The Impact of Disjunction on Query Answering Under Guarded-based Existential Rules

Pierre Bourhis, Michael Morak, and Andreas Pieris

Department of Computer Science, University of Oxford, UK
firstname.lastname@cs.ox.ac.uk

Abstract. We give the complete picture of the complexity of conjunctive query answering under (weakly-)(frontier-)guarded disjunctive existential rules, i.e., existential rules extended with disjunction, and their main subclasses, linear rules and inclusion dependencies.

1 Introduction

Rule-based languages have a prominent presence in the areas of AI and databases. A noticeable formalism, originally intended for expressing complex queries over relational databases, is Datalog, i.e., function-free first-order Horn logic. Strong interest in enhancing Datalog with existential quantification in rule-heads emerged in recent years, see, e.g., [1–5]. This interest stems from the inability of plain Datalog to infer the existence of new objects which are not already in the extensional database [6]. The obtained rules are known under a variety of names such as *existential rules*, *tuple-generating dependencies (TGDs)*, and *Datalog$^{\pm}$ rules*. Unfortunately, the addition of existential quantification as above easily leads to undecidability of the main reasoning tasks, and in particular of conjunctive query answering [7]. Therefore, several concrete languages which guarantee decidability have been proposed, see, e.g., [1, 3, 5, 8–11]. Nevertheless, TGDs are not powerful enough for nondeterministic reasoning. For example, a simple and natural statement like "a child is a boy *or* a girl" cannot be expressed using TGDs; however, it can be easily expressed using the disjunctive rule $child(X) \rightarrow boy(X) \vee girl(X)$.

Obviously, to be able to represent such kind of disjunctive knowledge, we need to enrich the existing classes of TGDs with disjunction in the head, or, equivalently, to consider *disjunctive TGDs (DTGDs)* [12]. Such an extension of plain Datalog (a.k.a. *full TGDs*), called *disjunctive Datalog*, has been studied in [13]. More recently, special cases of the problem of query answering under guarded-based DTGDs have been investigated [14, 15]. However, the picture of the computational complexity of the problem is still foggy, and there are several challenging issues to be tackled.

Our main goal is to better understand the impact of disjunction on query answering under the main guarded-based classes of TGDs, and how existing complexity results for TGDs are affected by adding disjunction. Notice that guardedness is a well-accepted paradigm, giving rise to robust languages that capture important lightweight description logics such as DL-Lite [16] and $\mathcal{EL}$ [17]. In the present work, we concentrate on the following fundamental questions: what is the exact complexity of conjunctive query

(CQ) answering under (weakly-)(frontier-)guarded DTGDs [1, 9], and their main subclasses, i.e., linear DTGDs and disjunctive inclusion dependencies (DIDs) [5]? How is it affected if we consider a signature of bounded arity, or a fixed set of dependencies? Moreover, how is it affected if we pose the queries in a more expressive query language, in particular using unions of CQs (UCQs)? As we shall see, the addition of disjunction has a significant effect on the complexity of CQ answering. We show an unexpectedly strong lower bound, which is critical towards the closing of the above issues.

Our contributions can be summarized as follows:

1. We show that CQ answering for (weakly-)(frontier-) guarded DTGDs is 2EXPTIME-complete in the combined complexity; this also holds for UCQs. Regarding the data complexity, we show that under frontier-guarded DTGDs it is coNP-complete, while for weakly-frontier-guarded it is EXPTIME-complete. The upper bounds are obtained by exploiting results on expressive languages such as guarded negation first-order logic [18], while the lower bounds are inherited from existing results.
2. We show that CQ answering under a *fixed* set of DIDs is 2EXPTIME-hard, even if restricted to predicates of arity at most three. In case of UCQs, the above result holds even for unary and binary predicates. These strong lower bounds are established by a reduction from an appropriate variant of the validity problem of CQs w.r.t. a Büchi automaton [19]. Together with the 2EXPTIME upper bound discussed above, this gives us the complete picture for the complexity of our problem.
3. We investigate a natural fragment of DIDs with lower combined complexity. In fact, we consider frontier-one dependencies (i.e., only one variable is propagated from the body to the head), and we show that the combined complexity decreases to EXPTIME-complete.
4. We show that frontier-guarded DTGDs, combined with negative constraints, are strictly more expressive than *DL-Lite*$^{\mathcal{H}}_{bool}$ [20], one of the most expressive languages of the DL-Lite family. This allows us to show that query answering under *DL-Lite*$^{\mathcal{H}}_{bool}$ is in 2EXPTIME in combined complexity. The matching lower bound holds since our complexity results on DIDs imply that, for every description logic equipped with limited existential quantification, role inverse and union, query answering is 2EXPTIME-hard.

