q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}, D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$.

Proof. Suppose that ϕ is $\mathtt{false}$, and consider the same $W = (R, E)$ used in the "if part" of Claim 2. Given that $W \in \mathsf{Sol}_{\text{MER}}(D_\phi, \Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}})$ and $O(o_1, o_2) \in R$, we can immediately derive that $W \in \mathsf{Sol}_{\text{MER}}(D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$ as well. Furthermore, $() \in q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}(D'_\phi, W)$ clearly holds because $H(o_2) \in (D'_\phi \setminus R)$. Since $W \in \mathsf{Sol}_{\text{MER}}(D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$ and $() \in q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}(D'_\phi, W)$, following Definition 5, we get that () is a MER-possible answer to $q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$ on D'_ϕ w.r.t. $\Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$, as required.

Assume now that ϕ is $\mathtt{true}$. Using analogous considerations done in the "only-if part" of Claim 2, one can immediately derive that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$ must satisfy $\alpha \in E$, where $\alpha = (o_1, o_2)$. By construction of the denial constraint $\delta_{O,H}$, it follows that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$ must satisfy $H(o_2) \in R$, clearly implying that $() \notin q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}(D'_\phi, W)$. Thus, since $() \notin q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}(D'_\phi, W)$ holds for every $W \in \mathsf{Sol}_{\text{MER}}(D'_\phi, \Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}})$, following Definition 5, we get that () is not a MER-possible answer to $q^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$ on D'_ϕ w.r.t. $\Sigma^{\text{POSS,M}}_{\forall\exists\text{3CNF}}$, as required. □

Lower Bound for $X = \text{DEL}$: The proof is by a LOGSPACE reduction from the complement of the ∀∃3CNF problem. Differently from the lower bound proofs for MER-CERTANS and MER-POSSANS provided above, here the universally quantified variables will play the role of the variables occurring in the removed facts, while the existentially quantified variables will play the role of the variables merged with either t or f. Without loss of generality, given a ∀∃3CNF instance $\phi = \forall \mathbf{y}. \forall \mathbf{x}. c_1 \land \ldots c_k$, we assume the following: (*i*) each clause c_i contains at least an existentially quantified variable $x \in \mathbf{x}$ (if not, then deciding whether ϕ is $\mathtt{true}$ is clearly not a Π^p_2-hard problem because ϕ would be trivially $\mathtt{false}$); (*ii*) given a clause $c_i = (l_{i,1} \lor l_{i,2} \lor l_{i,3})$ with only one occurrence of a universal variable $y \in \mathbf{y}$, we assume that y is the variable of the literal $l_{i,1}$. Analogously, given a clause $c_i = (l_{i,1} \lor l_{i,2} \lor l_{i,3})$ with two occurrences of (not necessarily distinct) universal variables in $\mathbf{y}$, we assume that they are variable(s) of the literal $l_{i,1}$ and of the literal $l_{i,2}$.

We define the fixed schema $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, DQ specification $\Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, and CQ $q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$. We have $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}} = \{R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ T_Y/1,\ F_Y/1,\ FV_X/1,\ Prec_X/2,\ LV_X/1,$

$T/1,\ F/1,\ L/1,\ C/2,\ C'/2,\ H/1\}$. Informally, as usual, the R predicates are used to store the clauses of ϕ, while F and T store the constants f and t, respectively, and L stores both the constants f and t. Then, both T_Y and F_Y store (the constants representing) the universally quantified variables $\mathbf{y}$, while FV_X and LV_X store, respectively, (the constants representing) the first existentially quantified variable $x_1 \in \mathbf{x}$ and the last existentially quantified variable $x_n \in \mathbf{x}$, and $Prec_X$ stores pairs of the form (x_i, x_{i+1}) of (constants representing) existential variables indicating that x_{i+1} comes soon after x_i. Finally, the predicate C only stores the pair (c_1, c_2) of constants, the predicate C' only stores the pair (c, c') of constants, and the predicate H only stores the constant c'.

The DQ specification $\Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}, \Delta^{\text{POSS,D}}_{\forall\exists\text{3CNF}} \rangle$ over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ is such that $\Gamma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ contains the following soft rules over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$:

- $\sigma_{FV} = FV_X(x) \wedge L(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of the (constant representing the) first existentially quantified variable x_1 with both t and f
- $\sigma_{Prec} = \exists z_p. L(z_p) \wedge Prec_X(z_p, x) \wedge L(y) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge of (the constant representing) the existential variable x_i ($2 \le i \le n$) with both constants t and f but only if the existential variable x_{i-1} has been previously merged with either constant t or constant f
- $\sigma_C = \exists z. C(x, y) \wedge LV_X(z) \wedge L(z) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge between constants c_1 and c_2 but only if the last (constant representing the) existentially quantified variable x_n has been previously merged with either constant t or constant f
- $\sigma_{C'} = C'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge between constants c and c'
- $\sigma'_C = \exists z. C'(z, z) \wedge C(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which allows the merge between constants c_1 and c_2 but only if constants c and c' have been previously merged

Then, $\Delta^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ comprises the following denial constraints over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$:

- $\delta_C = \neg(\exists y_1, y_2. C(y_1, y_2) \wedge y_1 \neq y_2)$, which is originally not satisfied by the database because C contains the pair (c_1, c_2). This enforces either the merge of constants c_1 and c_2 or the deletion of $C(c_1, c_2)$
- $\delta_{C'} = \neg(\exists y. C'(y, y) \wedge H(y))$, which enforces the deletion of $H(c')$ in the case that constants c and c' have been merged
- $\delta_Y = \neg(\exists y. T_Y(y) \wedge F_Y(y))$, which means that, for each (constant representing an) universally quantified variable $\mathbf{y_i} \in \mathbf{y}$, either $T_Y(y_i)$ or $F_Y(y_i)$ must be deleted from the database
- $\delta^0_{fff} = \neg(\exists y_1, y_2, y_3. R_{fff}(y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta^1_{fff} = \neg(\exists y_1, y_2, y_3. R_{fff}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta^2_{fff} = \neg(\exists y_1, y_2, y_3. R_{fff}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge T_Y(y_2) \wedge T(y_3))$
- $\delta^0_{fft} = \neg(\exists y_1, y_2, y_3. R_{fft}(y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta^1_{fft} = \neg(\exists y_1, y_2, y_3. R_{fft}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta^2_{fft} = \neg(\exists y_1, y_2, y_3. R_{fft}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge T_Y(y_2) \wedge F(y_3))$
- $\delta^0_{ftf} = \neg(\exists y_1, y_2, y_3. R_{ftf}(y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta^1_{ftf} = \neg(\exists y_1, y_2, y_3. R_{ftf}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta^2_{ftf} = \neg(\exists y_1, y_2, y_3. R_{ftf}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge F_Y(y_2) \wedge T(y_3))$
- $\delta^0_{ftt} = \neg(\exists y_1, y_2, y_3. R_{ftt}(y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta^1_{ftt} = \neg(\exists y_1, y_2, y_3 R_{ftt}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta^2_{ftt} = \neg(\exists y_1, y_2, y_3. R_{ftt}(y_1, y_2, y_3) \wedge T_Y(y_1) \wedge F_Y(y_2) \wedge F(y_3))$
- $\delta^0_{tff} = \neg(\exists y_1, y_2, y_3. R_{tff}(y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta^1_{tff} = \neg(\exists y_1, y_2, y_3. R_{tff}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta^2_{tff} = \neg(\exists y_1, y_2, y_3. R_{tff}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge T_Y(y_2) \wedge T(y_3))$
- $\delta^0_{tft} = \neg(\exists y_1, y_2, y_3. R_{tft}(y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta^1_{tft} = \neg(\exists y_1, y_2, y_3. R_{tft}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta^2_{tft} = \neg(\exists y_1, y_2, y_3. R_{tft}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge T_Y(y_2) \wedge F(y_3))$
- $\delta^0_{ttf} = \neg(\exists y_1, y_2, y_3. R_{ttf}(y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T(y_3))$

- $\delta^1_{ttf} = \neg(\exists y_1, y_2, y_3. R_{ttf}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta^2_{ttf} = \neg(\exists y_1, y_2, y_3. R_{ttf}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge F_Y(y_2) \wedge T(y_3))$
- $\delta^0_{ttt} = \neg(\exists y_1, y_2, y_3. R_{ttt}(y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta^1_{ttt} = \neg(\exists y_1, y_2, y_3. R_{ttt}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta^2_{ttt} = \neg(\exists y_1, y_2, y_3. R_{ttt}(y_1, y_2, y_3) \wedge F_Y(y_1) \wedge F_Y(y_2) \wedge F(y_3))$

Informally, consider a clause $c = (y_2 \vee \overline{x_4} \vee x_1)$. The denial δ^1_{tft} avoids that the (constants representing the) variables x_4 and x_1 are merged with t and f, respectively, and that $R_{tft}(y_2, x_4 = t, x_1 = f)$, $F_Y(y_2)$, $T(t = x_4)$, and $F(f = x_1)$ occur in the database.

Finally, the fixed Boolean CQ over $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ is $q^{\text{POSS,D}}_{\forall\exists\text{3CNF}} = \exists y. C'(y, y)$, asking whether constants c and c' have been merged.

Given an instance $\phi = \forall \mathbf{y}. \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_k$ of the $\forall\exists$3CNF problem, where $\mathbf{y} = (y_1, \ldots, y_m)$ and $\mathbf{x} = (x_1, \ldots, x_n)$, we construct an $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$-database D_ϕ as follows:

- D_ϕ contains the facts $T(t)$, $F(f)$, $L(t)$, $L(f)$, $C(c_1, c_2)$, $C'(c, c')$, and $H(c')$;
- D_ϕ contains the facts $T_Y(y_i)$ and $F_Y(y_i)$ for each $i = 1, \ldots, m$;
- D_ϕ contains the facts $FV_X(x_1)$, $LV_X(x_n)$, and the fact $Prec_X(x_i, x_{i+1})$ for each $i = 1, \ldots, n-1$;
- for each clause of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), the $\mathcal{S}^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$-database D_ϕ contains the fact $R_{fff}(v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{fft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftt}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tff}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttt}(v_{i,1}, v_{i,2}, v_{i,3})$), where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable in $\mathbf{x} \cup \mathbf{y}$ of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ can be constructed in LOGSPACE from an input $\forall\exists$3CNF instance ϕ. To conclude the proof of the claimed lower bound, we now show that ϕ is `true` if and only if $()$ is not a DEL-possible answer to $q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$.

Claim 5. *ϕ is `true` if and only if* $() \notin$ DEL-possAns$(q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}, D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$.

Proof. Consider any $W = (R, E)$ such that $W \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$. Due to δ_Y, the following holds: for each $i = 1, \ldots, m$, either $T_Y(y_i) \in R$ or $F_Y(y_i) \in R$. Furthermore, due to δ_C, we must have either $(c_1, c_2) \in E$ or $C(c_1, c_2) \in R$. Consider the case $(c_1, c_2) \in E$. By construction of the soft rules, the only way to get $(c_1, c_2) \in E$ is either via the soft rule σ_C or via the soft rule σ'_C. If the merge between c_1 and c_2 has been activated by σ'_C, then it follows that c and c' have been previously merged (which can be done due to the soft rule $\sigma_{C'}$). In this case, however, due to $\delta_{C'}$, we must have that $H(c') \in R$ (we cannot have $C'(c, c') \in R$, otherwise c and c' cannot be merged). On the contrary, if $(c, c') \notin E$, and therefore the merge between c_1 and c_2 has been activated by σ_C, then it follows that the (constant representing the) existentially quantified variable x_n has been merged with either t or f. Using a trivial inductive argument, it can be proven that such merging of x_n with either t or f can be done only if each of its preceding variables (if any, i.e. if $n > 1$) $x_1, \ldots, x_{n-1}$ has been previously merged with either t or f, where as base case the merge of x_1 with either t or f can be done due to the soft rule σ_{FV}. With these observations at hand, we now prove the if and only if statement in the claim.

Assume that ϕ is `true`. We now prove that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ must satisfy $\alpha \notin E$, where $\alpha = (c, c')$, clearly implying that $() \notin q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}(D_\phi, W)$. Specifically, consider any $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and suppose, for the sake of contradiction, that $\alpha \in E$. As already discussed above, due to $\delta_{C'}$ and the fact that $\alpha \in E$, we derive that $H(c') \in R$. Let $h_Y(\cdot)$ be the assignment such that, for each $i = 1, \ldots, m$, we have $h_Y(y_i) = \texttt{true}$ if $F_Y(y_i) \in R$; and $h_Y(y_i) = \texttt{false}$ otherwise (which implies $T_Y(y_i) \in R$ because, for each $i = 1, \ldots, m$, either $T_Y(y_i)$ or $F_Y(y_i)$ must belong to R).

Since by assumption ϕ is `true`, we have that there exists at least an assignment $h_X(\cdot)$ to the existentially quantified variables $\mathbf{x}$ that satisfies $\phi' = \exists \mathbf{x}. c'_1 \wedge \ldots \wedge c'_k$, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with `true` if $h_Y(y) = \texttt{true}$ and with `false` otherwise ($h_Y(y) = \texttt{false}$). Consider now $W' = (R', E')$ be such that (*i*) R' contains $F_Y(y_i)$ if $h_Y(y_i) = \texttt{true}$ (i.e. if $F_Y(y_i) \in R$) and $T_Y(y_i)$ if $h_Y(y_i) = \texttt{false}$ (which implies $T_Y(y_i) \in R$), for each $i = 1, \ldots, m$. No other fact is included in R'; and (*ii*) E' is the symmetric and transitive closure of the set containing (c, c) for each $c \in \mathsf{dom}(D_\phi \setminus R')$, the pair (c_1, c_2), and, for each $i = 1, \ldots, n$, the pair (x_i, t) if $h_X(x_i) = \texttt{true}$ and the pair (x_i, f) otherwise (i.e. if $h_X(x_i) = \texttt{false}$). Since by assumption $h_X(\cdot)$ makes ϕ' true, $(c_1, c_2) \in E'$, and $\alpha \notin E'$, we can easily derive that no denial constraint in $\Delta^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ is violated, i.e. $D'_{E'} \models \Delta^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, where $D' = (D_\phi \setminus R')$, and therefore $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$. Furthermore, since $R' \subset R$ because $H(c') \in R$ while $H(c') \notin R'$, we derive that $W \prec_{\text{DEL}} W'$. So, since $W' \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and $W \prec_{\text{DEL}} W'$, we have a contradiction to the fact that $W \in \mathsf{Sol}_{\text{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$. Thus, every $W \in \mathsf{Sol}_{\text{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ satisfies $\alpha \notin E$, and therefore also $() \notin q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}(D_\phi, W)$. Since $() \notin q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}(D_\phi, W)$

holds for every $W \in \mathsf{Sol}_{\mathrm{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$, following Definition 5, we get that $()$ is not a DEL-possible answer to $q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, as required.

Suppose that ϕ is not $\texttt{true}$, i.e. there exists an assignment $h_Y(\cdot)$ to the universally quantified variables $\mathbf{y}$ such that $\phi' = \exists \mathbf{x}. c'_1 \wedge \ldots \wedge c'_k$ is $\texttt{false}$, where ϕ' is the formula obtained from ϕ by replacing each variable $y \in \mathbf{y}$ with $\texttt{true}$ if $h_Y(y) = \texttt{true}$ and with $\texttt{false}$ otherwise ($h_Y(y) = \texttt{false}$). Consider now $W = (R, E)$ to be such that (*i*) R contains $H(c')$, and $F_Y(y_i)$ if $h_Y(y_i) = \texttt{true}$ while $T_Y(y_i)$ otherwise (i.e. if $h_Y(y_i) = \texttt{false}$), for each $i = 1, \ldots, m$; and (*ii*) E is the symmetric and transitive closure of the set containing (c, c) for each $c \in \mathsf{dom}(D_\phi \setminus R)$, the pair $\alpha = (c, c')$, and the pair (c_1, c_2). Clearly, we have $W \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$. Furthermore, due to the fact that ϕ' is $\texttt{false}$, by construction of the denial constraints in $\Delta^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, it is not possible to merge all variables $x_i \in \mathbf{x}$ each with either t or f (i ranges from 1 to n) without including other facts in R besides the ones already included. But then, it can be immediately verified that every $W' = (R', E')$ for which $R' \subset R$ is such that $W' \notin \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$.

Since $W = (R, E)$ is such that $W \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and $\alpha \in E$, and since every $W' = (R', E')$ for which $R' \subset R$ is such that $W' \notin \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$, it follows that there must exists at least one $W'' = (R, E'')$ such that $W'' \in \mathsf{Sol}_{\mathrm{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and $\alpha \in E''$ (such E'' can be obtained, e.g. by including the merges of the existentially quantified variables x_i with either t or f, for $i = 1, \ldots, l$, where x_l is the constant representing the last existentially quantified that is possible to merge with either t or f without violating a denial constraint). So, there exists at least one $W'' = (R, E'')$ such that $W'' \in \mathsf{Sol}_{\mathrm{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and $\alpha \in E''$, clearly implying that $() \in q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}(D_\phi, W'')$. Thus, since $W'' \in \mathsf{Sol}_{\mathrm{DEL}}(D_\phi, \Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}})$ and $() \in q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}(D_\phi, W'')$, following Definition 5, we get that $()$ is a DEL-possible answer to $q^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$ on D_ϕ w.r.t. $\Sigma^{\text{POSS,D}}_{\forall\exists\text{3CNF}}$, as required. □

PAR-POSSANS is NP-complete.

Upper Bound: Given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{c}$ of constants, we now show how to check whether $\mathbf{c} \in \text{PAR-}\mathsf{possAns}(q, D, \Sigma)$ in NP in the size of D. We first guess a pair $W = (R, D)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check whether (*i*) $W \in \mathsf{Sol}(D, \Sigma)$ and (*ii*) $\mathbf{c} \in q(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return $\texttt{true}$; otherwise, we return $\texttt{false}$. Correctness of the above procedure is guaranteed by Lemma 1. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 1, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 2, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{c} \in \text{PAR-}\mathsf{possAns}(q, D, \Sigma)$ can be done in NP in the size of D.

Lower Bound: The proof is by a LOGSPACE reduction from the 3SAT problem.