A technical report containing the full proofs is available at http://www.cs.ox.ac.uk/people/michael.morak/pubs/DL2013-techrep.pdf.

2 Preliminaries

Technical Definitions. We define the following pairwise disjoint (infinite) sets: a set Γ of *constants*, a set Γ_N of *labeled nulls*, and a set Γ_V of regular *variables*. We denote by $\mathbf{X}$ sequences (or sets) of variables $X_1, \ldots, X_k$. A *relational schema* $\mathcal{R}$ is a set of *relational symbols* (or *predicates*). A position $r[i]$ in $\mathcal{R}$ is identified by $r \in \mathcal{R}$ and its i-th argument. A *term* t is a constant, null, or variable. An *atom* has the form $r(t_1, \ldots, t_n)$, where r is a relation, and $t_1, \ldots, t_n$ are terms. For an atom $\underline{a}$, we denote $terms(\underline{a})$ and $var(\underline{a})$ the set of its terms and the set of its variables, respectively; these extend to sets of atoms. Conjunctions and disjunctions of atoms are often identified with the sets of their atoms. An *instance* I for a schema $\mathcal{R}$ is a (possibly infinite) set of atoms $r(\mathbf{t})$, where $r \in \mathcal{R}$ and $\mathbf{t}$ is a tuple of constants and nulls. A *database* D is a finite instance such that $terms(D) \subset \Gamma$. We assume the reader is familiar with *(unions of) conjunctive queries ((U)CQs)*. The answer to a (U)CQ q over an instance I is denoted $q(I)$. A Boolean (U)CQ q has positive answer over I, denoted $I \models q$, if $\langle\rangle \in q(I)$.

Disjunctive Tuple-generating Dependencies. A *disjunctive tuple-generating dependency (DTGD)* σ over a schema $\mathcal{R}$ is a first-order formula of the form $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{Y}\, \psi_i(\mathbf{X}, \mathbf{Y}_i)$, where $n \geqslant 1$, $\mathbf{X} \cup \mathbf{Y} \subset \Gamma_V$, and $\varphi, \psi_1, \ldots, \psi_n$ are conjunctions of atoms over $\mathcal{R}$; φ is the *body* of σ, denoted $body(\sigma)$, while $\bigvee_{i=1}^{n} \psi_i$ is the *head* of σ, denoted $head(\sigma)$. If $n = 1$, then σ is called *tuple-generating dependency (TGD)*. For brevity, we will omit the universal quantifiers in front of DTGDs. An instance I satisfies σ, written $I \models \sigma$, if whenever there exists a homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$, then there exists $i \in \{1, \ldots, n\}$ and $h' \supseteq h$ such that $h'(\psi_i(\mathbf{X}, \mathbf{Y}_i)) \subseteq I$; I satisfies a set Σ of DTGDs, denoted $I \models \Sigma$, if I satisfies each $\sigma \in \Sigma$.

A DTGD σ is *guarded* if there exists an atom $\underline{a} \in body(\sigma)$, called *guard*, which contains all the variables occurring in $body(\sigma)$. *Weakly-guarded* DTGDs extend guarded DTGDs by requiring only the body-variables that appear at *affected positions*, i.e., positions at which a null value may appear during the disjunctive chase (see below) to appear in the guard; for the formal definition see [9]. The concept of frontier can be used to generalize (weakly-)guarded DTGDs. The *frontier* of a DTGD σ is the set of variables $var(body(\sigma)) \cap var(head(\sigma))$. σ is *frontier-guarded* if there exists an atom $\underline{a} \in body(\sigma)$ which contains all the variables occurring in its frontier. The class of *weakly-frontier-guarded* DTGDs is defined analogously. A DTGD σ is *linear* if it has only one body-atom. *Disjunctive inclusion dependencies (DIDs)* are obtained by restricting linear DTGDs as follows: the head is a disjunction of atoms (and not of conjunctions), and there are no repeated variables in the body or in the head.

Query Answering. The *models* of D and Σ, denoted $mods(D, \Sigma)$, is the set of instances $\{I \mid I \supseteq D \text{ and } I \models \Sigma\}$. The *answer* to a CQ q w.r.t. D and Σ, denoted $ans(q, D, \Sigma)$, is the set of tuples of constants $\bigcap_{I \in mods(D,\Sigma)} \{\mathbf{t} \mid \mathbf{t} \in q(I)\}$. The answer to a Boolean CQ q w.r.t. D and Σ is *positive*, denoted $D \cup \Sigma \models q$, if $\langle\rangle \in ans(q, D, \Sigma)$. The answer to a UCQ w.r.t. D and Σ is defined analogously. The problem, called CQAns, tackled in this work is defined as follows: given a CQ q, a database D, a set Σ of DTGDs, and a tuple of constants $\mathbf{t}$, decide whether $\mathbf{t} \in ans(q, D, \Sigma)$. The problem UCQAns is defined analogously. Notice that (U)CQAns for arbitrary CQs can be easily reduced to (U)CQAns for Boolean CQs, just by substituting the given tuple $\mathbf{t}$ into the CQs; thus, we focus on Boolean CQs. The *data complexity* of the above problems is calculated taking only the database as input. For the *combined complexity*, the query and set of DTGDs count as input as well.

Disjunctive Chase. We employ the *disjunctive chase* introduced in [12]. Consider an instance I, and a DTGD $\sigma : \varphi(\mathbf{X}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{Y}\, \psi_i(\mathbf{X}, \mathbf{Y})$. We say that σ is *applicable* to I if there exists a homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$, and the result of applying σ to I with h is the set $\{I_1, \ldots, I_n\}$, where $I_i = I \cup h'(\psi_i(\mathbf{X}, \mathbf{Y}))$, for each $i \in \{1, \ldots, n\}$, and $h' \supseteq h$ is such that $h'(Y)$ is a "fresh" null not occurring in I, for each $Y \in \mathbf{Y}$. For such an application of a DTGD, which defines a single DTGD *chase step*, we write $I\langle\sigma, h\rangle\{I_1, \ldots, I_n\}$. A *disjunctive chase tree* of a database D and

	Combined complexity	**Bounded arity**	**Fixed rules**	**Data complexity**
L/ID	ExpTime-hard	Π_{P}^{2}-hard	Π_{P}^{2}-hard	**coNP-complete**
G	2ExpTime-hard	ExpTime-hard	Π_{P}^{2}-hard	**coNP-complete**
W-G	2ExpTime-hard	ExpTime-hard	ExpTime-hard	**ExpTime-complete**
F-G	2ExpTime-hard	2ExpTime-hard	Π_{P}^{2}-hard	coNP-hard
W-F-G	2ExpTime-hard	2ExpTime-hard	ExpTime-hard	ExpTime-hard

Table 1. Known complexity results for (U)CQAns.

a set Σ of DTGDs is a (possibly infinite) tree such that the root is D, and for every node I, assuming that $\{I_1, \ldots, I_n\}$ are the children of I, there exists $\sigma \in \Sigma$ and a homomorphism h such that $I\langle\sigma, h\rangle\{I_1, \ldots, I_n\}$. The disjunctive chase algorithm for D and Σ consists of an exhaustive application of DTGD chase steps in a fair fashion, which leads to a disjunctive chase tree T of D and Σ; we denote by $chase(D, \Sigma)$ the set $\{I \mid I \text{ is a leaf of } T\}$. Note that each leaf of T is well-defined as the least fixpoint of a monotonic operator. By construction, each instance of $chase(D, \Sigma)$ is a model of D and Σ. Interestingly, $chase(D, \Sigma)$ is a *universal set model* of D and Σ, i.e., for each $I \in mods(D, \Sigma)$, there exists $J \in chase(D, \Sigma)$ and a homomorphism h_J such that $h_J(J) \subseteq I$ [21]. This implies that, given a UCQ Q, $D \cup \Sigma \models Q$ iff $I \models Q$, for each $I \in chase(D, \Sigma)$.

Guarded Negation FO. *Guarded negation first-order logic (GNFO)* restricts first-order logic by requiring that all occurrences of negation are of the form $\underline{a} \wedge \neg\varphi$, where $\underline{a}$ is an atom containing all the free variables of φ [18]. The formulas of GNFO are generated by the recursive definition $\varphi ::= r(t_1, \ldots, t_n) | t_1 = t_2 | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \exists X\ \varphi | \underline{a} \wedge \neg\varphi$. GNFO is strictly more expressive than *guarded first-order logic (GFO)* [22].

3 Known Results on Guarded-based DTGDs

We give an overview over known results, and we survey the best existing lower bounds that can be immediately inherited. Our discussion is outlined in Table 1, where each row corresponds to a fragment of DTGDs (which is decoded by substituting L for linear, G for guarded, W for weakly and F for frontier), each column corresponds to a complexity variant, and known completeness results are shown in boldface.