Let us first define the fixed schema $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$, DQ specification $\Sigma^{\text{POSS,C}}_{\text{3SAT}}$ over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$, and CQ $q^{\text{POSS,C}}_{\text{3SAT}}$ over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$. We have $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}} = \{R_{fff}/4,\ R_{fft}/4,\ R_{ftf}/4,\ R_{ftt}/4,\ R_{tff}/4,\ R_{tft}/4,\ R_{ttf}/4,\ R_{ttt}/4,\ V_X/1,\ T/1,\ F/1,\ L/1,\ FV_C/1,\ Prec_C/2,\ C'/2,\ LV_{C'}/2\}$. Informally, similarly to the previous lower bound proofs, the R-predicates are used to store the clauses of ϕ, V_X stores (the constants representing) the existentially quantified variables $\mathbf{x}$, F and T store the constants t and f, respectively, while L stores both t and f. Additionally, the predicate FV_C stores (the constant representing) the first clause c_1 of ϕ, while $Prec_C$ stores pairs of the form (c_i, c_{i+1}) of clauses indicating that clause c_{i+1} comes soon after clause c_i. Finally, for each clause c_i of ϕ, the predicate C' stores pairs of the form (c_i, c'_i), where c'_i is a (constant representing a) copy of clause c_i, while $LV_{C'}$ only stores the pair (c_m, c'_m) for the last clause c_m of ϕ.

The DQ specification $\Sigma^{\text{POSS,C}}_{\text{3SAT}} = \langle \Gamma^{\text{POSS,C}}_{\text{3SAT}}, \Delta^{\text{POSS,C}}_{\text{3SAT}} \rangle$ over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$ is such that $\Gamma^{\text{POSS,C}}_{\text{3SAT}}$ contains the following soft rules over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$:

- $\sigma^T_X = V_X(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of the (constants representing the) existentially quantified variables $\mathbf{x}$ with the constant t
- $\sigma^F_X = V_X(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of the (constants representing the) existentially quantified variables $\mathbf{x}$ with the constant f
- For every $I \in \{fff,\ fft,\ ftf,\ ftt,\ tff,\ tft,\ ttf,\ ttt\}$, there are soft rules:
  - $\sigma^{FV}_I = \exists v_1, v_2, v_3. FV_C(x) \wedge R_I(x, v_1, v_2, v_3) \wedge L(v_1) \wedge L(v_2) \wedge L(v_3) \wedge C'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
  - $\sigma^{Prec}_I = \exists z_c, v_1, v_2, v_3. C'(z_c, z_c) \wedge Prec_C(z_c, x) \wedge R_I(x, v_1, v_2, v_3) \wedge L(v_1) \wedge L(v_2) \wedge L(v_3) \wedge C'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$

  Informally, the soft rule σ^{FV}_I allows the merge between the (constant representing the) clause c_1 and its copy c'_1, but only if all the existential variables occurring in the clause c_1 have each been merged with either t or f. For each $i = 2, \ldots, m$, the soft rule σ^{Prec}_I allows the merge between the (constant representing the) clause c_i with its copy c'_i, but only if the following two conditions hold: (*i*) all the existential variables occurring in the clause c_i have each been merged with either t or f and (*ii*) the (constant representing the) clause c_{i-1} has been previously merged with its copy c'_{i-1}.

Then, $\Delta^{\text{POSS,C}}_{\text{3SAT}}$ comprises the following denial constraints over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$:

- $\delta_{TF} = \neg(\exists y. T(y) \wedge F(y))$, which prevents the merge between the constants t and f. This means that every (constant representing an) existential variable in $\mathbf{x}$ can be merged with either the constant t or the constant f, but not both
- $\delta_{fff} = \neg(\exists c, y_1, y_2, y_3. R_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta_{fft} = \neg(\exists c, y_1, y_2, y_3. R_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta_{ftf} = \neg(\exists c, y_1, y_2, y_3. R_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta_{ftt} = \neg(\exists c, y_1, y_2, y_3. R_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F(y_3))$
- $\delta_{tff} = \neg(\exists c, y_1, y_2, y_3. R_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T(y_3))$
- $\delta_{tft} = \neg(\exists c, y_1, y_2, y_3. R_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F(y_3))$
- $\delta_{ttf} = \neg(\exists c, y_1, y_2, y_3. R_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T(y_3))$
- $\delta_{ttt} = \neg(\exists c, y_1, y_2, y_3. R_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F(y_3))$

Notice that the above denial constraints are nine of the ten denial constraints used in the lower bound proof for DEL-CERTANS and PAR-CERTANS. Consider a clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$. The denial δ_{tft} avoids that, at the same time, variables x_2 and x_1 are merged with the constant f and variable x_4 is merged with the constant t. In other words, once given an assignment to all the variables in $\mathbf{x}$, no clause can be unsatisfied.

Finally, the fixed Boolean CQ over $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$ is $q^{\text{POSS,C}}_{\text{3SAT}} = \exists y. LV_{C'}(y, y)$, asking whether the last (constant representing the) clause c_m has been merged with its copy c'_m.

Given an instance $\phi = \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_m$ of the 3SAT problem, where $\mathbf{x} = (x_1, \ldots, x_n)$, we construct an $\mathcal{S}^{\text{POSS,C}}_{\text{3SAT}}$-database D_ϕ as follows:

- D_ϕ contains the facts $F(f)$, $T(t)$, $L(f)$, $L(t)$, $FV_C(c_1)$, $LV_{C'}(c_m, c'_m)$;
- D_ϕ contains the fact $V_X(x_i)$ for each $i = 1, \ldots, n$;
- D_ϕ contains the fact $Prec_C(c_i, c_{i+1})$ for each $i = 1, \ldots, m-1$;
- D_ϕ contains the fact $C'(c_i, c'_i)$ for each $i = 1, \ldots, m$;
- finally, for each $i = 1, \ldots, m$, if clause c_i is of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), then D_ϕ contains the fact $R_{fff}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{fft}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftf}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftt}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tff}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tft}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttf}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttt}(c_i, v_{i,1}, v_{i,2}, v_{i,3})$), where $v_{i,1}$ (resp. $v_{i,2}, v_{i,3}$) denotes the variable in $\mathbf{x}$ of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ can be constructed in LOGSPACE from an input 3SAT instance ϕ. To conclude the proof of the claimed lower bound, we now show that ϕ is `true` if and only if $()$ is a PAR-possible answer to $q^{\text{POSS,C}}_{\text{3SAT}}$ on D_ϕ w.r.t. $\Sigma^{\text{POSS,C}}_{\text{3SAT}}$.

Claim 6. *ϕ is `true` if and only if* $() \in$ PAR-possAns$(q^{\text{POSS,C}}_{\text{3SAT}}, D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$.

Proof. First, observe that every $W = (R, E)$ such that $W \in \mathsf{Sol}(D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$ must satisfy $(t, f) \notin E$ (i.e. t and f cannot be merged). Indeed, if $T(t)$ (resp. $F(f)$) occurs in R, then by construction of the rules in $\Gamma^{\text{POSS,C}}_{\text{3SAT}}$, the constant t (resp. f) cannot be merged with any other constant. In the case that both $T(t)$ and $F(f)$ occur in $D_\phi \setminus R$, the merge of t with f would cause the violation of the denial constraint δ_{TF}, and so, again, t cannot be merged with f.

Suppose that ϕ is `true`, and let $h_X(\cdot)$ be the function assigning `true` or `false` to each variable $x \in \mathbf{x}$ that witnesses the truth of ϕ. Consider $W = (R, E)$ be such that $R = \emptyset$ and E is the symmetric and transitive closure of the following set S:

- S contains the pair (c, c) for each $c \in \mathsf{dom}(D_\phi)$;
- for each $i = 1, \ldots, n$, if $h_X(x_i) = \texttt{true}$, then S contains the pair (x_i, t); otherwise (i.e. $h_X(x_i) = \texttt{false}$), S contains the pair (y_i, f). Observe that both (x_i, t) and (x_i, f) can be included thanks to the soft rules σ^T_X and σ^F_X, respectively;
- for each $i = 1, \ldots, m$, S contains the pair (c_i, c'_i). Observe that, since all the (constants representing the) existential variables have each been merged with either t or f, the pair (c_1, c'_1) can be included thanks to the soft rules σ^{FV}_I (which one among the various $I \in \{fff,\ fft,\ ftf,\ ftt,\ tff,\ tft,\ ttf,\ ttt\}$ makes (c_1, c'_1) active depend on the form of ϕ), while all the other pairs $(c_2, c'_2), \ldots, (c_m, c'_m)$ (if any, i.e. if $m \geq 2$) can be included one after the other thanks to the soft rules σ^{Prec}_I for $I \in \{fff,\ fft,\ ftf,\ ftt,\ tff,\ tft,\ ttf,\ ttt\}$;
- no other pair is in S.

By construction of the denial constraints in $\Delta^{\text{POSS,C}}_{\text{3SAT}}$, since by assumption ϕ is $\mathtt{true}$, we have that $D_{\phi_E} \models \Delta^{\text{POSS,C}}_{\text{3SAT}}$, and therefore $W \in \mathsf{Sol}_{\text{PAR}}(D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$. Furthermore, since $(c_m, c'_m) \in E$, we have that $() \in q^{\text{POSS,C}}_{\text{3SAT}}(D_\phi, W)$. Thus, since $W \in \mathsf{Sol}_{\text{PAR}}(D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$ and $() \in q^{\text{POSS,C}}_{\text{3SAT}}(D_\phi, W)$, we have that $()$ is a PAR-possible answer to $q^{\text{POSS,C}}_{\text{3SAT}}$ on D_ϕ w.r.t. $\Sigma^{\text{POSS,C}}_{\text{3SAT}}$.

Suppose that $() \in$ PAR-$\mathsf{possAns}(q^{\text{POSS,C}}_{\text{3SAT}}, D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$, i.e. there exists a $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_\phi, \Sigma^{\text{POSS,C}}_{\text{3SAT}})$ and $() \in q^{\text{POSS,C}}_{\text{3SAT}}(D_\phi, W)$. Since $() \in q^{\text{POSS,C}}_{\text{3SAT}}(D_\phi, W)$, we immediately get that $(c_m, c'_m) \in E$. Using a trivial induction argument, it can be immediately proven that (c_m, c'_m) can be included only after including in E all the pairs (c_i, c'_i) for each $i = 1, \ldots, m-1$ (starting from $i = 1$). By construction of the soft rules, this also implies that all the (constants representing the) existential variables have each been merged with either t or f. Consider now the assignment $h_X(\cdot)$ such that, for each $i = 1, \ldots, n$, we have $h_X(x_i) = \mathtt{true}$ if $(x_i, t) \in E$, and $h_X(x_i) = \mathtt{false}$ otherwise (as observed at the beginning of the proof, we cannot have $(t, f) \in E$, which implies that, for no $i = 1, \ldots, n$, we have both $(x_i, t) \in E$ and $(x_i, f) \in E$). Since $D_{\phi_E} \models \Delta^{\text{POSS,C}}_{\text{3SAT}}$, by construction of the denial constraints in $\Delta^{\text{POSS,C}}_{\text{3SAT}}$, we derive that $h_X(\cdot)$ is an assignment witnessing that ϕ is $\mathtt{true}$, as required. □

Before providing the proof of Theorem 4, we introduce some important properties, which are crucial to establish all the results claimed in the theorem.

Lemma 3. *Let δ be a denial constraints over a schema $\mathcal{S}$ and D be an $\mathcal{S}$-database such that $D \not\models \delta$. Then, we have that $D' \not\models \delta$ holds for any $\mathcal{S}$-database D' with $D \subseteq D'$.*

Proof. Trivial to verify. □

Corollary 1. *Let Δ be a denial constraints over a schema $\mathcal{S}$ and D be an $\mathcal{S}$-database such that $D \not\models \Delta$. Then, we have that $D' \not\models \Delta$ holds for any $\mathcal{S}$-database D' with $D \subseteq D'$.*

Proof. Corollary of Lemma 3. □

Lemma 4. *Let δ be a denial constraints over a schema $\mathcal{S}$ without inequality atoms, D be an $\mathcal{S}$-database, and E be an equivalence relation over $\mathsf{dom}(D)$ such that $D_E \not\models \delta_E$. Then, we have that $D_{E'} \not\models \delta_{E'}$ holds for any equivalence relation E' over $\mathsf{dom}(D)$ with $E \subseteq E'$.*

Proof. First, let $\delta = \forall \mathbf{x}.\neg(\phi(\mathbf{x}))$. By construction, any $\mathcal{S}$-database D' is such that $D' \not\models \delta$ if and only if the Boolean CQ $q_\delta = \exists \mathbf{x}.\phi(\mathbf{x})$ holds in D', i.e. $D' \models q_\delta$. Thus, we derive $D_E \models q_{\delta_E}$ because $D_E \not\models \delta_E$ by assumption.

Consider now any equivalence relation E' over $\mathsf{dom}(D)$ with $E \subseteq E'$. Since $E \subseteq E'$, there is a homomorphism from D_E to $D_{E'}$. Since CQs are preserved under homomorphisms and since $D_E \models q_{\delta_E}$, we immediately obtain that $D_{E'} \models q_{\delta_{E'}}$, thus implying that $D_{E'} \not\models \delta_{E'}$, as required. □

Corollary 2. *Let Δ be a set of denial constraints over a schema $\mathcal{S}$ without inequality atoms, D be an $\mathcal{S}$-database, and E be an equivalence relation over $\mathsf{dom}(D)$ such that $(D, E) \not\models \Delta$. Then, we have that $(D, E') \not\models \Delta$ holds for any equivalence relation E' over $\mathsf{dom}(D)$ with $E \subseteq E'$.*

Proof. Corollary of Lemma 4. □

Corollary 3. *Let Δ be a set of denial constraints over a schema $\mathcal{S}$ without inequality atoms, D be an $\mathcal{S}$-database, and (R, E) be a pair with $R \subseteq D$ and E an equivalence relation over $\mathsf{dom}(D')$ such that $(D', E) \not\models \Delta$, where $D' = D \setminus R$. Then, we have that any pair (R', E') with $R' \subseteq R$ and E' an equivalence relation over $\mathsf{dom}(D'')$ with $E \subset E'$ is such that $(D'', E') \not\models \Delta$, where $D'' = D \setminus R'$.*

Proof. Combination of Corollaries 1 and 2. □

Theorem 4. *For restricted DQ specifications, we have that:*

- DEL-OPTREC *and* PAR-OPTREC *are* P*-complete;*
- X-CERTANS *is coNP-complete for* $X \in \{\text{DEL}, \text{PAR}\}$ *and* DEL-POSSANS *is NP-complete;*

Proof. The order we follow for proving the theorem for restricted DQ specifications is as follows: (*i*) we show that DEL-OPTREC and PAR-OPTREC are P-complete; (*ii*) we show that X-CERTANS is coNP-complete for $X \in \{\text{DEL}, \text{PAR}\}$; finally, (*iii*) we show that DEL-POSSANS is NP-complete.

For restricted DQ specifications, X-OPTREC is P-complete for $X \in \{\text{DEL}, \text{PAR}\}$.

Upper Bound for $X = \textsc{Del}$: Given a restricted DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, and a pair $W = (R, E)$, we now show how to check whether $W \notin \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$ in polynomial time in the size of D and W. First, following Definition 4, we have that $W \notin \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$ if and only if either $W \notin \mathsf{Sol}(D, \Sigma)$ or there exists $W' = (R', E')$ such that $W' \in \mathsf{Sol}(D, \Sigma)$ and $W \prec_{\textsc{Del}} W'$. We also recall that, by definition, $W \prec_{\textsc{Del}} W'$ if and only if either (*i*) $R' \subset R$ or (*ii*) $R' \subseteq R$ and $E \subset E'$.

So, as a first step we check whether $\mathcal{W} \notin \mathsf{Sol}(D, \Sigma)$. If this is the case, then we return `true`; otherwise, we continue with the second step. In the second step, we compute the $\mathcal{S}$-database $D' = D \setminus R$ and collect in a set S all those pairs of constants $\alpha = (c, c')$ such that α is active in (D', E) w.r.t. Γ_s and $\alpha \notin E$. Then, for each possible $\alpha \in S$, starting from $E' := \mathsf{EqRel}(E \cup \{\alpha\}, D')$, we repeat the following until a fixpoint is reached: if there is some pair (c, c') of constants occurring in D such that (c, c') is active in (D', E') w.r.t. Γ_h and $(c, c') \notin E'$, then set $E' := \mathsf{EqRel}(E' \cup \{(c, c')\}, D')$. Once the fixpoint is reached, we check whether the obtained E' is such that $(D', E') \models \Delta$. If this is the case for some $\alpha \in S$, then we return `true`; otherwise, we continue with the third step. In the third step, for each possible fact $r \in R$, we compute $R' = R \setminus r$, the $\mathcal{S}$-database $D' = D \setminus R'$, and, starting from $E' := \mathsf{EqRel}(\emptyset, D')$, repeat the following until a fixpoint is reached: if there is some pair (c, c') of constants occurring in D' such that (c, c') is active in (D', E') w.r.t. Γ_h and $(c, c') \notin E'$, then set $E' := \mathsf{EqRel}(E' \cup \{(c, c')\}, D')$. Once the fixpoint is reached, we check whether the obtained E' is such that $(D', E') \models \Delta$. If this is the case for some $r \in R$, then we return `true`. Finally, if the procedure has not yet terminated, then we return `false`.

The above procedure runs in polynomial time in the size of D and W because computing an $\mathcal{S}$-database $D' = D \setminus R$ given D and R can be done in polynomial time, checking whether a pair of constants is active in (D, E) w.r.t. Γ for a given pair (D, E) and a set Γ of rules can be done in polynomial time in the size of D and E, and computing $E' = \mathsf{EqRel}(E, D)$ for a given relation E and $\mathcal{S}$-database D can be clearly done in polynomial time.

The correctness of the above procedure, i.e. the fact that returns `true` if and only if $W \notin \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$, can be obtained using the following observations. The second step of the procedure tries, in all possible ways, to construct a pair $W' = (R, E')$ with $W' \in \mathsf{Sol}(D, \Sigma)$ and $E \subset E'$ (and therefore, $W \prec_{\textsc{Del}} W'$) by "minimally extending" E and check whether such minimal extension leads to a solution for (D, Σ). More precisely, a minimal extension consists in adding to E a single pair of constants $\alpha \notin E$ that is active in (D, E) w.r.t. Γ_s, and then compute an E' by adding the necessary merges to satisfy all the hard rules (clearly, the added α to E can now activate other hard rules). If each such attempt to minimally extending E ends up with an E' such that $(D', E') \not\models \Delta$, where $D' = D \setminus R$, then, due to Corollary 3, we immediately obtain that no pair $W' = (R', E')$ with $R' \subseteq R$ and $E \subset E'$ can be such that $W \prec_{\textsc{Del}} W'$. The third step of the procedure tries, in all possible ways, to construct a pair $W' = (R', E')$ with $W' \in \mathsf{Sol}(D, \Sigma)$ and $R' \subset R$ (and therefore, $W \prec_{\textsc{Del}} W'$) by "re-adding" some fact $\alpha \in R$ to the original $\mathcal{S}$-database D, and then compute an E' by adding the necessary merges to satisfy all the hard rules. If each such attempt to re-adding some fact to the original $\mathcal{S}$-database ends up with a pair (R', E') such that $(D', E') \not\models \Delta$, where $D' = D \setminus R'$, then, due to Corollary 3, we immediately obtain that no pair $W' = (R', E')$ with $R' \subset R$ can be such that $W \prec_{\textsc{Del}} W'$.

Upper Bound for $X = \textsc{Par}$: Given a restricted DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, and a pair $W = (R, E)$, we now show how to check whether $W \notin \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ in polynomial time in the size of D and W. First, following Definition 4, we have that $W \notin \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$ if and only if either $W \notin \mathsf{Sol}(D, \Sigma)$ or there exists $W' = (R', E')$ such that $W' \in \mathsf{Sol}(D, \Sigma)$ and $W \prec_{\textsc{Par}} W'$. We also recall that, by definition, $W \prec_{\textsc{Par}} W'$ if and only if either (*i*) $R' \subset R$ and $E \subseteq E'$ or (*ii*) $R' \subseteq R$ and $E \subset E'$.

We can use a procedure similar to the one used for the case of $X = \textsc{Del}$, except that the third step is modified as follows. For each possible fact $r \in R$, we compute $R' = R \setminus r$, the $\mathcal{S}$-database $D' = D \setminus R'$, and, starting from $E' := \mathsf{EqRel}(E, D')$, repeat the following until a fixpoint is reached: if there is some pair (c, c') of constants occurring in D' such that (c, c') is active in (D', E') w.r.t. Γ_h and $(c, c') \notin E'$, then set $E' := \mathsf{EqRel}(E' \cup \{(c, c')\}, D')$ (clearly, since $r \in D'$, other hard rules can be activated). Once the fixpoint is reached, we check whether the obtained E' is such that $(D', E') \models \Delta$. If this is the case for some $r \in R$, then we return `true`.

The difference between this third step and the third step for the case of $X = \textsc{Del}$ is that here we start with $E' := \mathsf{EqRel}(E, D')$ to seek for a W' that satisfies condition (*i*) $R' \subset R$ and $E \subseteq E'$, whereas for $X = \textsc{Del}$ we start with $E' := \mathsf{EqRel}(\emptyset, D')$ because the condition (*i*) for $X = \textsc{Del}$ only requires $R' \subset R$. Correctness of the above procedure and the polynomial running time in the size of D and W can be obtained similarly as done for the case of $X = \textsc{Del}$.

Lower Bound: The proof can be obtained from [Bienvenu *et al.*, 2022, Theorem 8] by adopting exactly the same line of reasoning used in the lower bound proof for $X = \textsc{Del}$ and $X = \textsc{Par}$ of Theorem 2. Specifically, from [Bienvenu *et al.*, 2022, Theorem 8] we know that there exists a fixed, restricted DQ specification $\Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}}$ over a fixed schema $\mathcal{S}^{\text{RESTR,D/C}}_{\textsc{OptRec}}$ such that, given an $\mathcal{S}^{\text{RESTR,D/C}}_{\textsc{OptRec}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, it is P-hard the problem of deciding whether E is a maximal ER solution for $(D, \Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}})$ in the sense of [Bienvenu *et al.*, 2022, Definition 3], i.e. $E \in \mathsf{ERSol}(D, \Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}})$ and there is no $E' \in \mathsf{ERSol}(D, \Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}})$ such that $E \subset E'$. The reduction from the above problem is as follows: given an $\mathcal{S}^{\text{RESTR,D/C}}_{\textsc{OptRec}}$-database D and an equivalence relation E over $\mathsf{dom}(D)$, we construct in $\textsc{LogSpace}$ a pair $W_E = (R, E)$, where $R = \emptyset$. Since, as already observed in the paper, for any database-specification pair (D, Σ) and equivalence relation E over $\mathsf{dom}(D)$, we have that E is a maximal ER solution for (D, Σ) if and only if $W = (\emptyset, E) \in \mathsf{Sol}_{\textsc{Del}}(D, \Sigma)$ (resp. $W = (\emptyset, E) \in \mathsf{Sol}_{\textsc{Par}}(D, \Sigma)$), we derive that E is a maximal ER solution for $(D, \Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}})$ if and only if $W_E \in \mathsf{Sol}_{\textsc{Del}}(D, \Sigma^{\text{RESTR,D/C}}_{\textsc{OptRec}})$

(resp. $W_E \in \mathsf{Sol}_{\text{PAR}}(D, \Sigma^{\text{RESTR,D/C}}_{\text{OPTREC}})$), thus obtaining the claimed lower bound.

<u>For restricted DQ specifications, X-CERTANS is coNP-complete for $X \in \{\text{DEL}, \text{PAR}\}$.</u>

Upper Bound: Given a restricted DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{c}$ of constants, for both $X = \text{DEL}$ and $X = \text{PAR}$, we now show how to check whether $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ in NP in the size of D, thus obtaining the claimed upper bound. First, following Definition 5, we have that $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ if and only if there exists a W such that $W \in \mathsf{Sol}_X(D, \Sigma)$ and $\mathbf{c} \notin q(D, W)$.

So, we first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{c} \notin q(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ is trivial. As for its running time, we observe that W is polynomially related to D. Furthermore, as shown above in the upper bound of X-OPTREC for restricted DQ specifications, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 2, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, for restricted DQ specifications checking whether $\mathbf{c} \notin X\text{-}\mathsf{certAns}(q, D, \Sigma)$ can be done in NP in the size of D for both $X = \text{DEL}$ and $X = \text{PAR}$.

Lower Bound: The proof is by a LOGSPACE reduction from the complement of 3SAT.

Let us first define the fixed schema $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$, restricted DQ specification $\Sigma^{\text{RESTR,D/C}}_{\text{3SAT}}$ over $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$, and CQ $q^{\text{RESTR,D/C}}_{\text{3SAT}}$ over $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$. We have $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}} = \{F/1,\ T/1,\ R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ V/1,\ O/2\}$. Informally, T and F store the constants t and f. The predicate O stores the pair (o, o') of constants. The predicate V stores (the constants representing) the variables $\mathbf{x}$. Finally, as usual, the R predicates are used to store the clauses of ϕ. For instance, a clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$ occurring in a 3SAT instance ϕ will be represented as $R_{tft}(x_2, x_4, x_1)$.

The restricted DQ specification $\Sigma^{\text{RESTR,D/C}}_{\text{3SAT}} = \langle \Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}, \Delta^{\text{RESTR,D/C}}_{\text{3SAT}} \rangle$ over $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$ is such that $\Delta^{\text{RESTR,D/C}}_{\text{3SAT}} = \{\neg(\exists y. T(y) \wedge F(y))\}$ prevents the merge between constants t and f, and $\Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}$ contains the following soft rules over $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$:

- $\sigma^T_V = V(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of the (constants representing the) existentially quantified variables $\mathbf{x}$ with the constant t
- $\sigma^F_V = V(x) \wedge F(y)$, which simply allows the merge of the (constants representing the) existentially quantified variables $\mathbf{x}$ with the constant f
- $\sigma_{ttt} = \exists u_1, u_2, u_3. R_{ttt}(u_1, u_2, u_3) \wedge F(u_1) \wedge F(u_2) \wedge F(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{ttf} = \exists u_1, u_2, u_3. R_{ttf}(u_1, u_2, u_3) \wedge F(u_1) \wedge F(u_2) \wedge T(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{tft} = \exists u_1, u_2, u_3. R_{tft}(u_1, u_2, u_3) \wedge F(u_1) \wedge T(u_2) \wedge F(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{tff} = \exists u_1, u_2, u_3. R_{tff}(u_1, u_2, u_3) \wedge F(u_1) \wedge T(u_2) \wedge T(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{ftt} = \exists u_1, u_2, u_3. R_{ftt}(u_1, u_2, u_3) \wedge T(u_1) \wedge F(u_2) \wedge F(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{ftf} = \exists u_1, u_2, u_3. R_{ftf}(u_1, u_2, u_3) \wedge T(u_1) \wedge F(u_2) \wedge T(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{fft} = \exists u_1, u_2, u_3. R_{fft}(u_1, u_2, u_3) \wedge T(u_1) \wedge T(u_2) \wedge F(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$
- $\sigma_{fff} = \exists u_1, u_2, u_3. R_{fff}(u_1, u_2, u_3) \wedge T(u_1) \wedge T(u_2) \wedge T(u_3) \wedge O(x, y) \dashrightarrow \mathsf{EQ}(x, y)$

Informally, consider a clause $c_5 = (x_2 \vee \overline{x_4} \vee x_1)$. The soft rule σ_{tft} allows the merge between the constant o and o' but only if (the constants representing the variables) x_2 and x_1 have been previously merged with the constant f and (the constants representing the variables) x_4 has been previously merged with the constant t. In other words, the soft rule σ_{tft} allows the merge between the constant o and o' but only if the clause c_5 is not satisfied under a given assignment to the variables $\mathbf{x}$.

Finally, the fixed Boolean CQ over $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$ is $q^{\text{RESTR,D/C}}_{\text{3SAT}} = \exists y. O(y, y)$, asking whether constants o and o' have been merged.

Given an instance $\phi = \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_m$ of the 3SAT problem, where $\mathbf{x} = (x_1, \ldots, x_n)$, we construct an $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$-database D_ϕ as follows:

- D_ϕ contains the facts $T(t)$, $F(f)$, and $O(o, o')$;
- D_ϕ contains the fact $V(x_i)$ for each $i = 1, \ldots, n$;
- Finally, for each $i = 1, \ldots, m$, if clause c_i is of the form $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$ (resp. $(\overline{v_{i,1}} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(\overline{v_{i,1}} \vee v_{i,2} \vee v_{i,3})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee \overline{v_{i,2}} \vee v_{i,3})$, $(v_{i,1} \vee v_{i,2} \vee \overline{v_{i,3}})$, $(v_{i,1} \vee v_{i,2} \vee v_{i,3})$), then D_ϕ contains the fact $R_{fff}(v_{i,1}, v_{i,2}, v_{i,3})$ (resp. $R_{fft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ftt}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tff}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{tft}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttf}(v_{i,1}, v_{i,2}, v_{i,3})$, $R_{ttt}(v_{i,1}, v_{i,2}, v_{i,3})$), where $v_{i,1}$ (resp. $v_{i,2}$, $v_{i,3}$) denotes the variable in $\mathbf{x}$ of the first (resp. second, third) literal of clause c_i.

It is immediate to verify that D_ϕ can be constructed in LOGSPACE from an input 3SAT instance ϕ. To conclude the proof of the claimed lower bound, we now show that, for both $X = \text{DEL}$ and $X = \text{PAR}$, ϕ is $\texttt{true}$ if and only if $()$ is not an X-certain answer to $q_{3SAT}^{\text{RESTR,D/C}}$ on D_ϕ w.r.t. $\Sigma_{3SAT}^{\text{RESTR,D/C}}$.

Claim 7. *For both $X = \text{DEL}$ and $X = \text{PAR}$, ϕ is $\texttt{true}$ if and only if* $() \notin X\text{-certAns}(q_{3SAT}^{\text{RESTR,D/C}}, D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$.

Proof. Suppose that ϕ is $\texttt{true}$, i.e. there exists an assignment $h_X(\cdot)$ to the variables $\mathbf{x}$ such that $\phi' = c_1' \wedge \ldots \wedge c_m'$ is $\texttt{true}$, where ϕ' is the formula obtained from ϕ by replacing each variable $x \in \mathbf{x}$ with $\texttt{true}$ if $h_X(x) = \texttt{true}$ and with $\texttt{false}$ otherwise ($h_X(x) = \texttt{false}$). Consider $W = (R, E)$ to be such that $R = \emptyset$ and E is the symmetric and transitive closure of the following set S:

- S contains the pair (c, c) for each $c \in \mathsf{dom}(D_\phi)$;
- for each $i = 1, \ldots, n$, if $h_X(x_i) = \texttt{true}$, then S contains the pair (x_i, t); otherwise (i.e. $h_X(x_i) = \texttt{false}$), S contains the pair (x_i, f). Observe that both (x_i, t) and (x_i, f) can be included thanks to the soft rules σ_V^T and σ_V^F, respectively;
- no other pair is in S.

Clearly, $() \notin q_{3SAT}^{\text{RESTR,D/C}}(D_\phi, W)$ holds because $(o, o') \notin E$. We now show that $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$, thus implying $() \notin X\text{-certAns}(q_{3SAT}^{\text{RESTR,D/C}}, D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$ as per Definition 5. Since $R = \emptyset$, by definition, for both $X = \text{DEL}$ and $X = \text{PAR}$, the only way for a $W' = (R', E')$ to be such that $W \prec_X W'$ is that $R' = \emptyset$ and $E \subset E'$. However, by construction of $\Sigma_{3SAT}^{\text{RESTR,D/C}}$ and the fact that ϕ is $\texttt{true}$ under the assignment $h_X(\cdot)$ to the variables $\mathbf{x}$, the pair (o, o') is not active in (D_ϕ, E) w.r.t. $\Sigma_{3SAT}^{\text{RESTR,D/C}}$, immediately implying that every $W' = (\emptyset, E')$ with $E \subset E'$ is such that $W' \notin \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$. Thus, $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$.

Suppose that $() \notin X\text{-certAns}(q_{3SAT}^{\text{RESTR,D/C}}, D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$. By Definition 5, this means that there exists a $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$ and $() \notin q_{3SAT}^{\text{RESTR,D/C}}(D_\phi, W)$ (or, equivalently, $(o, o') \notin E$). Since $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$, we clearly have that (*i*) $O(o, o') \notin R$ and (*ii*) for every $i = 1, \ldots, n$, either $(x_i, t) \in E$ or $(x_i, f) \in E$ (indeed, if either (*i*) or (*ii*) does not hold, then we can immediately construct a W' such that $W \prec_X W'$ and $W' \in \mathsf{Sol}(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$, thus contradicting the fact that $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$). Furthermore, since $(o, o') \notin E$, we soon derive that (o, o') is not active in $((D_\phi \setminus R), E)$ w.r.t. $\Sigma_{3SAT}^{\text{RESTR,D/C}}$ (otherwise, the W' which additionally includes (o, o') in the set of merges is clearly such that $W \prec_X W'$ and $W' \in \mathsf{Sol}(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$, thus contradicting the fact that $W \in \mathsf{Sol}_X(D_\phi, \Sigma_{3SAT}^{\text{RESTR,D/C}})$). But then, consider the assignment $h_X(\cdot)$ such that, for each $i = 1, \ldots, n$, we have $h_X(x_i) = \texttt{true}$ if $(x_i, t) \in E$, and $h_X(x_i) = \texttt{false}$ otherwise (observe that, due to $\Delta_{3SAT}^{\text{RESTR,D/C}}$, for no $i = 1, \ldots, n$ we can have both $(x_i, t) \in E$ and $(x_i, f) \in E$). By construction of $\Sigma_{3SAT}^{\text{RESTR,D/C}}$ and the fact that (o, o') is not active in $((D_\phi \setminus R), E)$ w.r.t. $\Sigma_{3SAT}^{\text{RESTR,D/C}}$, we immediately derive that $h_X(\cdot)$ is an assignment witnessing the fact that ϕ is $\texttt{true}$. □

For restricted DQ specifications, DEL-POSSANS is NP-complete.

Upper Bound: Given a restricted DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{c}$ of constants, we now show how to check whether $\mathbf{c} \in \text{DEL-possAns}(q, D, \Sigma)$ in NP in the size of D. First, following Definition 5, we have that $\mathbf{c} \in \text{DEL-possAns}(q, D, \Sigma)$ if and only if there exists a W such that $W \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$ and $\mathbf{c} \in q(D, W)$.

So, we first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$ and (*ii*) $\mathbf{c} \in q(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return $\texttt{true}$; otherwise, we return $\texttt{false}$. Correctness of the above procedure for checking $\mathbf{c} \in \text{DEL-possAns}(q, D, \Sigma)$ is trivial. As for its running time, we observe that W is polynomially related to D. Furthermore, as shown above in the upper bound of DEL-OPTREC for restricted DQ specifications, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 2, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, for restricted DQ specifications checking whether $\mathbf{c} \in \text{DEL-possAns}(q, D, \Sigma)$ can be done in NP in the size of D.

Lower Bound: We can adopt exactly the same LOGSPACE reduction from the 3SAT problem used in the lower bound proof for PAR-POSSANS of Theorem 3. Specifically, recall the fixed schema $\mathcal{S}_{3SAT}^{\text{POSS,C}}$, DQ specification $\Sigma_{3SAT}^{\text{POSS,C}}$ over $\mathcal{S}_{3SAT}^{\text{POSS,C}}$, and CQ $q_{3SAT}^{\text{POSS,C}}$ over $\mathcal{S}_{3SAT}^{\text{POSS,C}}$ used in that proof. Note that $\Sigma_{3SAT}^{\text{POSS,C}}$ is a restricted DQ specification. Furthermore, given an instance ϕ of the 3SAT problem, recall the $\mathcal{S}_{3SAT}^{\text{POSS,C}}$-database D_ϕ used in that proof.

By construction of $\Sigma_{3SAT}^{\text{POSS,C}}$, it is immediate to verify that $\mathsf{Sol}_{\text{PAR}}(D_\phi, \Sigma_{3SAT}^{\text{POSS,C}}) = \mathsf{Sol}_{\text{DEL}}(D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$ holds for any 3SAT instance ϕ. This clearly implies that $\text{PAR-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$=$\text{DEL-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$ holds for any 3SAT instance ϕ. Furthermore, since Claim 6 shows that ϕ is $\texttt{true}$ if and only if $() \in \text{PAR-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$, and since $\text{PAR-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$=$\text{DEL-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$ holds for any 3SAT instance ϕ, we derive that ϕ is $\texttt{true}$ if and only if $() \in \text{DEL-possAns}(q_{3SAT}^{\text{POSS,C}}, D_\phi, \Sigma_{3SAT}^{\text{POSS,C}})$, thus obtaining the claimed lower bound. □

Before providing the proof of Theorem 5, we observe that, for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, the language corresponding to the decision problem X-MICERTANS (i.e. the set of instances to which the answer is "yes") can be equivalently defined as the intersection of the languages associated to two decision problems, namely X-SETCERTANS and X-NOBETTERCERTANS: given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q, and a tuple $\mathbf{C}$ of sets of constants, X-SETCERTANS and X-NOBETTERCERTANS are the problems of deciding whether $\mathbf{C} \in X\text{-SetCert}(q, D, \Sigma)$ and whether there is no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, respectively.