Overview. To the best of our knowledge, the only work done on query answering under guarded-based disjunctive DTGDs can be found in [14] and [15]. The first paper investigates the data complexity of query answering under (weakly-)guarded and linear DTGDs. For weakly-guarded it is ExpTime-complete, while for guarded and linear it is coNP-complete. Moreover, the case of atomic queries has been considered, and it was shown that it is in LogSpace. Notice that the above coNP-hardness is implicit in [23], where it was shown that query answering under a TBox with a single axiom $A \sqsubseteq B \sqcup C$, which is equivalent to $A(X) \rightarrow B(X) \vee C(X)$, is coNP-hard. The second paper studies both the combined and data complexity of atomic query answering under guarded and linear DTGDs. For guarded DTGDs the combined complexity is 2ExpTime-complete, while the data complexity is coNP-complete (which agrees with the analogous result above). For linear DTGDs the combined complexity is ExpTime-complete, while the data complexity is in AC_0 (improving the LogSpace upper bound mentioned above).

Notice that the AC_0 upper bound was obtained by showing that the problem is first-order rewritable.

Inherited Lower Bounds. The best existing lower bounds for our problem are the following: *(i)* 2EXPTIME in combined complexity, and also EXPTIME in case of bounded arity, for guarded DTGDs [9], *(ii)* 2EXPTIME for frontier-guarded DTGDs in case of bounded arity [2, 24], *(iii)* EXPTIME for DIDs in combined complexity; this holds since the rules employed in [15] to prove an analogous result for linear DTGDs are DIDs, and *(iv)* Π_{P}^2 for fixed sets of DIDs; this follows from a result in [25] which states that query answering under fixed universal GFO sentences is Π_{P}^2-hard. Notice that in the proof of this result a sentence of the form $\forall X \forall Y \forall Z\, r(X,Y,Z) \rightarrow s(X,Y) \oplus s(X,Z)$ is used; however, the result holds even if we replace $\oplus$ with $\vee$ since the minimal models of $\underline{a} \vee \underline{b}$ coincide with those of $\underline{a} \oplus \underline{b}$.

4 The Complexity of Query Answering

Apart from the three known completeness results which are shown in boldface in Table 1, for all the other cases the exact complexity is unknown. We tackle these open problems, and we present a complete complexity picture.

4.1 Combined Complexity

Upper Bound. First, we establish an upper bound for query answering under the most expressive class that we treat in this paper, i.e., weakly-frontier-guarded DTGDs, by exploiting a result on satisfiability of GNFO.

Theorem 1. *UCQAns under weakly-frontier-guarded DTGDs is in* 2EXPTIME *in combined complexity.*

Proof (sketch). We provide a reduction to satisfiability of GNFO which is in 2EXPTIME [18]. First, we polynomially reduce our problem to UCQAns under frontier-guarded DTGDs by exploiting the reduction from weakly-frontier-guarded TGDs to frontier-guarded TGDs proposed in [2]. Thus, given a UCQ Q, a database D, and set Σ of weakly-frontier-guarded DTGDs, there exists a polynomial translation τ such that $D \cup \Sigma \models Q$ iff $\tau(D) \cup \tau(\Sigma) \models \tau(Q)$, where $\tau(\Sigma)$ is a set of frontier-guarded DTGDs. It is easy to see that $\tau(\Sigma)$ can be equivalently rewritten as a GNFO formula [26]. More precisely, a frontier-guarded DTGD $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \exists \mathbf{Y}\, \psi(\mathbf{X}, \mathbf{Y})$ is equivalent to $\neg(\exists \mathbf{X}\, \varphi(\mathbf{X}) \wedge \neg \exists \mathbf{Y}\, \psi(\mathbf{X}, \mathbf{Y}))$ which falls in GNFO since all the free variables of $\exists \mathbf{Y}\, \psi(\mathbf{X}, \mathbf{Y})$ appear in the frontier-guard of $\varphi(\mathbf{X})$. Moreover, $\neg \tau(Q)$ trivially falls in GNFO. Therefore, $\tau(D) \wedge \tau(\Sigma) \wedge \neg \tau(Q)$ is a GNFO formula and the claim follows since $\tau(D) \cup \tau(\Sigma) \models \tau(Q)$ iff $\tau(D) \wedge \tau(\Sigma) \wedge \neg \tau(Q)$ is unsatisfiable. □

Notice that an alternative way to obtain the above result is to reduce our problem to query answering under GFO which is also in 2EXPTIME [25].

Lower Bound. Recall that CQAns for guarded TGDs is 2EXPTIME-hard [9] in combined complexity, while for frontier-guarded TGDs remains 2EXPTIME-hard even in the case of bounded arity [2]. Although these results, together with Theorem 1, close the combined complexity for (weakly-)(frontier-)guarded, and also the case of bounded arity for (weakly-)frontier-guarded, they are not strong enough to complete the complexity picture of our problem. In what follows we present a series of strong 2EXPTIME lower bounds for query answering. We assume the reader is familiar with Büchi automata and infinite trees (see, e.g., [27]).

Theorem 2. *CQAns under DIDs is 2*EXPTIME*-hard, even for predicates of arity at most two.*

Before proving the above theorem, we first introduce the following intermediate result: Given a finite set of labels Λ, we define a schema $\mathcal{S} = \{\mathit{child}, \mathit{parentorchild}\} \cup \Lambda$. These predicates are used to represent binary trees, with the obvious semantics. Given a CQ q over $\mathcal{S}$ and a Büchi tree automaton T over binary trees, where all states are accepting and have at least one successor state, we define q to be *valid* w.r.t. T, iff it holds that every (possibly infinite) binary tree accepted by T entails q. We claim that deciding this problem is hard for 2EXPTIME, which can be shown by adapting the 2EXPTIME-hardness of the same problem for finite trees over schema $\mathcal{S}' = \{\mathit{child}, \mathit{descendent}\} \cup \Lambda$ in [19].

Proof (sketch). The proof is by reduction from the validity problem of CQs q over $\mathcal{S}$ w.r.t. Büchi tree automata T over binary trees, as defined above. We construct a database D, a set Σ of DIDs, and a query $q' = q$ over schema $\mathcal{R}$, such that $D \cup \Sigma \models q'$ iff q is valid w.r.t. T. Let S_T be the set of states of T. The schema $\mathcal{R}$ is as follows: It includes $\mathcal{S}$, and for each pair (s, a), where $s \in S_T$ and $a \in \Lambda$, there are unary predicates s and $p_{s,a}$ in $\mathcal{R}$. Moreover, for each transition $(s, a) \mapsto s_1, s_2$ in the transition function δ of T, we have binary predicates $\mathit{child}^i[(s, a), s_1, s_2]$, for $i \in \{1, 2\}$, in $\mathcal{R}$. Intuitively, $\mathit{child}^i[(s, a), s_1, s_2](X, Y)$ says that Y is the i-th child of X, where X is in state s and labelled a, and Y is in state s_i. We now define Σ as follows:

- For each $s \in S_T$, $s(X) \rightarrow \bigvee_{a \in \Lambda \text{ and } \delta(s,a) \neq \varnothing} p_{s,a}(X)$
- For each $(s, a) \in S_T \times \Lambda$, $p_{s,a}(X) \rightarrow a(X)$
- For all transitions $(s, a) \mapsto s_1, s_2$:
  - $p_{s,a}(X) \rightarrow \exists Y\, \mathit{child}^i[(s, a), s_1, s_2](X, Y)$
  - $\mathit{child}^i[(s, a), s_1, s_2](X, Y) \rightarrow s_i(Y)$
  - $\mathit{child}^i[(s, a), s_1, s_2](X, Y) \rightarrow \mathit{child}(X, Y)$
- $\mathit{child}(X, Y) \rightarrow \mathit{parentorchild}(X, Y)$
- $\mathit{child}(X, Y) \rightarrow \mathit{parentorchild}(Y, X)$.

The database D contains a single atom $s_I(c)$, where s_I is the initial state of T. For each instance $I \in \mathit{chase}(D, \Sigma)$, if restricted to the child- and label-predicates only, by construction and due to the fact that each state of T has at least one successor, this is an infinite binary tree accepted by T. Moreover, the $\mathit{parentorchild}$-predicate in I is semantically equivalent to the $\mathit{parentorchild}$-relation of T, and therefore we get that $D \cup \Sigma \models q'$ only if q is valid w.r.t. T. The converse direction follows from the fact that all states are accepting. □

Notice that the constructed set Σ of DIDs in the above proof depends on T, and the underlying schema $\mathcal{R}$ contains a predicate for every state and label of T. This proof can now be extended, such that Σ is a fixed set of DIDs and $\mathcal{R}$ a fixed schema with arity at most two. However, to devise this encoding, we need the expressive power of UCQs.

Theorem 3. *UCQAns under fixed sets of DIDs is* 2EXPTIME*-hard, even for predicates of arity at most two.*

Proof (sketch). We adapt the proof of Theorem 2 as follows: Instead of labelling each tree node with a state s and label a, we generate a chain of nodes, with the length of the chain encoding s and a. This can be done by the DIDs $\mathit{next}(X) \rightarrow \exists Y\, \mathit{chain}(X, Y)$ and $\mathit{chain}(X, Y) \rightarrow \mathit{next}(Y) \vee \mathit{end}(Y)$. Using this adaptation, neither the schema nor the set of DIDs depends on T any longer. The CQ of the proof of Theorem 2 can now be carefully adapted to this new encoding of states and labels, and also to check that: *(i)* any chain has length at most n, where n is polynomial in the size of T, and *(ii)* each node and its two children are consistent with the transition function of T. □

In the following, we will show that the above theorem holds also for CQs, at the expense of increasing the arity of the underlying schema by one. In the sequel, given a schema $\mathcal{R}$, let $\mathit{arity}(\mathcal{R})$ be the maximum arity over all predicates of $\mathcal{R}$. Notice that the following technical result holds for arbitrary DTGDs, and not just for DIDs.