Analogously, for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$, the language associated to the decision problem X-MIPOSSANS can be equivalently defined as the intersection of the languages associated to two decision problems, namely X-SETPOSSANS and X-NOBETTERPOSSANS: given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q, and a tuple $\mathbf{C}$ of sets of constants, X-SETPOSSANS and X-NOBETTERPOSSANS are the problems of deciding whether $\mathbf{C} \in X\text{-SetPoss}(q, D, \Sigma)$ and whether there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, respectively.

We now introduce two lemmata, which can be seen as the analogous of Lemmata 1 and 2 for set-answers.

Lemma 5. *Let Σ be a DQ specification over a schema $\mathcal{S}$, D be an $\mathcal{S}$-database, q be an n-ary CQ over $\mathcal{S}$, and $\mathbf{C} = (C_1, \ldots, C_n)$ be an n-tuple of sets of constants. We have that $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$ if and only if $\mathbf{C} \in \overline{q}(D, W)$ for some $W \in \text{Sol}(D, \Sigma)$.*

Proof. First, suppose that $\mathbf{C} \notin \overline{q}(D, W)$ for every $W \in \text{Sol}(D, \Sigma)$. Then, following Definition 7, we have that $\mathbf{C} \notin \text{PAR-SetPoss}(q, D, \Sigma)$.

Now, suppose that $\mathbf{C} \in \overline{q}(D, W)$ for some $W \in \text{Sol}(D, W)$, where $W = (R, E)$. Since $W \in \text{Sol}(D, \Sigma)$, following Definition 4, we have that either $W \in \text{Sol}_{\text{PAR}}(D, \Sigma)$ or there exists at least one pair $W' = (R', E')$ such that $W' \in \text{Sol}_{\text{PAR}}(D, \Sigma)$ and $W \prec_{\text{PAR}} W'$. In the former case, following Definition 7, we immediately get that $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$. Consider now the latter case. According to Definition 6, $\mathbf{C} \in \overline{q}(D, W)$ implies that the following holds: (*i*) C_i contains constants in the same equivalence class in E, for each $i = 1, \ldots, n$, and (*ii*) there exists a tuple of constants $\mathbf{c} = (c_1, \ldots, c_n)$, with $c_i \in C_i$ for each $i = 1, \ldots, n$, such that $\mathbf{c} \in q(D, W)$. Since $\mathbf{c} \in q(D, W)$ and $W \prec_{\text{PAR}} W'$, using exactly the same arguments in the "if part" proof of Lemma 1, we derive that $\mathbf{c} \in q(D, W')$ as well. Furthermore, $W \prec_{\text{PAR}} W'$ implies that $E \subseteq E'$, and therefore the following holds: (*i*) C_i contains constants in the same equivalence class in E', for each $i = 1, \ldots, n$, and (*ii*) the above tuple of constants $\mathbf{c} = (c_1, \ldots, c_n)$, with $c_i \in C_i$ for each $i = 1, \ldots, n$, is such that $\mathbf{c} \in q(D, W')$. It follows that $\mathbf{C} \in \overline{q}(D, W')$ as well. Thus, since $\mathbf{C} \in \overline{q}(D, W')$ for a $W' \in \text{Sol}_{\text{PAR}}(D, \Sigma)$, following Definition 7, we get that $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$, as required. □

Lemma 6. *Let $\mathcal{S}$ be a schema, D be an $\mathcal{S}$-database, q be an n-ary CQ, $\mathbf{C} = (C_1, \ldots, C_n)$ be an n-tuple of sets of constants, and $W = (R, E)$ be a pair such that $R \subseteq D$ and E is an equivalence relation over $\text{dom}(D \setminus R)$. Then, checking whether $\mathbf{C} \in \overline{q}(D, W)$ can be done in polynomial time in the size of D and W.*

Proof. We recall that, according to Definition 6, $\mathbf{C} \in \overline{q}(D, W)$ if and only if (*i*) C_i contains constants in the same equivalence class in E, for each $i = 1, \ldots, n$, and (*ii*) there exists an n-tuple of constants $\mathbf{c} = (c_1, \ldots, c_n)$, with $c_i \in C_i$ for each $i = 1, \ldots, n$, such that $\mathbf{c} \in q(D, W)$. Clearly, condition (*i*) can be checked in polynomial time in the size of E.

Once established that condition (*i*) is satisfied (otherwise, we simply return `false`), by definition of the set $q(D, W)$ we have two possible cases: either $\mathbf{c}' \in q(D, W)$ holds for any possible n-tuple of constants $\mathbf{c}' = (c'_1, \ldots, c'_n)$ such that $c'_i \in C_i$ for each $i = 1, \ldots, n$, or no n-tuple of constants $\mathbf{c}' = (c'_1, \ldots, c'_n)$, with $c'_i \in C_i$ for each $i = 1, \ldots, n$, is such that $\mathbf{c}' \in q(D, W)$ (or equivalently condition (*ii*) is not satisfied). This easily follows from the fact that condition (*i*) is satisfied and the definition of set of answers to a query q over a schema $\mathcal{S}$ w.r.t. an $\mathcal{S}$-database D and an equivalence relation E. Thus, in order to check condition (*ii*), it is enough to pick any constant c_i from C_i, for $i = 1, \ldots, n$, and then check whether the resulting n-tuple of constants $\mathbf{c} = (c_1, \ldots, c_n)$ is such that $\mathbf{c} \in q(D, W)$. Since due to Lemma 2 checking whether $\mathbf{c} \in q(D, W)$ can be done in polynomial time in the size of D and W, we immediately get an overall procedure for checking whether $\mathbf{C} \in \overline{q}(D, W)$ that runs in polynomial time in the size of D and W. □

As mentioned in a footnote in the paper, we recall that the complexity classes BH(2) (a.k.a. DP) and BH$_3(2)$ (a.k.a. DP$_2$) are the second level of the Boolean hierarchy over NP sets and over Σ_2^p sets, respectively [Chang and Kadin, 1996]. Equivalently, a decision problem is in BH(2) (resp. BH$_3(2)$) if and only if its set of yes-instances is the intersection of the yes-instances of a decision problem in NP (resp. Σ_2^p) and the yes-instances of a decision problem in coNP (resp. Π_2^p).

We also recall that SAT-UNSAT is the prototypical BH(2)-complete problem of deciding, given two CNF formulae ϕ and ϕ', whether ϕ is `true` and ϕ' is `false`. With a trivial generalization of the arguments given in the proof of [**?**, Theorem 17.1] for showing that SAT-UNSAT is BH(2)-hard, we now prove the hardnesses of two decision problems which will be used in the proof of Theorem 5 to show hardnesses of our decision problems of interest. The first one is 3CNF-NO3CNF: given two 3SAT formulae ϕ and ϕ', 3CNF-NO3CNF is the problem of deciding whether ϕ is `true` and ϕ' is `false`. The second one is $\forall\exists$3CNF-NO$\forall\exists$3CNF: given two $\forall\exists$3CNF formulae ϕ and ϕ', $\forall\exists$3CNF-NO$\forall\exists$3CNF is the problem of deciding whether ϕ is `true` and ϕ' is `false`.

Lemma 7. 3CNF-NO3CNF *is BH(2)-hard.*

Proof. The proof can be obtained exactly as in the hardness proof of [**?**, Theorem 17.1] by replacing SAT with 3SAT and SAT-UNSAT with 3CNF-NO3CNF. □

Lemma 8. ∀∃3CNF-NO∀∃3CNF *is* $BH_3(2)$*-hard.*

Proof. By generalizing the arguments given in the proof of [**?**, Theorem 17.1], we now show that any decision problem in $\mathrm{BH}_3(2)$ can be reduced in polynomial time to ∀∃3CNF-NO∀∃3CNF. Consider any decision problem P in $\mathrm{BH}_3(2)$. By definition of the $\mathrm{BH}_3(2)$ complexity class, there exist two decision problems P_1 and P_2 such that (*i*) P_1 is in Σ_2^p, (*ii*) P_2 is in Π_2^p, and (*iii*) the language corresponding to P is the intersection of the languages corresponding to P_1 and P_2. Since ∀∃3CNF is Σ_2^p-complete, we know that there is a polynomial time reduction R_1 from P_1 to ∀∃3CNF and a polynomial time reduction R_2 from the complement of P_2 to ∀∃3CNF, i.e. given an instance x for the problem P_1 (resp. P_2), we have that x is a "yes" instance of P_1 (resp. P_2) if and only if the ∀∃3CNF formula $R_1(x)$ is `true` (resp. $R_2(x)$ is `false`). The polynomial time reduction R from P to ∀∃3CNF-NO∀∃3CNF is this, for any input x:

$$R(x) = (R_1(x), R_2(x)).$$

We have that $R(x)$ is a "yes" instance of ∀∃3CNF-NO∀∃3CNF if and only if $R_1(x)$ is `true` and $R_2(x)$ is `false`, which is the case if and only if x is a "yes" instance of both P_1 and P_2, or equivalently x is a "yes" instance of P. □

We are now ready to face Theorem 5's proof.

Theorem 5. X-MICERTANS *is* DP_2*-complete*[3] *for any* $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$*,* X-MIPOSSANS *is* DP_2*-complete for* $X \in \{\textsc{Mer}, \textsc{Del}\}$*, and* PAR-MIPOSSANS *is DP-complete.*

Proof. We first show that X-MICERTANS is $\mathrm{BH}_3(2)$-complete for any $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we then show that X-MIPOSSANS is $\mathrm{BH}_3(2)$-complete for both $X = \textsc{Mer}$ and $X = \textsc{Del}$, and finally we show that PAR-MIPOSSANS is BH(2)-complete.

<u>X-MICERTANS is $\mathrm{BH}_3(2)$-complete for any $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$.</u>

Upper Bound: Due to the remark preceding Lemma 5, it is enough to show that, for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, X-SETCERTANS and X-NOBETTERCERTANS are in Π_2^p and in Σ_2^p in data complexity, respectively.

As for X-SETCERTANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we now show how to check whether $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ in Σ_2^p in the size of D, thus obtaining that X-SETCERTANS is in Π_2^p in data complexity. We first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{C} \notin \overline{q}(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ directly follows from the definition of the set $X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ of set X-certain answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 2, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ can be done in NP in the size of D for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$.

As for X-NOBETTERCERTANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, for each $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we need to show that checking whether there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in Σ_2^p in the size of D. Let $\mathbf{C} = (C_1, \ldots, C_n)$ and let us call an n-tuple $\mathbf{C}' = (C'_1, \ldots, C'_n)$ of sets of constants a *minimal more informative extension of* $\mathbf{C}$ if there exists a natural number $j \in [1, n]$ and a constant $c \in \mathsf{dom}(D)$ such that (*i*) $C'_j = C_j \cup \{c\}$, (*ii*) $c \notin C_j$, and (*iii*) $C_i = C'_i$ for any $i = 1, \ldots, n$ with $i \neq j$. Moreover, for a pair $p = (j, c)$ of a natural number $j \in [1, n]$ and a constant $c \in \mathsf{dom}(D)$ such that $c \notin C_j$, we let $\mathbf{C}_p$ be the minimal more informative extension of $\mathbf{C}$ such that $\mathbf{C}_p = (C_1, \ldots, C_{j-1}, C_j \cup \{c\}, C_{j+1}, \ldots, C_n)$. Observe that, if m is the cardinality of the set $\mathsf{dom}(D)$, i.e. the number of constants used in D, then the number of minimal more informative extensions of $\mathbf{C}$ cannot be more than $n * m$, and therefore they are polynomial in the size of D.

So, for each possible pair $p = (j, c)$ of a natural number $j \in [1, n]$ and a constant $c \in \mathsf{dom}(D)$ such that $c \notin C_j$, we guess a pair $W_p = (R_p, E_p)$, where $R_p \subseteq D$ and E_p is an equivalence relation over $\mathsf{dom}(D \setminus R_p)$. We then check whether both (*i*) $W_p \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{C}_p \notin \overline{q}(D, W_p)$ hold (and therefore $\mathbf{C}_p \notin X\text{-}\mathsf{SetCert}(q, D, W_p)$). If *each* pair p as above satisfies both conditions (*i*) and (*ii*), then we return `true`; otherwise, we return `false`. Correctness of the above procedure, i.e. the fact that returns `true` if and only if there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, is guaranteed by the following trivial property: if a tuple $\mathbf{C}'$ of sets of constants is such that $\mathbf{C}' \in \overline{q}(D, \Sigma)$, then any tuple $\mathbf{C}''$ of sets of constants for which $\mathbf{C}'$ is strictly more informative than $\mathbf{C}''$ is such that $\mathbf{C}'' \in \overline{q}(D, \Sigma)$. This immediately

[3] We recall that the complexity classes DP (a.k.a. BH(2)) and DP_2 (a.k.a. $\mathrm{BH}_3(2)$) are the second level of the Boolean hierarchy of NP sets and of Σ_2^p sets, respectively [Chang and Kadin, 1996].

implies that if there exists a tuple $\mathbf{C}'$ of sets of constants such that $\mathbf{C}' \in X\text{-SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, then there must exist a tuple $\mathbf{C}_p$ of sets of constants such that $\mathbf{C}_p$ is a minimal more informative extension of $\mathbf{C}$ for which $\mathbf{C}_p \in X\text{-SetCert}(q, D, \Sigma)$. As for its running time, we observe that each W_p is polynomially related to D. Furthermore, due to Theorem 2, for each p as above, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W_p (and therefore, in the size of D as well because W_p is polynomially related to D). Finally, due to Lemma 6, for each p as above, condition (*ii*) can be checked in polynomial time in the size of D and W_p (and therefore, in the size of D as well because W_p is polynomially related to D). So, overall, checking whether there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in Σ_2^p in the size of D for each $X \in \{\text{MER}, \text{DEL}, \text{PAR}\}$.

Lower Bound for $X = $ MER: The proof is by a LOGSPACE reduction from the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, shown to be $\text{BH}_3(2)$-hard in Lemma 8.

We define the fixed schema $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$, DQ specification $\Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$, and CQ $q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$. We have $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists} = \{R_{fff}/4,\ R_{fft}/4,\ R_{ftf}/4,\ R_{ftt}/4,\ R_{tff}/4,\ R_{tft}/4,\ R_{ttf}/4,\ R_{ttt}/4,\ V_Y/1,\ T/1,\ F/1,\ P/4,\ T_X/1,\ F_X/1,\ O/2,\ R'_{fff}/4,\ R'_{fft}/4,\ R'_{ftf}/4,\ R'_{ftt}/4,\ R'_{tff}/4,\ R'_{tft}/4,\ R'_{ttf}/4,\ R'_{ttt}/4,\ V'_Y/1,\ P'/4,\ T'_X/1,\ F'_X/1,\ O'/2\}$. Informally, the predicates R_I and R'_I, for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, are used to store the clauses of ϕ and ϕ', respectively. The predicates V_Y and V'_Y store, respectively, (the constants representing) the universally quantified variables $\mathbf{y}$ of ϕ and $\mathbf{y}'$ of ϕ'. Both T_X and F_X (resp. T'_X and F'_X) store (the constants representing) the existentially quantified variables $\mathbf{x}$ of ϕ (resp. $\mathbf{x}'$ of ϕ'). Furthermore, the predicate T and F only store the constant t and f, respectively, while O and O' store, respectively, the pair (o_1, o_2) and the pair (o_2, o_3). Finally, consider a clause $c_5 = (y_2 \vee \overline{x_4} \vee x_1)$ (resp. $c'_5 = (y'_2 \vee \overline{x'_4} \vee x'_1)$) occurring in ϕ (resp. ϕ'). The predicate P (resp. P') will store two quadruples of the form $(c_5, x_4, a^{c_5}_{x_4}, b^{c_5}_{x_4})$ and $(c_5, x_1, a^{c_5}_{x_1}, b^{c_5}_{x_1})$ (resp. $(c'_5, x'_4, a^{c'_5}_{x'_4}, b^{c'_5}_{x'_4})$ and $(c'_5, x'_1, a^{c'_5}_{x'_1}, b^{c'_5}_{x'_1})$), where, e.g. $a^{c_5}_{x_4}$ and $b^{c_5}_{x_4}$ (resp. $a^{c'_5}_{x'_4}$ and $b^{c'_5}_{x'_4}$) are constants representing the fact that the existentially quantified variable x_4 and (resp. x'_4) occur in clause c_5 (resp. c'_5) of ϕ (resp. ϕ'). Note that the predicates $R_{fff}/4,\ R_{fft}/4,\ R_{ftf}/4,\ R_{ftt}/4,\ R_{tff}/4,\ R_{tft}/4,\ R_{ttf}/4,\ R_{ttt}/4,\ V_Y/1,\ P/4,\ T_X/1,\ F_X/1$ play exactly the same role as in the lower bound proof for MER-CERTANS for representing ϕ, while the predicates $R'_{fff}/4,\ R'_{fft}/4,\ R'_{ftf}/4,\ R'_{ftt}/4,\ R'_{tff}/4,\ R'_{tft}/4,\ R'_{ttf}/4,\ R'_{ttt}/4,\ V'_Y/1,\ P'/4,\ T'_X/1,\ F'_X/1$ do the same for representing ϕ'.

Recall the DQ specification $\Sigma^{\text{CERT,M}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}, \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \rangle$ used in the lower bound proof for MER-CERTANS of Theorem 3. The DQ specification $\Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists} = \langle \Gamma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, \Delta^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists} \rangle$ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$ is such that:

- $\Gamma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists} = \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \cup \Gamma'$, where Γ' is obtained from $\Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ by replacing every occurrence of the predicate name V_Y (resp. O, P, T_X, F_X, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name V'_Y (resp. O', P', T'_X, F'_X, R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, since $\sigma^T_Y \in \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, then ${\sigma'}^T_Y = V'_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$ occurs in Γ'. As another example, since $\sigma^f_{ftf,1} \in \Gamma^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, then ${\sigma'}^f_{ftf,1} = \exists c, v_1, v_2, v_3. P'(c, v_1, x, y) \wedge F'_X(v_1) \wedge R'_{ftf}(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x, y)$ occurs in Γ';
- $\Delta^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists} = \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}} \cup \Delta'$, where Δ' is obtained from $\Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$ by replacing every occurrence of the predicate name O (resp. T_X, F_X, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name O' (resp. T'_X, F'_X, R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, since $\delta^1_{ftf} \in \Delta^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$, then ${\delta'}^1_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F'_X(y_2) \wedge T'_X(y_3))$ occurs in Δ'.

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$ is $q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}(x) = O(x, x)$.