Lemma 1. *Let $\mathcal{R}$ be a relational schema. Consider a UCQ Q over $\mathcal{R}$, a database D for $\mathcal{R}$, and a set Σ of DTGDs over $\mathcal{R}$. We can construct in polynomial time a CQ q' over a schema $\mathcal{R}'$, a database D' for $\mathcal{R}'$, and a set Σ' of DTGDs such that $\mathit{arity}(\mathcal{R}') = \mathit{arity}(\mathcal{R}) + 1$, and $D \cup \Sigma \models Q$ iff $D' \cup \Sigma' \models q'$.*

Proof (sketch). The schema $\mathcal{R}'$ is obtained from $\mathcal{R}$ by increasing the arity of every predicate by one. Moreover, we add three predicates or, true and false. Each DTGD in Σ is adapted in such a way that it always propagates this additional position to the atoms in the head. Each CQ $q_i \in Q$ is translated into a new CQ $q'_i[X_i]$, where a fresh variable X_i is added to all atoms in q_i at the new position. The body of the new query q' is $\mathit{false}(Z_1) \bigwedge_{q_i \in Q} q'_i[X_i] \wedge \mathit{or}(Z_i, X_i, Z_{i+1}) \wedge \mathit{true}(Z_{k+1})$, where $k = |Q|$. The database D' is obtained from D by extending each atom in such a way that the fresh constant $t \in \Gamma$ appears in the new position. Also, the atoms $\mathit{true}(t)$ and $\mathit{false}(f)$, where $f \in \Gamma$ is a fresh constant, are added. Furthermore, we add an isomorphic image of every $q'_i[X_i]$ to D', where X_i is replaced by f. Finally, we add the atoms $\mathit{or}(t, t, t)$, $\mathit{or}(f, t, t)$, $\mathit{or}(t, f, t)$ and $\mathit{or}(f, f, f)$.

We can show that the above construction is correct. For any query $q'_i[X_i]$, there exists a homomorphism mapping it to D'. However, this is not useful to satisfy q', as X_i is mapped to f. By construction however, the only way to satisfy q' is to map at least one subquery $q'_i[X_i]$ to $\mathit{chase}(D', \Sigma')$, such that X_i maps to t. Note that the only atoms in $\mathit{chase}(D', \Sigma')$ containing t are the ones obtained from the original copy of D. Thus, we have that whenever a subquery $q_i \in Q$ is true in a model of $D \cup \Sigma$, then q' is true in the corresponding model of $D' \cup \Sigma'$, and the claim follows. □

Theorem 3 and Lemma 1 immediately imply the following:

Corollary 1. *CQAns under fixed sets of DIDs is* 2ExpTime*-hard, even for predicates of arity at most three.*

Interestingly, the above corollary closes an open question stated in [25], regarding the complexity of query answering under fixed GFO sentences. It was shown that the problem in question is PSpace-hard even for CQs, and in ExpTime in case of acyclic CQs. However, the exact complexity was left as an open problem. Clearly, Corollary 1 gives a 2ExpTime-completeness result since query answering under GFO is in 2ExpTime in general. By combining Theorem 1 and Corollary 1, we get the following.

Corollary 2. *(U)CQAns under (weakly-)(frontier-)guarded DTGDs, linear DTGDs and DIDs is* 2ExpTime*-complete in combined complexity. This holds even for predicates of arity at most three, and for fixed sets of dependencies.*

4.2 Data Complexity

As already discussed in Section 3, for guarded and weakly-guarded DTGDs the data complexity is coNP-complete and ExpTime-complete, respectively. Below, we show that it remains the same for (weakly-)frontier-guarded DTGDs.

Theorem 4. *(U)CQAns under frontier-guarded DTGDs is* coNP*-complete, while for weakly-frontier-guarded DTGDs it is* ExpTime*-complete in data complexity.*

Proof (sketch). The coNP upper bound is obtained by reducing our problem to UCQAns under GFO sentences. This can be done by employing the linear reduction of CQAns under frontier-guarded TGDs to UCQAns under GFO sentences given in [2]. The lower bound follows immediately since CQAns under DIDs is already coNP-hard. Consider now a UCQ Q, a database D, and set Σ of weakly-frontier-guarded DTGDs. First, we reduce our problem to UCQAns under frontier-guarded DTGDs by replacing the non-affected variables in rules with all possible constants in D. Clearly, the obtained set Σ' is of exponential size in the number of non-affected variables, but of polynomial size in $|terms(D)|$. As discussed above, a linear translation τ exists such that $D \cup \Sigma' \models Q$ iff $D \cup \tau(\Sigma') \models \tau(Q)$, where $\tau(\Sigma')$ is a GFO sentence and $\tau(Q)$ a UCQ. It is important to say that, although $|\tau(Q)|$ depends on D, the size of each CQ of $\tau(Q)$ does not depend on D. As shown in [25], UCQAns under GFO is in 2ExpTime w.r.t. to the size of each CQ of the given UCQ and the maximum arity of the schema, and in ExpTime w.r.t. to the size of the sentence. Since the size of each query of $\tau(Q)$ and the maximum arity of the schema are constant, and the size of $\tau(\Sigma')$ is polynomial in D, we get an ExpTime upper bound w.r.t. D. The lower bound follows immediately since UCQAns for weakly-guarded TGDs is ExpTime-hard [9]. □

5 Reducing the Complexity

In this section, we demonstrate a way of reducing the combined complexity of query answering under DIDs. We consider *frontier-one* DIDs, i.e., DIDs with a frontier of cardinality exactly one, for which the complexity is ExpTime-complete. Notice that the class of frontier-one TGDs has been proposed in [1]. Clearly, frontier-one formalisms are quite close to DL axioms since concept inclusions propagate only one object.

Theorem 5. *(U)CQAns for frontier-one DIDs is* ExpTime*-complete in combined complexity.*

Proof (sketch). Consider a database D, and a set Σ of frontier-one DIDs. It is possible to associate a tree structure with every instance $I \in \mathit{chase}(D, \Sigma)$: I is partitioned into bags of atoms, such that I' is one such bag, and for each term t occurring in an atom of $I \setminus I'$, there exists a bag, denoted $\mathit{bag}(t)$, such that the atoms in $\mathit{bag}(t)$ contain t; t is called the *input value* of $\mathit{bag}(t)$. These bags are used as labels of the tree structure, such that the bag containing I' labels the root, and two bags $\mathit{bag}(t_1)$, $\mathit{bag}(t_2)$ are in a parent-child relation iff there exists an atom containing t_2 in $\mathit{bag}(t_1)$. Due to the monadic nature of frontier-one DIDs, the number of atoms in each bag is polynomial in D and Σ and the number of isomorphic bags is exponential in the size of Σ. The above tree structure was introduced in [28] for finite instances, but can be extended to infinite trees. It is therefore possible to show that there exists a *Rabin automaton* (see, e.g., [27]) that is empty iff no instance of $\mathit{chase}(D, \Sigma)$ satisfies q. This tree automaton represents the tree structure of the instances of $\mathit{chase}(D, \Sigma)$ that do not satisfy q. Moreover, the size of the automaton is exponential in D and Σ due to the fact that, as shown in [28], for every instance $I \in \mathit{chase}(D, \Sigma)$ with $I \not\models q$, the tree structure is *diversified* (i.e., there is no bag except the root that contains two atoms with the same predicate, and there are no directly related bags sharing a term at the same position). Given that Rabin automata can be checked for emptiness in linear time, we establish the desired upper bound.

The lower bound is obtained by a careful adaptation of the proof of Theorem 2 in order to simulate a PSpace alternating Turing machine using a chain of length n, instead of a binary tree of depth n, to store the configurations. □

Another formalism with a lower combined complexity is the class of *full-identity* DIDs, that is, rules of the form $r(X_1, \ldots, X_n) \rightarrow \bigvee_{i=1}^{m} p_i(X_1, \ldots, X_n)$ which allow us only to copy a tuple. It is easy to show that the combined complexity reduces to coNP-complete. As we are not able to permutate terms, each instance of the chase is of polynomial size in the database and the schema. Thus, it suffices to guess such an instance I, and check that it does not entail the query. The lower bound is implicit in [23], where it was shown that query answering under rules of the form $A(X) \rightarrow B(X) \vee C(X)$ is coNP-hard in data complexity. Notice that query answering under DIDs where each rule is frontier-one *or* full-identity is ExpTime-complete.

6 Relationships with Existing DLs

As already shown in [15], guarded DTGDs are strictly more expressive than $\mathcal{ELU}$ [29], that is, the well-known DL $\mathcal{EL}$ extended with disjunction. It is indeed straightforward to see that every normalized $\mathcal{ELU}$ TBox, which may contains axioms of the form $A \sqsubseteq B$, $A \sqcap B \sqsubseteq C$, $A \sqsubseteq \exists R.B$ and $A \sqsubseteq B \sqcup C$, where A, B, C are concept names and R is a role name, can be translated in logarithmic space into a set of guarded DTGDs.