Given an instance ϕ of the $\forall\exists$3CNF problem, recall the $\mathcal{S}^{\text{CERT,M}}_{\forall\exists\text{3CNF}}$-database D_ϕ used in the lower bound proof for MER-CERTANS Theorem 3. Then, given an instance (ϕ, ϕ') of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, where $\phi = \forall \mathbf{y}. \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_k$ and $\phi' = \forall \mathbf{y}'. \exists \mathbf{x}'. c'_1 \wedge \ldots \wedge c'_{k'}$ with $\mathbf{y} = (y_1, \ldots, y_m)$, $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y}' = (y'_1, \ldots, y'_{m'})$, and $\mathbf{x}' = (x'_1, \ldots, x'_{n'})$, we construct an $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$-database $D_{(\phi,\phi)} = D_\phi \cup D'_{\phi'}$, where $D'_{\phi'}$ represents ϕ' exactly as D_ϕ does for ϕ, i.e. $D'_{\phi'}$ is as follows:

- $D'_{\phi'}$ contains the fact $V'_Y(y'_i)$ for each $i = 1, \ldots, m'$;
- $D'_{\phi'}$ contains the fact $O'(o_2, o_3)$, and the facts $T'_X(x'_i)$ and $F'_X(x'_i)$ for each $i = 1, \ldots, n'$;
- for each clause c'_i (i ranges from 1 to k') with no occurrences of universally quantified variables in $\mathbf{y}'$, $D'_{\phi'}$ contains the facts $P'(c'_i, v'_{i,1}, a^{c'_i}_{v'_{i,1}}, b^{c'_i}_{v'_{i,1}})$, $P'(c'_i, v'_{i,2}, a^{c'_i}_{v'_{i,2}}, b^{c'_i}_{v'_{i,2}})$, and $P'(c'_i, v'_{i,3}, a^{c'_i}_{v'_{i,3}}, b^{c'_i}_{v'_{i,3}})$, where $v'_{i,1}$ (resp. $v'_{i,2}$, $v'_{i,3}$) denotes the existentially quantified variable of the first (resp. second, third) literal of clause c'_i;
- for each clause c'_i (i ranges from 1 to k') with exactly one occurrence of a universally quantified variable in $\mathbf{y}'$, $D'_{\phi'}$ contains the facts $P'(c'_i, v'_{i,2}, a^{c'_i}_{v'_{i,2}}, b^{c'_i}_{v'_{i,2}})$ and $P'(c'_i, v'_{i,3}, a^{c'_i}_{v'_{i,3}}, b^{c'_i}_{v'_{i,3}})$, where $v'_{i,2}$ and $v'_{i,3}$ denote the existentially quantified variables of the second and the third, respectively, literal of clause c_i;

- for each clause c'_i (i ranges from 1 to k') with exactly two occurrences of (not necessarily distinct) universally quantified variable(s) in $\mathbf{y}'$, $D'_{\phi'}$ contains the fact $P'(c'_i, v'_{i,3}, a^{c'_i}_{v'_{i,3}}, b^{c'_i}_{v'_{i,3}})$, where $v'_{i,3}$ denotes the existentially quantified variable of the third literal of clause c'_i;
- Finally, for each $i = 1, \ldots, k'$, if clause c'_i is of the form $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$ (resp. $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee v'_{i,3})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(v'_{i,1} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee v'_{i,2} \vee v'_{i,3})$), then $D'_{\phi'}$ contains the fact $R'_{fff}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$ (resp. $R'_{fft}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftf}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftt}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tff}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tft}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttf}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttt}(c'_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$), where $v'_{i,1}$ (resp. $v'_{i,2}, v'_{i,3}$) denotes the variable in $\mathbf{x}' \cup \mathbf{y}'$ of the first (resp. second, third) literal of clause c'_i.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input ∀∃3CNF-NO∀∃3CNF instance (ϕ, ϕ'). To conclude the proof of the claimed lower bound, we now show that (ϕ, ϕ') is a "yes" instance of the ∀∃3CNF-NO∀∃3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{o_1, o_2\})$ is a most informative MER-certain answer to $q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$.

Claim 8. *ϕ is `true` and ϕ' is `false` if and only if* $(\{o_1, o_2\}) \in$ MER-MIcertAns$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$.

Proof. Suppose that ϕ is `true` and ϕ' is `false`. Using exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can immediately derive the following: (*i*) since ϕ is `true`, we have that $(o_1, o_2) \in E$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$; (*ii*) since ϕ' is `false`, we have that there exists a $W' = (R', E')$ such that $W' \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $(o_2, o_3) \notin E'$. Due to (*i*), we easily derive that $(\{o_1, o_2\}) \in \overline{q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2\}) \in$ MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. Furthermore, due to (*ii*), we have that $(\{o_1, o_2, o_3\}) \notin \overline{q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W')$ for at least one $W' \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2, o_3\}) \notin$ MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. By construction, it follows that $(\{o_1, o_2\})$ is most informative in MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$, i.e. $(\{o_1, o_2\}) \in$ MER-MIcertAns$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$.

Suppose now that (ϕ, ϕ') is a "no" instance of the ∀∃3CNF-NO∀∃3CNF problem, i.e. either ϕ is `false` or ϕ' is `true`. Assume first that ϕ is `false`. Using exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can immediately derive that there exists at least one $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $O(o_1, o_2) \in R$. For such W, we clearly have that $(\{o_1, o_2\}) \notin \overline{q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$, and therefore $(\{o_1, o_2\}) \notin$ MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. It follows that $(\{o_1, o_2\}) \notin$ MER-MIcertAns$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. Assume now that ϕ is `true`, and thus also ϕ' is `true`. Using exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can immediately derive that both $(o_1, o_2) \in E$ and $(o_2, o_3) \in E$ (and therefore, $(o_1, o_3) \in E$ due to transitivity) hold for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. By construction, this means that $(\{o_1, o_2, o_3\}) \in \overline{q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ holds for every $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2, o_3\}) \in$ MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$. Since $(\{o_1, o_2, o_3\}) \in$ MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $(\{o_1, o_2, o_3\})$ is strictly more informative than $(\{o_1, o_2\})$, we soon derive that $(\{o_1, o_2\})$ cannot be most informative in MER-SetCert$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2\}) \notin$ MER-MIcertAns$(q^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists})$ also in this case. □

Lower Bound for X = DEL and X = PAR: The proof is again by a LOGSPACE reduction from the ∀∃3CNF-NO∀∃3CNFproblem.

We define the fixed schema $\mathcal{S}^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$, DQ specification $\Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$, and CQ $q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$. We have $\mathcal{S}^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists} = \{R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ V_Y/1,\ FV_X/1,\ LV_X/1,\ Prec_X/2,\ T/1,\ F/1,\ L/1,\ C/2,\ C'/2,\ R'_{fff}/3,\ R'_{fft}/3,\ R'_{ftf}/3,\ R'_{ftt}/3,\ R'_{tff}/3,\ R'_{tft}/3,\ R'_{ttf}/3,\ R'_{ttt}/3,\ V'_Y/1,\ FV'_X/1,\ LV'_X/1,\ Prec'_X/2,\ G/2,\ G'/2\}$. Informally, the predicates R_I and R'_I, for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, are used to store the clauses of ϕ and ϕ', respectively. The predicates V_Y and V'_Y store, respectively, (the constants representing) the universally quantified variables $\mathbf{y}$ of ϕ and $\mathbf{y}'$ of ϕ'. The predicates FV_X and LV_X store (the constants representing) the first and the last existentially quantified variables $\mathbf{x}$ of ϕ, respectively, and $Prec_X$ stores pairs of the form (x_i, x_{i+1}) of existential variables indicating that variable x_{i+1} comes soon after variable x_i. Similarly, the predicates FV'_X and LV'_X store (the constants representing) the first and the last existentially quantified variables $\mathbf{x}'$ of ϕ', respectively, and $Prec'_X$ stores pairs of the form (x'_i, x'_{i+1}) of existential variables indicating that variable x'_{i+1} comes soon after variable x'_i. Furthermore, the predicate T and F only store the constant t and f, respectively, while the predicate L stores both the constants t and f. Finally, C, C', G, and G' only store the pair of constants (c_1, c_2), (c, c'), (c_2, c_3), and (c', c''), respectively. Note that the predicates $R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ V_Y/1,\ FV_X/1,\ LV_X/1,\ Prec_X/1$ play exactly the same role as in the lower bound proof for DEL-CERTANS and PAR-CERTANS for representing ϕ, while the predicates $R'_{fff}/3$, $R'_{fft}/3$, $R'_{ftf}/3$, $R_{ftt}/3$, $R'_{tff}/3$, $R'_{tft}/3$, $R'_{ttf}/3$, $R'_{ttt}/3$, $V'_Y/1$, $FV'_X/1$, $LV'_X/1$, $Prec'_X/1$ do the same for representing ϕ'.

Recall the DQ specification $\Sigma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}} = \langle \Gamma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}, \Delta^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}} \rangle$ used in the lower bound proof for DEL-CERTANS and PAR-CERTANS. The DQ specification $\Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists} = \langle \Gamma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, \Delta^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists} \rangle$ over $\mathcal{S}^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$ is such that:

- $\Gamma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists} = \Gamma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}} \cup \Gamma'$, where Γ' is obtained from $\Gamma^{\text{CERT,M}}_{\forall\exists 3\text{CNF}}$ by replacing every occurrence of the predicate name V_Y (resp. C', FV_X, $Prec_X$, LV_X, and C) with the predicate name V'_Y (resp. G', FV'_X, $Prec'_X$, LV'_X, and G). For example, since $\sigma^T_Y \in \Gamma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$, then ${\sigma'}^T_Y = V'_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x,y)$ occurs in Γ'. As another example, since $\sigma_{C_1,C_2} \in \Gamma^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$, then ${\sigma'}_{C_1,C_2} = \exists z. G(x,y) \wedge LV'_X(z) \wedge L(z) \dashrightarrow \mathsf{EQ}(x,y)$ occurs in Γ';
- $\Delta^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists} = \Delta^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}} \cup \Delta'$, where Δ' is obtained from $\Delta^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$ by replacing every occurrence of the predicate name C' (resp. C, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name G' (resp. G, R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, since $\delta_C \in \Delta^{\text{CERT,M}}_{\forall\exists 3\text{CNF}}$, then ${\delta'}_G = \neg(\exists y, y_1, y_2. G'(y,y) \wedge G(y_1,y_2) \wedge y_1 \neq y_2)$ occurs in Δ'. As another example, since $\delta_{ftf} \in \Delta^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$, then ${\delta'}_{ftf} = \neg(\exists y_1,y_2,y_3. R'_{ftf}(y_1,y_2,y_3) \wedge T(y_1) \wedge F(y_2) \wedge T(y_3))$ occurs in Δ'.

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$ is $q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}(x) = C'(x,x)$.

Given an instance ϕ of the $\forall\exists$3CNF problem, recall the $\mathcal{S}^{\text{CERT,D/C}}_{\forall\exists 3\text{CNF}}$-database D_ϕ used in the lower bound proof for DEL-CERTANS and PAR-CERTANS. Then, given an instance (ϕ,ϕ') of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, where $\phi = \forall \mathbf{y}.\exists \mathbf{x}. c_1 \wedge \ldots \wedge c_k$ and $\phi' = \forall \mathbf{y}'.\exists \mathbf{x}'. c'_1 \wedge \ldots \wedge c'_{k'}$ with $\mathbf{y} = (y_1,\ldots,y_m)$, $\mathbf{x} = (x_1,\ldots,x_n)$, $\mathbf{y}' = (y'_1,\ldots,y'_{m'})$, and $\mathbf{x}' = (x'_1,\ldots,x'_{n'})$, we construct an $\mathcal{S}^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$-database $D_{(\phi,\phi)} = D_\phi \cup D'_{\phi'}$, where $D'_{\phi'}$ represents ϕ' exactly as D_ϕ does for ϕ, i.e. $D'_{\phi'}$ is as follows:

- $D'_{\phi'}$ contains the fact $V'_Y(y'_i)$ for each $i = 1,\ldots,m'$, the fact $FV'_X(x'_1)$, the fact $Prec'_X(x'_i, x'_{i+1})$ for each $i = 1,\ldots,n'-1$, the fact $LV'_X(x'_m)$, and the two facts $G'(c',c'')$ and $G(c_2,c_3)$;
- for each clause c'_i of the form $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$ (resp. $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee v'_{i,3})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(v'_{i,1} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee v'_{i,2} \vee v'_{i,3})$), $D'_{\phi'}$ contains the fact $R'_{fff}(v'_{i,1},v'_{i,2},v'_{i,3})$ (resp. $R'_{fft}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{ftf}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{ftt}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{tff}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{tft}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{ttf}(v'_{i,1},v'_{i,2},v'_{i,3})$, $R'_{ttt}(v'_{i,1},v'_{i,2},v'_{i,3})$), where $v'_{i,1}$ (resp. $v'_{i,2}$, $v'_{i,3}$) denotes the variable in $\mathbf{x}' \cup \mathbf{y}'$ of the first (resp. second, third) literal of clause c'_i.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input $\forall\exists$3CNF-NO$\forall\exists$3CNF instance (ϕ,ϕ'). To conclude the proof of the claimed lower bound, we now show that, for both $X = $ DEL and $X = $ PAR, (ϕ,ϕ') is a "yes" instance of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{c,c'\})$ is a most informative X-certain answer to $q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}$.

Claim 9. *For both $X = $ DEL and $X = $ PAR, we have that ϕ is `true` and ϕ' is `false` if and only if $(\{c,c'\}) \in X$-$\mathsf{MIcertAns}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$.*

Proof. Suppose that ϕ is `true` and ϕ' is `false`. Using exactly the same consideration as in the lower bound proof for X-CERTANS of Theorem 3, we can immediately derive the following: (*i*) since ϕ is `true`, we have that $(c,c') \in E$ for every $W = (R,E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$; (*ii*) since ϕ' is `false`, we have that there exists a $W' = (R',E')$ such that $W' \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$ and $(c',c'') \notin E'$. Due to (*i*), we easily derive that $(\{c,c'\}) \in \overline{q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c,c'\}) \in X$-$\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. Furthermore, due to (*ii*), we have that $(\{c,c',c''\}) \notin \overline{q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W')$ for at least one $W' \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c,c',c''\}) \notin X$-$\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. By construction, it follows that $\{c,c'\}$ is most informative in X-$\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$, i.e. $(\{c,c'\}) \in X$-$\mathsf{MIcertAns}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$.

Suppose now that (ϕ,ϕ') is a "no" instance of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, i.e. either ϕ is `false` or ϕ' is `true`. Assume first that ϕ is `false`. Using exactly the same consideration as in the lower bound proof for X-CERTANS of Theorem 3, we can immediately derive that there exists at least one $W = (R,E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$ and $(c,c') \notin E$. For such W, we clearly have that $(\{c,c'\}) \notin \overline{q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$, and therefore $(\{c,c'\}) \notin X$-$\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. It follows that $(\{c,c'\}) \notin X$-$\mathsf{MIcertAns}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. Assume now that ϕ is `true`, and thus also ϕ' is `true`. Using exactly the same consideration as in the lower bound proof for X-CERTANS of Theorem 3, we can immediately derive that both $(c,c') \in E$ and (c',c'') (and therefore, $(c,c'') \in E$ due to transitivity) hold for every $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. By construction, this means that $(\{c, c', c''\}) \in \overline{q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$ for every $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c, c', c''\}) \in X\text{-}\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$. Since $(\{c, c', c''\}) \in X\text{-}\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$ and $(\{c, c', c''\})$ is strictly more informative than $(\{c, c'\})$, we soon derive that $(\{c, c'\})$ is not most informative in X-$\mathsf{SetCert}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c, c'\}) \notin X\text{-}\mathsf{MIcertAns}(q^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MICERT,D/C}}_{\forall\exists\text{-NO}\forall\exists})$ also in this case. □

X-MIPOSSANS is $\text{BH}_3(2)$-complete for $X \in \{\text{MER}, \text{DEL}\}$.

Upper Bound: Due to the remark preceding Lemma 5, it is enough to show that, for both $X = \text{MER}$ and $X = \text{DEL}$, X-SETPOSSANS and X-NOBETTERPOSSANS are in Σ^p_2 and in Π^p_2 in data complexity, respectively.

As for X-SETPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, we now show how to check whether $\mathbf{C} \in X\text{-}\mathsf{MIpossAns}(q, D, \Sigma)$ in Σ^p_2 in the size of D. We first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{C} \in \overline{q}(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{C} \in X\text{-}\mathsf{SetPoss}(q, D, \Sigma)$ directly follows from the definition of the set $X\text{-}\mathsf{SetPoss}(q, D, \Sigma)$ of set X-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 2, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{C} \in X\text{-}\mathsf{SetPoss}(q, D, \Sigma)$ can be done in Σ^p_2 in the size of D for both $X = \text{MER}$ and $X = \text{DEL}$.

As for X-NOBETTERPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, for both $X = \text{MER}$ and $X = \text{DEL}$, we now show that the complement of X-NOBETTERPOSSANS is in Σ^p_2 in data complexity, i.e. we now show how to check in Σ^p_2 in the size of D whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$.

First, we simply guess an n-tuple $\mathbf{C}'$ of sets of constants and a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$, (*ii*) $\mathbf{C}' \in \overline{q}(D, W)$, and (*iii*) $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$. If conditions (*i*), (*ii*), and (*iii*) all hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking the complement of X-NOBETTERPOSSANS directly follows from the definition of the set X-$\mathsf{SetPoss}(q, D, \Sigma)$ of set X-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 2, condition (*i*) can be checked by means of a coNP-oracle in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, condition (*iii*) can be checked in polynomial time. So, overall, checking whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in Σ^p_2 in the size of D for both $X = \text{MER}$ and $X = \text{DEL}$.

Lower Bound for $X = \text{MER}$: The proof is by a LOGSPACE reduction from the ∀∃3CNF-NO∀∃3CNF problem, shown to be $\text{BH}_3(2)$-hard in Lemma 8.

We define the fixed schema $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$, DQ specification $\Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$, and CQ $q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$. We have $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists} = \{T/1,\ F/1,\ L/1,\ O'/2,\ O/3, O''/2,\ H/1,\ R'_{fff}/4,\ R'_{fft}/4,\ R'_{ftf}/4,\ R'_{ftt}/4,\ R'_{tff}/4,\ R'_{tft}/4,\ R'_{ttf}/4,\ R'_{ttt}/4,\ V'_Y/1,\ P'/4,\ T'_X/1,\ F'_X/1,\ R_{fft}/4,\ R_{ftf}/4,\ R_{ftt}/4,\ R_{tff}/4,\ R_{tft}/4,\ R_{ttf}/4,\ R_{ttt}/4,\ V_Y/1,\ P/4,\ T_X/1,\ F_X/1\}$. Informally, T and F store the constants t and f, respectively. The predicate H and L simply store the constant o_2 and the pair (o_1, o_2) of constants. The predicates O', O, and O'' simply store the pair (o'_1, o'_2), the triple (o_2, o_3, o_4), and the pair (o_5, o_6), respectively. Finally, the predicates R_I (resp. R'_I), for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, and the predicates P, V_Y, T_X, and F_X (resp. P', V'_Y, T'_X, and F'_X) are used to store the clauses of ϕ (resp. ϕ') exactly as done in the lower bound proof for MER-CERTANS of Theorem 3 and in the above lower bound proof for MER-MICERTANS.

The DQ specification $\Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists} = \langle \Gamma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, \Delta^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists} \rangle$ over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ is such that $\Gamma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ contains the following soft rules over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$:

- $\sigma_L = L(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_1 with constant o_2
- $\sigma_{O'} = O'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o'_1 with constant o'_2 in the presence of $O'(o_1, o_2)$
- $\sigma_O = \exists z. O(x, y, z) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_2 with constant o_3
- $\sigma'_O = \exists z. O(z, x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_3 with constant o_4
- $\sigma_{O''} = \exists z_{2,3}, z. O(z_{2,3}, z_{2,3}, z) \wedge O''(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, which simply allows the merge of constant o_5 with constant o_6 but only if constants o_2 and o_3 have been previously merged and $O''(o_5, o_6)$ is present

- ${\sigma'}^T_{Y'} = V'_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x,y)$, which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}'$ with the constant t
- ${\sigma'}^F_{Y'} = V'_Y(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x,y)$, which which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}'$ with the constant f
- For every $I \in \{fff,\ fft,\ ftf,\ ftt,\ tff,\ tft,\ ttf,\ ttt\}$, we have the soft rules:
  - ${\sigma'}^t_{I,1} = \exists c, v_1, v_2, v_3. P'(c, v_1, x, y) \wedge T'_X(v_1) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - ${\sigma'}^f_{I,1} = \exists c, v_1, v_2, v_3. P'(c, v_1, x, y) \wedge F'_X(v_1) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - ${\sigma'}^t_{I,2} = \exists c, v_1, v_2, v_3. P'(c, v_2, x, y) \wedge T'_X(v_2) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - ${\sigma'}^f_{I,2} = \exists c, v_1, v_2, v_3. P'(c, v_2, x, y) \wedge F'_X(v_2) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - ${\sigma'}^t_{I,3} = \exists c, v_1, v_2, v_3. P'(c, v_3, x, y) \wedge T'_X(v_3) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - ${\sigma'}^f_{I,3} = \exists c, v_1, v_2, v_3. P'(c, v_3, x, y) \wedge F'_X(v_3) \wedge R'_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$

  Informally, the above soft rules and the soft rules ${\sigma'}^T_{Y'}$ and ${\sigma'}^F_{Y'}$ are the same as in the lower bound proof for MER-CERTANS of Theorem 3 but defined for the clauses of ϕ'.
- $\sigma^T_Y = V_Y(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x,y)$, which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}$ with the constant t
- $\sigma^F_Y = V_Y(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x,y)$, which simply allows the merge of the (constants representing the) universally quantified variables $\mathbf{y}$ with the constant f
- For every $I \in \{fff,\ fft,\ ftf,\ ftt,\ tff,\ tft,\ ttf,\ ttt\}$, there are soft rules:
  - $\sigma^t_{I,1} = \exists c, v_1, v_2, v_3. P(c, v_1, x, y) \wedge T_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - $\sigma^f_{I,1} = \exists c, v_1, v_2, v_3. P(c, v_1, x, y) \wedge F_X(v_1) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - $\sigma^t_{I,2} = \exists c, v_1, v_2, v_3. P(c, v_2, x, y) \wedge T_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - $\sigma^f_{I,2} = \exists c, v_1, v_2, v_3. P(c, v_2, x, y) \wedge F_X(v_2) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - $\sigma^t_{I,3} = \exists c, v_1, v_2, v_3. P(c, v_3, x, y) \wedge T_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$
  - $\sigma^f_{I,3} = \exists c, v_1, v_2, v_3. P(c, v_3, x, y) \wedge F_X(v_3) \wedge R_I(c, v_1, v_2, v_3) \dashrightarrow \mathsf{EQ}(x,y)$

  Informally, the above soft rules and the soft rules σ^T_Y and σ^F_Y are the same as in the lower bound proof for MER-CERTANS of Theorem 3 for the clauses of ϕ.

Then, $\Delta^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ comprises the following denial constraints over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$:

- $\delta_{TF} = \neg(\exists y. T(y) \wedge F(y))$, which prevents the merge between the constants t and f. This means that every (constant representing a) universally quantified variable in $\mathbf{y}$ and in $\mathbf{y}'$ can be merged with either the constant t or the constant f, but not both
- $\delta_O = \neg(\exists y. O(y, y, y))$, which means that the merges between o_2 and o_3 and between o_3 and o_4 cannot occur at the same time, i.e. every solution will contain in the set of merges either (o_2, o_3) or (o_3, o_4) but not both
- $\delta_{O',H} = \neg(\exists y_1, y_2, y_3. O'(y_1, y_2) \wedge H(y_3))$, which means that every solution will contain in the set of removed facts either $O(o'_1, o'_2)$ or $H(o_2)$. Notice that this is similar to the denial constraint $\delta_{O,H}$ used in the lower bound proof for MER-POSSANS of Theorem 3
- $\delta_{O,O'',H} = \neg(\exists y_{2,3}, y_4, y_5, y_6. O(y_{2,3}, y_{2,3}, y_4) \wedge H(y_{2,3}) \wedge O''(y_5, y_6))$, which means that, if constants o_2 and o_3 have been merged, then either $H(o_2)$ or $O''(o_5, o_6)$ must occur in the set of removed facts
- We have the following denial constraints for the clauses of ϕ', which are similar to the ones used in the lower bound proof for MER-CERTANS of Theorem 3 and in the above lower bound proof for MER-MICERTANS:
  - ${\delta'}^0_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fff}(c, y_1, y_2, y_3) \wedge T'_X(y_1) \wedge T'_X(y_2) \wedge T'_X(y_3))$
  - ${\delta'}^1_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T'_X(y_2) \wedge T'_X(y_3))$
  - ${\delta'}^2_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T'_X(y_3))$
  - ${\delta'}^0_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fft}(c, y_1, y_2, y_3) \wedge T'_X(y_1) \wedge T'_X(y_2) \wedge F'_X(y_3))$
  - ${\delta'}^1_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T'_X(y_2) \wedge F'_X(y_3))$
  - ${\delta'}^2_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F'_X(y_3))$

- $\delta'^{0}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftf}(c, y_1, y_2, y_3) \wedge T'_X(y_1) \wedge F'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{1}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{2}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T'_X(y_3))$
- $\delta'^{0}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftt}(c, y_1, y_2, y_3) \wedge T'_X(y_1) \wedge F'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{1}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{2}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F'_X(y_3))$
- $\delta'^{0}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tff}(c, y_1, y_2, y_3) \wedge F'_X(y_1) \wedge T'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{1}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{2}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T'_X(y_3))$
- $\delta'^{0}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tft}(c, y_1, y_2, y_3) \wedge F'_X(y_1) \wedge T'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{1}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{2}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F'_X(y_3))$
- $\delta'^{0}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttf}(c, y_1, y_2, y_3) \wedge F'_X(y_1) \wedge F'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{1}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F'_X(y_2) \wedge T'_X(y_3))$
- $\delta'^{2}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T'_X(y_3))$
- $\delta'^{0}_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttt}(c, y_1, y_2, y_3) \wedge F'_X(y_1) \wedge F'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{1}_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F'_X(y_2) \wedge F'_X(y_3))$
- $\delta'^{2}_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O'(z_1, z_2) \wedge R'_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F'_X(y_3))$

• Finally, we have the following denial constraints for the clauses of ϕ, which are similar to the ones used in the lower bound proof for MER-CERTANS of Theorem 3 and in the above lower bound proof for MER-MICERTANS:
  - $\delta^{0}_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
  - $\delta^{1}_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
  - $\delta^{2}_{fff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fff}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge T_X(y_3))$
  - $\delta^{0}_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
  - $\delta^{1}_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
  - $\delta^{2}_{fft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{fft}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge T(y_2) \wedge F_X(y_3))$
  - $\delta^{0}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
  - $\delta^{1}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
  - $\delta^{2}_{ftf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftf}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge T_X(y_3))$
  - $\delta^{0}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
  - $\delta^{1}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
  - $\delta^{2}_{ftt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ftt}(c, y_1, y_2, y_3) \wedge T(y_1) \wedge F(y_2) \wedge F_X(y_3))$
  - $\delta^{0}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
  - $\delta^{1}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T_X(y_2) \wedge T_X(y_3))$
  - $\delta^{2}_{tff} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tff}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge T_X(y_3))$
  - $\delta^{0}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
  - $\delta^{1}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T_X(y_2) \wedge F_X(y_3))$
  - $\delta^{2}_{tft} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{tft}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F_X(y_3))$
  - $\delta^{0}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
  - $\delta^{1}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F_X(y_2) \wedge T_X(y_3))$
  - $\delta^{2}_{ttf} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttf}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge T_X(y_3))$
  - $\delta^{0}_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F_X(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$

– $\delta^1_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F_X(y_2) \wedge F_X(y_3))$
– $\delta^2_{ttt} = \neg(\exists z_1, z_2, c, y_1, y_2, y_3. O''(z_1, z_2) \wedge R_{ttt}(c, y_1, y_2, y_3) \wedge F(y_1) \wedge F(y_2) \wedge F_X(y_3))$

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ is $q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}(x) = H(x)$.

Given an instance (ϕ, ϕ') of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, we construct an $\mathcal{S}^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$-database $D_{(\phi,\phi')}$ as follows:

- The extension of the predicates R'_I and R_I, for I, and extension of the predicates P', P, T'_X, T_X, F'_X, F_X, V'_Y, and V_Y are exactly as in the $\mathcal{S}^{\text{MICERT,M}}_{\forall\exists\text{-NO}\forall\exists}$-database illustrated in the above lower bound proof for MER-MICERTANS;
- Furthermore, $D_{(\phi,\phi')}$ contains $T(t)$, $F(f)$, $L(o_1, o_2)$, $O'(o'_1, o'_2)$, $O(o_2, o_3, o_4)$, $O''(o_5, o_6)$, and $H(o_2)$.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input $\forall\exists$3CNF-NO$\forall\exists$3CNF instance (ϕ, ϕ'). To conclude the proof of the claimed lower bound, we now show that (ϕ, ϕ') is a "yes" instance of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{o_1, o_2\})$ is a most informative MER-possible answer to $q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}$.

Claim 10. *ϕ is `true` and ϕ' is `false` if and only if* $(\{o_1, o_2\}) \in$ MER-MIpossAns$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$.

Proof. First of all, we provide two crucial observations: (*i*) every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ must satisfy $(o_1, o_2) \in E$, where the merge between constant o_1 and constant o_2 can be activated by σ_L; (*ii*) due to δ_O, every $W = (R, E)$ such that $W \in \mathsf{Sol}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ cannot have both $(o_2, o_3) \in E$ and $(o_3, o_4) \in E$, where the former merge can be activated by σ_O and the latter by σ'_O. By construction of the soft rules and the denial constraints, this means that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ must satisfy either $(o_2, o_3) \in E$ or $(o_3, o_4) \in E$, but cannot satisfy both at the same time.

Suppose that (ϕ, ϕ') is a "no" instance of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, i.e. either ϕ is `false` or ϕ' is `true`. Assume first that ϕ' is `true`. In this case, using exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can immediately derive that $(o'_1, o'_2) \in E$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $O'(o'_1, o'_2) \notin R$ (otherwise it would not be possible to merge o'_1 with o'_2). Due to the denial constraint $\delta_{O',H}$, this also means that $H(o_2) \in R$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$. It follows that $\overline{q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W) = \emptyset$ for every $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2\}) \notin$ MER-MIpossAns$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$. Assume now that ϕ' is `false`, and thus also ϕ is `false`. Since both ϕ' and ϕ are `false`, using again exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can easily construct a $W = (R, E)$ such that (*i*) $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$, (*ii*) $O'(o'_1, o'_2) \in R$ and $O''(o_5, o_6) \in R$ (the former because ϕ' is `false` and the latter because ϕ is `false`), and therefore even with (*iii*) $H(o_2) \notin R$. More precisely, one can see that there exist two $W_1 = (R_1, E_1)$ and $W_2 = (R_2, E_2)$ satisfying (*i*), (*ii*), and (*iii*), one with $(o_2, o_3) \in E_1$ and the other with $(o_3, o_4) \in E_2$. For W_1 with $(o_2, o_3) \in E_1$, we clearly have that $(\{o_1, o_2, o_3\}) \in \overline{q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W_1)$ because $H(o_2) \notin R_1$, $(o_1, o_2) \in E_1$ (recall that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ must satisfy $(o_1, o_2) \in E$), and $(o_2, o_3) \in E_1$, which implies that o_1, o_2, and o_3 are in the same equivalence class in E_1. Thus, $(\{o_1, o_2, o_3\}) \in$ MER-SetPoss$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ because $(\{o_1, o_2, o_3\}) \in \overline{q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W_1)$ for $W_1 \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$. Since $(\{o_1, o_2, o_3\}) \in$ MER-SetPoss$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $(\{o_1, o_2, o_3\})$ is strictly more informative than $(\{o_1, o_2\})$, we soon derive that $(\{o_1, o_2\})$ cannot be most informative in MER-SetPoss$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{o_1, o_2\}) \notin$ MER-MIpossAns$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ also in this case.

Suppose now that ϕ is `true` and ϕ' is `false`. Since ϕ' is `false`, using exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can immediately derive that there exists at least one $W = (R, E)$ such that (*i*) $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ and (*ii*) $O'(o'_1, o'_2) \in R$. More precisely, one can see that there exist two $W_1 = (R_1, E_1)$ and $W_2 = (R_2, E_2)$ satisfying (*i*) and (*ii*), one with $(o_2, o_3) \in E_1$ and the other with $(o_3, o_4) \in E_2$. Consider W_2. Since $(o_3, o_4) \in E_2$, as already discussed above, we derive that $(o_2, o_3) \notin E_2$. By construction of the soft rules, this also implies that $(o_5, o_6) \notin E$ (note that the merge between constant o_5 and o_6 can be activated only by $\sigma_{O''}$ and only if o_2 and o_3) have been merged. In turn, this implies that the neither the denial constraint $\delta_{O',H}$ nor the denial constraint $\delta_{O,O'',H}$ can be violated even in the presence of $H(o_2)$ (the former because $O'(o'_1, o'_2) \in R_2$ and the latter because o_2 and o_3 have not been merged), and therefore $H(o_2) \notin R_2$ (otherwise, this would easily contradict the fact that $W_2 \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$). This means that $(\{o_1, o_2\}) \in \overline{q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W_2)$ because $H(o_2) \notin R_2$ and $(o_1, o_2) \in E_2$ (recall that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$ must satisfy $(o_1, o_2) \in E$), and therefore $(\{o_1, o_2\}) \in$ MER-SetPoss$(q^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,M}}_{\forall\exists\text{-NO}\forall\exists})$. We now show that, since ϕ is `true`, we have $(\{o_1, o_2, o_3\}) \notin$

MER-SetPoss$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$. Consider any $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$. As already discussed above, we have two possible cases: either $(o_3, o_4) \in E$ or $(o_2, o_3) \in E$. In the former case, we trivially have that o_3 is not in the same equivalence class of o_1 and o_2, and therefore $(\{o_1, o_2, o_3\}) \notin \overline{q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$. Consider now the latter case. One can see that, if ϕ is $\texttt{true}$, then, using again exactly the same consideration as in the lower bound proof for MER-CERTANS of Theorem 3, we can derive that every $W' = (R', E')$ such that $W' \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $(o_2, o_3) \in E'$ must satisfy $(o_5, o_6) \in E'$, where this latter merge can be activated by $\sigma_{O''}$. Since, by assumption, we know that $(o_2, o_3) \in E$ and ϕ is $\texttt{true}$, we derive that $(o_5, o_6) \in E$ as well. Due to the denial constraint $\delta_{O,O'',H}$, this also means that $H(o_2) \in R$, and therefore $(\{o_1, o_2, o_3\}) \notin \overline{q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$. So, since ϕ is $\texttt{true}$, we have derived that $(\{o_1, o_2, o_3\}) \notin \overline{q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ holds for every W with $W \in \mathsf{Sol}_{\text{MER}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$, which directly implies that $(\{o_1, o_2, o_3\}) \notin$ MER-SetPoss$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$. To conclude the proof, observe that, by construction, $(\{o_1, o_2\}) \in$ MER-SetPoss$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$ and $(\{o_1, o_2, o_3\}) \notin$ MER-SetPoss$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$ directly imply that $(\{o_1, o_2\})$ is most informative in MER-SetPoss$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$, i.e. $(\{o_1, o_2\}) \in$ MER-MIpossAns$(q^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,M}}_{\forall\exists\text{-NO}\forall\exists})$. □

Lower Bound for $X =$ DEL: The proof is again by a LOGSPACE reduction from the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem.

We define the fixed schema $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$, DQ specification $\Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$, and CQ $q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$ over $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$. We have $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists} = \{T/1,\ F/1,\ L/1,\ R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ T_Y/1,\ F_Y/1,\ FV_X/1,\ Prec_X/2,\ LV_X/1,\ C/2,\ C'/2,\ H/1,\ R'_{fff}/3,\ R'_{fft}/3,\ R'_{ftf}/3,\ R'_{ftt}/3,\ R'_{tff}/3,\ R'_{tft}/3,\ R'_{ttf}/3,\ R'_{ttt}/3,\ T'_Y/1,\ F'_Y/1,\ FV'_X/1,\ Prec'_X/2,\ LV'_X/1,\ G/2,\ G'/2,\ H'/1\}$. Informally, the predicates T and F store the constants t and f, respectively, while L stores both the constants t and f. Then, the predicates C, C', G, G', only stores the pairs (c_1, c_2), (c, c'), (c_3, c_4), and (c', c''), respectively. Furthermore, both the predicates H and H' only store the constant c'. Finally, the predicates R_I (resp. R'_I), for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, and the predicates T_Y, F_Y, FV_X, $Prec_X$, and LV_X (resp. T'_Y, F'_Y, FV'_X, $Prec'_X$, and LV'_X) are used to store the clauses of ϕ (resp. ϕ') exactly as done in the lower bound proof for DEL-POSSANS of Theorem 3.

Recall the DQ specification $\Sigma^{\text{poss,D}}_{\forall\exists\text{3CNF}} = \langle \Gamma^{\text{poss,D}}_{\forall\exists\text{3CNF}}, \Delta^{\text{poss,D}}_{\forall\exists\text{3CNF}} \rangle$ used in the lower bound proof for DEL-POSSANS of Theorem 3. The DQ specification $\Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists} = \langle \Gamma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, \Delta^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists} \rangle$ over $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$ is such that:

- $\Gamma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists} = \Gamma^{\text{poss,D}}_{\forall\exists\text{3CNF}} \cup \Gamma'$, where Γ' is obtained from $\Gamma^{\text{poss,D}}_{\forall\exists\text{3CNF}}$ by replacing every occurrence of the predicate name FV_X (resp. $Prec_X$, LV_X, C, and C') with the predicate name FV'_X (resp. $Prec'_X$, LV'_X, G, and G'). For example, since $\sigma_{Prec} \in \Gamma^{\text{poss,D}}_{\forall\exists\text{3CNF}}$, then $\sigma'_{Prec} = \exists z_p. L(z_p) \wedge Prec'_X(z_p, x) \wedge L(y) \dashrightarrow \mathsf{EQ}(x, y)$ occurs in Γ'. As another example, since $\sigma'_C \in \Gamma^{\text{poss,D}}_{\forall\exists\text{3CNF}}$, then the following soft rule occurs in Γ': $\exists z. G'(z, z) \wedge G(x, y) \dashrightarrow \mathsf{EQ}(x, y)$;
- $\Delta^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists} = \Delta^{\text{poss,D}}_{\forall\exists\text{3CNF}} \cup \Delta'$, where Δ' is obtained from $\Delta^{\text{poss,D}}_{\forall\exists\text{3CNF}}$ by replacing every occurrence of the predicate name C (resp. C', H, T_Y, F_Y, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name G (resp. G', H', T'_Y, F'_Y, R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, Δ' contains the denial constraints: $\delta'_G = \neg(\exists y_1, y_2. G(y_1, y_2) \wedge y_1 \neq y_2)$, $\delta'_{G'} = \neg(\exists y. G'(y, y) \wedge H'(y))$, and $\delta'_{Y'} = \neg(\exists y. T'_Y(y) \wedge F'_Y(y))$. As another example, since $\delta^1_{ftf} \in \Delta^{\text{poss,D}}_{\forall\exists\text{3CNF}}$, then $\delta'^1_{ftf} = \neg(\exists y_1, y_2, y_3. R'_{ftf}(y_1, y_2, y_3) \wedge T'_Y(y_1) \wedge F(y_2) \wedge T(y_3))$ occurs in Δ'.

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$ is $q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}(x) = G'(x, x)$.

Given an instance ϕ of the $\forall\exists$3CNF problem, recall the $\mathcal{S}^{\text{poss,D}}_{\forall\exists\text{3CNF}}$-database D_ϕ used in the lower bound proof for DEL-POSSANS of Theorem 3. Then, given an instance (ϕ, ϕ') of the $\forall\exists$3CNF-NO$\forall\exists$3CNF problem, where $\phi = \forall \mathbf{y}. \exists \mathbf{x}. c_1 \wedge \ldots \wedge c_k$ and $\phi' = \forall \mathbf{y}'. \exists \mathbf{x}'. c'_1 \wedge \ldots \wedge c'_{k'}$ with $\mathbf{y} = (y_1, \ldots, y_m)$, $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y}' = (y'_1, \ldots, y'_{m'})$, and $\mathbf{x}' = (x'_1, \ldots, x'_{n'})$, we construct an $\mathcal{S}^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$-database $D_{(\phi,\phi)} = D_\phi \cup D'_{\phi'}$, where $D'_{\phi'}$ represents ϕ' exactly as D_ϕ does for ϕ, i.e. $D'_{\phi'}$ is as follows:

- $D'_{\phi'}$ contains the facts $G(c_3, c_4)$, $G'(c', c'')$, and $H'(c')$;
- $D'_{\phi'}$ contains the fact $T'_Y(y'_i)$ and $F'_Y(y'_i)$ for each $i = 1, \ldots, m'$;
- $D'_{\phi'}$ contains the facts $FV'_X(x'_1)$, $LV'_X(x'_n)$, and the fact $Prec'_X(x'_i, x'_{i+1})$ for each $i = 1, n' - 1$;
- for each $i = 1, \ldots, k'$, if clause c'_i is of the form $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$ (resp. $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee v'_{i,3})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(v'_{i,1} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee v'_{i,2} \vee v'_{i,3})$), then $D'_{\phi'}$ contains the fact $R'_{fff}(v'_{i,1}, v'_{i,2}, v'_{i,3})$ (resp. $R'_{fft}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftf}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftt}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tff}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tft}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttf}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttt}(v'_{i,1}, v'_{i,2}, v'_{i,3})$), where $v'_{i,1}$ (resp. $v'_{i,2}$, $v'_{i,3}$) denotes the variable in $\mathbf{x}' \cup \mathbf{y}'$ of the first (resp. second, third) literal of clause c'_i.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input ∀∃3CNF-NO∀∃3CNF instance (ϕ, ϕ'). To conclude the proof of the claimed lower bound, we now show that (ϕ, ϕ') is a "yes" instance of the ∀∃3CNF-NO∀∃3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{c', c''\})$ is a most informative DEL-possible answer to $q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}$.

Claim 11. *ϕ is `true` and ϕ' is `false` if and only if* $(\{c', c''\}) \in \text{DEL-MIpossAns}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$.

Proof. Suppose that ϕ is `true` and ϕ' is `false`. Using exactly the same consideration as in the lower bound proof for DEL-POSSANS of Theorem 3, we can immediately derive the following: (*i*) since ϕ is `true`, we have that $(c, c') \notin E$ for every $W = (R, E)$ such that $W \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$; (*ii*) since ϕ' is `false`, we have that there exists a $W' = (R', E')$ such that $W' \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$ and $(c', c'') \in E'$. Due to (*i*), we easily derive that $(\{c, c', c''\}) \notin \overline{q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c, c', c''\}) \notin \text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$. Furthermore, due to (*ii*), we have that $(\{c', c''\}) \in \overline{q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W')$ for at least one $W' \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c', c''\}) \in \text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$. By construction, it follows that $(\{c', c''\})$ is most informative in $\text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, i.e. $(\{c', c''\}) \in \text{DEL-MIpossAns}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$.

Suppose now that (ϕ, ϕ') is a "no" instance of the ∀∃3CNF-NO∀∃3CNF problem, i.e. either ϕ is `false` or ϕ' is `true`. Assume first that ϕ' is `true`. Using exactly the same consideration as in the lower bound proof for DEL-POSSANS of Theorem 3, we can immediately derive that every $W = (R, E)$ such that $W \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$ satisfies $(c', c'') \notin E$. This clearly means that $(\{c', c''\}) \notin \overline{q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c', c''\}) \notin \text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$. It follows that $(\{c', c''\}) \notin \text{DEL-MIpossAns}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$. Assume now that ϕ' is `false`, and thus also ϕ is `false`. Using exactly the same consideration as in the lower bound proof for DEL-POSSANS of Theorem 3, we can immediately derive that there exists at least one $W = (R, E)$ such that (*i*) $W \in \text{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, (*ii*) $(c, c') \in E$, and (*iii*) $(c', c'') \in E$ (and therefore, $(c, c'') \in E$ due to transitivity). Point number (*ii*) because ϕ is `false`, whereas point number (*iii*) because ϕ' is `false`. For such W, we clearly have that $(\{c, c', c''\}) \in \overline{q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}}(D_{(\phi,\phi')}, W)$, and therefore $(\{c, c', c''\}) \in \text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$. Since $(\{c, c', c''\}) \in \text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$ and $(\{c, c', c''\})$ is strictly more informative than $(\{c', c''\})$, we soon derive that $(\{c', c''\})$ cannot be most informative in $\text{DEL-SetPoss}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$, and therefore $(\{c', c''\}) \notin \text{DEL-MIpossAns}(q^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIposs,D}}_{\forall\exists\text{-NO}\forall\exists})$ also in this case. □

PAR-MIPOSSANS is BH(2)-complete.

Upper Bound: Due to the remark preceding Lemma 5, it is enough to show that PAR-SETPOSSANS and PAR-NOBETTERPOSSANS are in NP and in coNP in data complexity, respectively.

As for PAR-SETPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, we now show how to check whether $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$ in NP in the size of D. We first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\text{dom}(D \setminus R)$. We then check (*i*) $W \in \text{Sol}(D, \Sigma)$ and (*ii*) $\mathbf{C} \in \overline{q}(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$ is guaranteed by Lemma 5 and the definition of the set $\text{PAR-SetPoss}(q, D, \Sigma)$ of set PAR-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 1, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, checking whether $\mathbf{C} \in \text{PAR-SetPoss}(q, D, \Sigma)$ can be done in NP in the size of D.

As for PAR-NOBETTERPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, we now show that the complement of PAR-NOBETTERPOSSANS is in NP in data complexity, i.e. we now show how to check in NP in the size of D whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in \text{PAR-SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$.

First, we simply guess an n-tuple $\mathbf{C}'$ of sets of constants and a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\text{dom}(D \setminus R)$. We then check (*i*) $W \in \text{Sol}(D, \Sigma)$, (*ii*) $\mathbf{C}' \in \overline{q}(D, W)$, and (*iii*) $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$. If conditions (*i*), (*ii*), and (*iii*) all hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking the complement of PAR-NOBETTERPOSSANS is guaranteed by Lemma 5 and the definition of the set $\text{PAR-SetPoss}(q, D, \Sigma)$ of set PAR-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, due to Theorem 1, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, condition (*iii*) can be checked in polynomial time. So, overall, checking whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in \text{PAR-SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in NP in the size of D.

Lower Bound: The proof is by a LOGSPACE reduction from the 3CNF-NO3CNF problem, shown to be BH(2)-hard in Lemma 7. Given an instance (ϕ, ϕ') of the 3CNF-NO3CNF problem, we let $\phi = \exists \mathbf{x}.c_1 \wedge \ldots \wedge c_m$ and $\phi' = \exists \mathbf{x}'.g_1 \wedge g_{m'}$, where $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}' = (x'_1, \ldots, x'_{n'})$.

We define the fixed schema $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, DQ specification $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, and CQ $q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$. We have $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists} = \{L/1,\ T/1,\ F/1,\ R_{fff}/4,\ R_{fft}/4,\ R_{ftf}/4,\ R_{ftt}/4,\ R_{tff}/4,\ R_{tft}/4,\ R_{ttf}/4,\ R_{ttt}/4,\ V_X/1,\ FV_C/1,\ Prec_C/2,\ C'/2,\ LV_{C'}/2,\ O/2,\ R'_{fff}/4,\ R'_{fft}/4,\ R'_{ftf}/4,\ R'_{ftt}/4,\ R'_{tff}/4,\ R'_{tft}/4,\ R'_{ttf}/4,\ R'_{ttt}/4,\ V'_X/1,\ FV_G/1,\ Prec_G/2,\ G'/2,\ LV_{G'}/2,\ O'/2\}$. Informally, the predicates T and F store the constants t and f, respectively, while L stores both the constants t and f. The predicate C' (resp. G') stores pairs of the form (c_i, c'_i) (resp. (g_i, g'_i)) for each $i = 1, \ldots, m$ (resp. $i = 1, \ldots, m'$). As usual, c_i (resp. g_i) is (the constant representing) the clause c_i (resp. g_i) of ϕ (resp. ϕ') while c'_i (resp. g'_i) is its copy. Furthermore, the predicates O and O' only store the pairs (o, o') and (o', o''), respectively. Finally, the predicates R_I (resp. R'_I), for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, and the predicates V_X, FV_C, $Prec_C$, and $LV_{C'}$ (resp. V'_X, FV_G, $Prec_G$, and $LV_{G'}$) are used to store the clauses of ϕ (resp. ϕ') exactly as done in the lower bound proof for PAR-POSSANS of Theorem 3.

Recall the DQ specification $\Sigma^{\text{POSS,C}}_{3\text{SAT}} = \langle \Gamma^{\text{POSS,C}}_{3\text{SAT}}, \Delta^{\text{POSS,C}}_{3\text{SAT}} \rangle$ used in the lower bound proof for PAR-POSSANS of Theorem 3. The DQ specification $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists} = \langle \Gamma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, \Delta^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists} \rangle$ over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ is such that:

- $\Gamma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists} = \Gamma^{\text{POSS,C}}_{3\text{SAT}} \cup \Gamma' \cup \{\sigma_O, \sigma_{O'}\}$, where $\sigma_O = \exists z.LV_{C'}(z,z) \wedge O(x,y) \dashrightarrow \mathsf{EQ}(x,y)$, $\sigma_{O'} = \exists z.LV_{G'}(z,z) \wedge O'(x,y) \dashrightarrow \mathsf{EQ}(x,y)$, and Γ' is obtained from $\Gamma^{\text{POSS,C}}_{3\text{SAT}}$ by replacing every occurrence of the predicate name V_X (resp. , FV_C, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}, C', $Prec_C$) with the predicate name V'_X (resp. , FV_G, R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}, G', $Prec_G$). For example, since $\sigma^F_X \in \Gamma^{\text{POSS,C}}_{3\text{SAT}}$, then ${\sigma'}^F_X = V'_X(x) \wedge F(y) \dashrightarrow \mathsf{EQ}(x,y)$ occurs in Γ'. As another example, since $\sigma^{Prec}_{ftf} \in \Gamma^{\text{POSS,C}}_{3\text{SAT}}$, then ${\sigma'}^{Prec}_{ftf} = \exists z_c, v_1, v_2, v_3.G'(z_c, z_c) \wedge Prec_G(z_c, x) \wedge R'_{ftf}(x, v_1, v_2, v_3) \wedge L(v_1) \wedge L(v_2) \wedge L(v_3) \wedge G'(x,y) \dashrightarrow \mathsf{EQ}(x,y)$ occurs in Γ'. Note that the soft rule σ_O (resp. $\sigma_{O'}$) allows the merge of constant o with constant o' (resp. constant o' with constant o'') but only if constants c_m and c'_m (resp. $g_{m'}$ and $g'_{m'}$) have been previously merged;
- $\Delta^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists} = \Delta^{\text{POSS,C}}_{3\text{SAT}} \cup \Delta'$, where Δ' is obtained from by replacing every occurrence of the predicate name R_{fff} (resp. R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name R'_{fff} (resp. R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, since $\delta_{tft} \in \Delta^{\text{POSS,C}}_{3\text{SAT}}$, then $\delta'_{tft} = \neg(\exists g, y_1, y_2, y_3.R'_{tft}(g, y_1, y_2, y_3) \wedge F(y_1) \wedge T(y_2) \wedge F(y_3))$ occurs in Δ'.

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ is $q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}(x) = O(x,x)$.

Given an instance ϕ of the 3SAT problem, recall the $\mathcal{S}^{\text{POSS,C}}_{3\text{SAT}}$-database D_ϕ used in the lower bound proof for PAR-POSSANS Theorem 3. Then, given an instance (ϕ, ϕ') of the 3CNF-NO3CNF problem, where $\phi = \exists \mathbf{x}.c_1 \wedge \ldots \wedge c_m$ and $\phi' = \exists \mathbf{x}'.g_1 \wedge \ldots g_{m'}$ with $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}' = (x'_1, \ldots, x'_{n'})$, we construct an $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$-database $D_{(\phi,\phi)} = D_\phi \cup D'_{\phi'} \cup \{O(o,o'), O'(o',o'')\}$, where $D'_{\phi'}$ represents ϕ' exactly as D_ϕ does for ϕ, i.e. $D'_{\phi'}$ is as follows:

- $D'_{\phi'}$ contains the facts $FV_G(g_1)$ and $LV_{G'}(g_m, g'_{m'})$;
- $D'_{\phi'}$ contains the fact $V'_X(x'_i)$ for each $i = 1, \ldots, n'$;
- $D'_{\phi'}$ contains the fact $Prec_G(g_i, g_{i+1})$ for each $i = 1, \ldots, m' - 1$;
- $D'_{\phi'}$ contains the fact $G'(g_i, g'_i)$ for each $i = 1, \ldots, m'$;
- finally, for each $i = 1, \ldots, m'$, if clause g_i is of the form $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$ (resp. $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee v'_{i,3})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(v'_{i,1} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee v'_{i,2} \vee v'_{i,3})$), then $D'_{\phi'}$ contains the fact $R'_{fff}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$ (resp. $R'_{fft}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftf}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftt}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tff}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tft}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttf}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttt}(g_i, v'_{i,1}, v'_{i,2}, v'_{i,3})$), where $v'_{i,1}$ (resp. $v'_{i,2}$, $v'_{i,3}$) denotes the variable in $\mathbf{x}'$ of the first (resp. second, third) literal of clause g_i.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input 3CNF-NO3CNF instance (ϕ, ϕ'). To conclude the proof of the claimed lower bound, we now show that (ϕ, ϕ') is a "yes" instance of the 3CNF-NO3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{o, o'\})$ is a most informative PAR-possible answer to $q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$.

Claim 12. *ϕ is `true` and ϕ' is `false` if and only if* $(\{o, o'\}) \in \text{PAR-MIpossAns}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$.

Proof. First of all note that, due to the soft rules σ_O and σ'_O and the fact that neither O nor O' are mentioned in the denial constraints, it is trivial to verify that the following holds for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$: $(o, o') \in E$ (resp. $(o', o'') \in E$) if and only if $(c_m, c'_m) \in E$ (resp. $(g_{m'}, g'_{m'}) \in E$).

Suppose that ϕ is `true` and ϕ' is `false`. Using exactly the same consideration as in the lower bound proof for PAR-POSSANS of Theorem 3, we can immediately derive the following: (*i*) since ϕ is `true`, we have that there exists a $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ and $(c_m, c'_m) \in E$; (*ii*) since ϕ' is `false`, we have that $(g_{m'}, g'_{m'}) \notin E$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. It follows that (*i*) there exists a $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ and $(o, o') \in E$; (*ii*) $(o', o'') \notin E$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. Due to (*ii*), we easily derive that $(\{o, o', o''\}) \notin \overline{q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o', o''\}) \notin$ PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. Furthermore, due to (*i*), we have that $(\{o, o'\}) \in \overline{q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$ for at least one $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o'\}) \in$ PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. By construction, it follows that $(\{o, o'\})$ is most informative in PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, i.e. $(\{o, o'\}) \in$ PAR-$\mathsf{MIpossAns}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$.

Suppose now that (ϕ, ϕ') is a "no" instance of the 3CNF-NO3CNF problem, i.e. either ϕ is `false` or ϕ' is `true`. Assume first that ϕ is `false`. Using exactly the same consideration as in the lower bound proof for PAR-POSSANS of Theorem 3, we can immediately derive that every $W = (R, E)$ such that $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ satisfies $(c_m, c'_m) \notin E$, and therefore also $(o, o') \notin E$. This clearly means that $(\{o, o'\}) \notin \overline{q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o'\}) \notin$ PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. It follows that $(\{o, o'\}) \notin$ PAR-$\mathsf{MIpossAns}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. Assume now that ϕ is `true`, and thus also ϕ' is `true`. Using exactly the same consideration as in the lower bound proof for PAR-POSSANS of Theorem 3, we can immediately derive that there exists at least one $W = (R, E)$ such that (*i*) $W \in \mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, (*ii*) $(c_m, c'_m) \in E$ (and thus $(o, o') \in E$ as well), and (*iii*) $(g_{m'}, g'_{m'}) \in E$ (and thus $(o', o'') \in E$ as well). Point number (*ii*) because ϕ is `true`, whereas point number (*iii*) because ϕ' is `true`. Due to transitivity, we have $(o, o'') \in E$. For such W, we clearly have that $(\{o, o', o''\}) \in \overline{q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$, and therefore $(\{o, o', o''\}) \in$ PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$. Since $(\{o, o', o''\}) \in$ PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ and $(\{o, o', o''\})$ is strictly more informative than $(\{o, o'\})$, we soon derive that $(\{o, o'\})$ cannot be most informative in PAR-$\mathsf{SetPoss}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o'\}) \notin$ PAR-$\mathsf{MIpossAns}(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ also in this case. □

Theorem 6. *For restricted DQ specifications, the decision problems* DEL-MICERTANS*,* PAR-MICERTANS*, and* DEL-MIPOSSANS *are DP-complete.*

Proof. The order we follow for proving the theorem for restricted DQ specifications is as follows: (*i*) we show that X-MICERTANS is BH(2)-complete for $X \in \{\text{DEL}, \text{PAR}\}$; and then (*ii*) we show that DEL-MIPOSSANS is BH(2)-complete.

For restricted DQ specifications, X-MICERTANS is BH(2)-complete for $X \in \{\text{DEL}, \text{PAR}\}$.

Upper Bound: Due to the remark preceding Lemma 5, it is enough to show that, for both $X = \text{DEL}$ and $X = \text{PAR}$, X-SETCERTANS and X-NOBETTERCERTANS are in coNP and in NP in data complexity, respectively.

As for X-SETCERTANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, for both $X = \text{DEL}$ and $X = \text{PAR}$, we now show how to check whether $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ in NP in the size of D, thus obtaining that X-SETCERTANS is in coNP in data complexity. We first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{C} \notin \overline{q}(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return `true`; otherwise, we return `false`. Correctness of the above procedure for checking $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ directly follows from the definition of the set $X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ of set X-certain answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, as shown in the upper bound for X-OPTREC for restricted DQ specifications of Theorem 4, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, for restricted DQ specifications checking whether $\mathbf{C} \notin X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ can be done in NP in the size of D for both $X = \text{DEL}$ and $X = \text{PAR}$.

As for X-NOBETTERCERTANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, for both $X = \text{DEL}$ and $X = \text{PAR}$, we need to show that checking whether there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in NP in the size of D. Let $\mathbf{C} = (C_1, \ldots, C_n)$ and recall the notion of minimal more informative extension of $\mathbf{C}$ introduced in the upper bound proof for X-MICERTANS for general DQ specifications of Theorem 5.

For each possible pair $p = (j, c)$ of a natural number $j \in [1, n]$ and a constant $c \in \mathsf{dom}(D)$ such that $c \notin C_j$ (we recall that the number of such p is at most $m * n$, where m is the cardinality of the set $\mathsf{dom}(D)$), we guess a pair $W_p = (R_p, E_p)$, where $R_p \subseteq D$ and E_p is an equivalence relation over $\mathsf{dom}(D \setminus R_p)$. We then check whether both (*i*) $W_p \in \mathsf{Sol}_X(D, \Sigma)$ and (*ii*) $\mathbf{C}_p \notin \overline{q}(D, W_p)$ hold (and therefore $\mathbf{C}_p \notin X\text{-}\mathsf{SetCert}(q, D, W_p)$). If *each* pair p as above satisfies both conditions (*i*) and (*ii*), then we return `true`; otherwise, we return `false`. Correctness of the above procedure, i.e. the fact that returns `true` if and only if there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, is guaranteed by the same property observed in the upper bound for X-MICERTANS for general DQ specifications of Theorem 5, namely: if there exists a tuple $\mathbf{C}'$ of sets of constants such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$, then there must exist a tuple $\mathbf{C}_p$ of sets of constants such that $\mathbf{C}_p$ is a minimal more informative extension of $\mathbf{C}$ for which $\mathbf{C}_p \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$. As for its running time, we observe that each W_p is polynomially related to D. Furthermore, as shown in the upper bound for X-OPTREC for restricted DQ specifications of Theorem 4, for each p as above, condition (*i*) can be checked in polynomial time in the size of D and W_p (and therefore, in the size of D as well because W_p is polynomially related to D). Finally, due to Lemma 6, for each p as above, condition (*ii*) can be checked in polynomial time in the size of D and W_p (and therefore, in the size of D as well because W_p is polynomially related to D). So, overall, for restricted DQ specifications checking whether there exists no $\mathbf{C}'$ such that $\mathbf{C}' \in X\text{-}\mathsf{SetCert}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in NP in the size of D for both $X = $ DEL and $X = $ PAR.

Lower Bound: The proof is by a LOGSPACE reduction from the 3CNF-NO3CNF problem.

We define the fixed schema $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$, restricted DQ specification $\Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$ over $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$, and CQ $q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$ over $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$. We have $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists} = \{F/1,\ T/1,\ R_{fff}/3,\ R_{fft}/3,\ R_{ftf}/3,\ R_{ftt}/3,\ R_{tff}/3,\ R_{tft}/3,\ R_{ttf}/3,\ R_{ttt}/3,\ V/1,\ O/2,\ R'_{fff}/3,\ R'_{fft}/3,\ R'_{ftf}/3,\ R'_{ftt}/3,\ R'_{tff}/3,\ R'_{tft}/3,\ R'_{ttf}/3,\ R'_{ttt}/3,\ V'/1,\ O'/2\}$. Informally, T and F store the constants t and f. The predicates O and O' store the pairs (o, o') and (o', o''), respectively. The predicates V and V' store (the constants representing) the variables $\mathbf{x}$ of ϕ and the variables $\mathbf{x}'$ of ϕ'. Finally, as usual, the predicates R_I and R'_I, for $I \in \{fff, fft, ftf, ftt, tff, tft, ttf, ttt\}$, are used to store the clauses of ϕ and the clauses of ϕ', respectively. Note that the predicates $R_{fff},\ R_{fft},\ R_{ftf},\ R_{ftt},\ R_{tff},\ R_{tft},\ R_{ttf},\ R_{ttt},\ V/1,\ O/2$ play exactly the same role as in the lower bound proof for the restricted DQ specification case of X-CERTANS ($X \in \{\text{DEL}, \text{PAR}\}$) for representing ϕ, while the predicates $R'_{fff},\ R'_{fft},\ R'_{ftf},\ R'_{ftt},\ R'_{tff},\ R'_{tft},\ R'_{ttf},\ R'_{ttt},\ V'/1,\ O'/2$ do the same for representing ϕ'.

Recall the DQ specification $\Sigma^{\text{RESTR,D/C}}_{\text{3SAT}} = \langle \Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}, \Delta^{\text{RESTR,D/C}}_{\text{3SAT}} \rangle$ used in the lower bound proof for the restricted specification case of X-CERTANS ($X \in \{\text{DEL}, \text{PAR}\}$). The DQ specification $\Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists} = \langle \Gamma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, \Delta^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists} \rangle$ over $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$ is such that:

- $\Gamma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists} = \Gamma^{\text{RESTR,D/C}}_{\text{3SAT}} \cup \Gamma'$, where Γ' is obtained from $\Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}$ by replacing every occurrence of the predicate name V (resp. O, R_{fff}, R_{fft}, R_{ftf}, R_{ftt}, R_{tff}, R_{tft}, R_{ttf}, R_{ttt}) with the predicate name V' (resp. O', R'_{fff}, R'_{fft}, R'_{ftf}, R'_{ftt}, R'_{tff}, R'_{tft}, R'_{ttf}, R'_{ttt}). For example, since $\sigma^T_V \in \Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}$, then ${\sigma'}^T_V = V'(x) \wedge T(y) \dashrightarrow \mathsf{EQ}(x, y)$ occurs in Γ'. As another example, since $\sigma_{ftf} \in \Gamma^{\text{RESTR,D/C}}_{\text{3SAT}}$, then $\sigma'_{ftf} = \exists u_1, u_2, u_3.R'_{ftf}(u_1, u_2, u_3) \wedge T(u_1) \wedge F(u_2) \wedge T(u_3) \wedge O'(x, y) \dashrightarrow \mathsf{EQ}(x, y)$ occurs in Γ';
- $\Delta^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists} = \Delta^{\text{RESTR,D/C}}_{\text{3SAT}} = \{\neg(\exists y.T(y) \wedge F(y))\}$.

Finally, the fixed unary CQ over $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$ is $q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}(x) = O'(x, x)$.

Given an instance ϕ of the 3SAT problem, recall the $\mathcal{S}^{\text{RESTR,D/C}}_{\text{3SAT}}$-database D_ϕ used in the lower bound proof for the restricted DQ specification case of X-CERTANS ($X \in \{\text{DEL}, \text{PAR}\}$). Then, given an instance (ϕ, ϕ') of the 3CNF-NO3CNF problem, where $\phi = \exists \mathbf{x}.c_1 \wedge \ldots \wedge c_m$ and $\phi' = \exists \mathbf{x}'.c'_1 \wedge \ldots \wedge c'_{m'}$ with $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{x}' = (x'_1, \ldots, x'_{n'})$, we construct an $\mathcal{S}^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$-database $D_{(\phi,\phi)} = D_\phi \cup D'_{\phi'}$, where $D'_{\phi'}$ represents ϕ' exactly as D_ϕ does for ϕ, i.e. $D'_{\phi'}$ is as follows:

- $D'_{\phi'}$ contains the fact $O'(o', o'')$;
- $D'_{\phi'}$ contains the fact $V'(x'_i)$ for each $i = 1, \ldots, n'$;
- Finally, for each $i = 1, \ldots, m'$, if clause c'_i is of the form $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$ (resp. $(\overline{v'_{i,1}} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(\overline{v'_{i,1}} \vee v'_{i,2} \vee v'_{i,3})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee \overline{v'_{i,2}} \vee v'_{i,3})$, $(v'_{i,1} \vee v'_{i,2} \vee \overline{v'_{i,3}})$, $(v'_{i,1} \vee v'_{i,2} \vee v'_{i,3})$), then $D'_{\phi'}$ contains the fact $R'_{fff}(v'_{i,1}, v'_{i,2}, v'_{i,3})$ (resp. $R'_{fft}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftf}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ftt}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tff}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{tft}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttf}(v'_{i,1}, v'_{i,2}, v'_{i,3})$, $R'_{ttt}(v'_{i,1}, v'_{i,2}, v'_{i,3})$), where $v'_{i,1}$ (resp. $v'_{i,2}$, $v'_{i,3}$) denotes the variable in $\mathbf{x}'$ of the first (resp. second, third) literal of clause c'_i.

It is immediate to verify that $D_{(\phi,\phi')}$ can be constructed in LOGSPACE from an input 3CNF-NO3CNF instance (ϕ, ϕ'). To conclude the proof of the claimed lower bound, we now show that, for both $X = $ DEL and $X = $ PAR, (ϕ, ϕ') is a "yes" instance of the 3CNF-NO3CNF problem (i.e. ϕ is `true` and ϕ' is `false`) if and only if $(\{o', o''\})$ is a most informative X-certain answer to $q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$ on $D_{(\phi,\phi')}$ w.r.t. $\Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}$.

Claim 13. *For both $X = $ DEL and $X = $ PAR, ϕ is `true` and ϕ' is `false` if and only if $(\{o', o''\}) \in X$-$\mathsf{MIcertAns}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$.*

Proof. Suppose that ϕ is $\texttt{true}$ and ϕ' is $\texttt{false}$. Using exactly the same consideration as in the lower bound proof for the restricted specification case of X-CERTANS, we can immediately derive the following: (*i*) since ϕ' is $\texttt{false}$, we have that $(o', o'') \in E$ for every $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$; (*ii*) since ϕ is $\texttt{true}$, we have that there exists a $W' = (R', E')$ such that $W' \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$ and $(o, o') \notin E'$. Due to (*i*), we easily derive that $(\{o', o''\}) \in \overline{q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$ for every $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o', o''\}) \in X$-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. Furthermore, due to (*ii*), we have that $(\{o, o', o''\}) \notin \overline{q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W')$ for at least one $W' \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o', o''\}) \notin X$-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. By construction, it follows that $(\{o', o''\})$ is most informative in X-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$, i.e. $(\{o', o''\}) \in X$-$\mathsf{MIcertAns}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$.

Suppose now that (ϕ, ϕ') is a "no" instance of the 3CNF-NO3CNF problem, i.e. either ϕ is $\texttt{false}$ or ϕ' is $\texttt{true}$. Assume first that ϕ' is $\texttt{true}$. Using exactly the same consideration as in the lower bound proof for the restricted specification case of X-CERTANS, we can immediately derive that there exists at least one $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$ and $(o', o'') \notin E$. For such W, we clearly have that $(\{o', o''\}) \notin \overline{q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$, and therefore $(\{o', o''\}) \notin X$-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. It follows that $(\{o', o''\}) \notin X$-$\mathsf{MIcertAns}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. Assume now that ϕ' is $\texttt{false}$, and thus also ϕ is $\texttt{false}$. Using exactly the same consideration as in the lower bound proof for the restricted specification case of X-CERTANS, we can immediately derive that both $(o, o') \in E$ and $(o', o'') \in E$ (and therefore, $(o_1, o_3) \in E$ due to transitivity) hold for every $W = (R, E)$ such that $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. By construction, this means that $(\{o, o', o''\}) \in \overline{q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}}(D_{(\phi,\phi')}, W)$ holds for every $W \in \mathsf{Sol}_X(D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o, o', o''\}) \in X$-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$. Since $(\{o, o', o''\}) \in X$-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$ and $(\{o, o', o''\})$ is strictly more informative than $(\{o', o''\})$, we soon derive that $(\{o', o''\})$ cannot be most informative in X-$\mathsf{SetCert}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$, and therefore $(\{o', o''\}) \notin X$-$\mathsf{MIcertAns}(q^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIRE,D/C}}_{\exists\text{-NO}\exists})$ also in this case. □

For restricted DQ specifications, DEL-MIPOSSANS is BH(2)-complete.

Upper Bound: Due to the remark preceding Lemma 5, it is enough to show that DEL-SETPOSSANS and DEL-NOBETTERPOSSANS are in NP and in coNP in data complexity, respectively.

As for DEL-SETPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, we now show how to check whether $\mathbf{C} \in$ DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ in NP in the size of D. We first guess a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$ and (*ii*) $\mathbf{C} \in \overline{q}(D, W)$. If both conditions (*i*) and (*ii*) hold, then we return $\texttt{true}$; otherwise, we return $\texttt{false}$. Correctness of the above procedure for checking $\mathbf{C} \in$ DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ directly follows from the definition of the set DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ of set DEL-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, as shown in the upper bound for DEL-OPTREC for restricted DQ specifications of Theorem 4, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). So, overall, for restricted DQ specifications checking whether $\mathbf{C} \in$ DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ can be done in NP in the size of D.

As for DEL-NOBETTERPOSSANS, given a DQ specification Σ over a schema $\mathcal{S}$, an $\mathcal{S}$-database D, a CQ q over $\mathcal{S}$ of arity n, and an n-tuple $\mathbf{C}$ of sets of constants, we now show that the complement of DEL-NOBETTERPOSSANS is in NP in data complexity, i.e. we now show how to check in NP in the size of D whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in$ DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$.

First, we simply guess an n-tuple $\mathbf{C}'$ of sets of constants and a pair $W = (R, E)$, where $R \subseteq D$ and E is an equivalence relation over $\mathsf{dom}(D \setminus R)$. We then check (*i*) $W \in \mathsf{Sol}_{\text{DEL}}(D, \Sigma)$, (*ii*) $\mathbf{C}' \in \overline{q}(D, W)$, and (*iii*) $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$. If conditions (*i*), (*ii*), and (*iii*) all hold, then we return $\texttt{true}$; otherwise, we return $\texttt{false}$. Correctness of the above procedure for checking the complement of DEL-NOBETTERPOSSANS directly follows from the definition of the set DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ of set DEL-possible answers to q on D w.r.t. Σ. As for its running time, we observe that W is polynomially related to D. Furthermore, as shown in the upper bound for X-OPTREC for restricted DQ specifications of Theorem 4, condition (*i*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Due to Lemma 6, condition (*ii*) can be checked in polynomial time in the size of D and W (and therefore, in the size of D as well because W is polynomially related to D). Finally, condition (*iii*) can be checked in polynomial time. So, overall, for restricted DQ specifications checking whether there exists a $\mathbf{C}'$ such that $\mathbf{C}' \in$ DEL-$\mathsf{SetPoss}(q, D, \Sigma)$ and $\mathbf{C}'$ is strictly more informative than $\mathbf{C}$ can be done in NP in the size of D.

Lower Bound: We can adopt exactly the same LOGSPACE reduction from the 3CNF-NO3CNF problem used in the lower bound proof for PAR-MIPOSSANS of Theorem 5. Specifically, recall the fixed schema $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, DQ specification $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$

over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, unary CQ $q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}(x)$ over $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, and unary tuple $(\{o, o'\})$ used in that proof. Note that $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$ is a restricted DQ specification. Furthermore, given an instance (ϕ, ϕ') of the 3CNF-NO3CNF problem, recall the $\mathcal{S}^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$-database $D_{(\phi,\phi')}$ used in that proof.

By construction of $\Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}$, it is immediate to verify that $\mathsf{Sol}_{\text{PAR}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}) = \mathsf{Sol}_{\text{DEL}}(D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ holds for any 3CNF-NO3CNF instance (ϕ, ϕ'). This clearly implies that PAR-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$=DEL-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ holds for any 3CNF-NO3CNF instance (ϕ, ϕ'). Furthermore, since Claim 12 shows that (ϕ, ϕ') is a "yes" instance of the 3CNF-NO3CNF problem if and only if $(\{o, o'\}) \in$ PAR-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, and since PAR-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$=DEL-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$ holds for any 3CNF-NO3CNF instance (ϕ, ϕ'), we derive that (ϕ, ϕ') is a "yes" instance of the 3CNF-NO3CNF problem if and only if $(\{o, o'\}) \in$ DEL-MIpossAns$(q^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists}, D_{(\phi,\phi')}, \Sigma^{\text{MIPOSS,C}}_{\exists\text{-NO}\exists})$, thus obtaining the claimed lower bound. $\square$