The goal of this section is to show an analogous result for *DL-Lite*$_{bool}^{\mathcal{H}}$ [20], one of the most expressive languages of the DL-Lite family, and also to investigate the impact of our previously established results on query answering under description logics, and in particular under *DL-Lite*$^{\mathcal{H}}_{bool}$. Let us recall that this logical language contains object names $a_0, a_1, \ldots$, concept names $A_0, A_1, \ldots$, and role names $P_0, P_1, \ldots$. Complex *roles* R and *concepts* C are defined as follows: $R ::= P_k | P_k^-$, $B ::= \bot | A_k | \exists R$ and $C ::= B | \neg B | C_1 \sqcap C_2$. A TBox $\mathcal{T}$ is a finite set of concept and role inclusion axioms of the form $C_1 \sqsubseteq C_2$ and $R_1 \sqsubseteq R_2$, and its semantics can be defined by translating it into first-order logic by using an operator τ. We denote by $\mathcal{T}_C$ and $\mathcal{T}_R$ the set of concept inclusions and the set of role inclusions of $\mathcal{T}$, respectively.

Expressive Power. We establish that *frontier-guarded*[$\bot$] DTGDs, that is, the formalism obtained by combining frontier-guarded DTGDs with *negative constraints (NCs)* of the form $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bot$, where φ is a conjunction of atoms without any syntactic restrictions, are strictly more expressive than *DL-Lite*$^{\mathcal{H}}_{bool}$. Notice that the complexity of query answering under frontier-guarded[$\bot$] DTGDs is the same as for frontier-guarded, since deciding whether the given set of dependencies is consistent can be reduced to query answering under frontier-guarded DTGDs; see, e.g., [5].

It is easy to see that role inclusions are translated by τ into IDs; e.g., $\tau(R \sqsubseteq S) = R(X,Y) \rightarrow S(X,Y)$. However, in general, for a concept inclusion $C \sqsubseteq D$, $\tau(C \sqsubseteq D)$ is neither a DTGD nor an NC; however, we can show that any $\tau(C \sqsubseteq D)$ can be transformed into a set of frontier-guarded[$\bot$] DTGDs in polynomial time.

Lemma 2. *For every concept inclusion $\alpha = C \sqsubseteq D$, a set Σ_α of frontier-guarded[$\bot$] DTGDs can be constructed in polynomial time such that $\tau(C \sqsubseteq D)$ and Σ_α are equisatisfiable.*

Proof (sketch). Let $\psi = \tau(C \sqsubseteq D)$. Viewed as an implication, we can treat the left-hand side as a conjunction and the right-hand side as a disjunction of subformulas. Whenever such a subformula $\varphi(X)$ is not atomic or existential (or a negated version of the two), we introduce a new auxiliary predicate $t_\varphi(X)$ and two implications $\varphi(X) \leftrightarrow t_\varphi(X)$. If φ is itself a disjunction, we additionally split the first implication (i.e., where $\varphi(X)$ is on the left) into separate implications for each disjunct, preserving the right-hand side. We then recursively apply this transformation to the new implications, until a fixpoint is reached. The resulting implications, after removing negations by converting negated conjuncts on the left to non-negated disjuncts on the right (and conversely for disjuncts on the right) are clearly DTGDs or NCs. □

Lemma 2 implies that $\Sigma_C = \bigcup_{\alpha \in \mathcal{T}_C} \Sigma_\alpha$ is a set of frontier-guarded[$\bot$] DTGDs which can be constructed in polynomial time, and it is equisatisfiable with $\mathcal{T}_C$. Thus, $\tau(\mathcal{T}_R) \cup \Sigma_C$ is a set of frontier-guarded[$\bot$] DTGDs which is equisatisfiable with $\mathcal{T}$. Since $s(X) \rightarrow r(X,X)$ is not expressible in *DL-Lite*$^{\mathcal{H}}_{bool}$, the next result follows.

Theorem 6. *Frontier-guarded[$\bot$] DTGDs are strictly more expressive than DL-Lite$^{\mathcal{H}}_{bool}$.*

Complexity Results. By exploiting the above construction, query answering under *DL-Lite*$^{\mathcal{H}}_{bool}$ can be reduced in polynomial time to CQAns under frontier-guarded[$\bot$] DTGDs. Recall that an ABox $\mathcal{A}$ is a finite set of assertions of the form $A_k(a_i)$, $\neg A_k(a_i)$, $P_k(a_i, a_j)$ and $\neg P_k(a_i, a_j)$; the semantics of $\mathcal{A}$ are defined by τ. A TBox $\mathcal{T}$ together with $\mathcal{A}$ constitute a *knowledge base (KB)* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

Lemma 3. *UCQAns under DL-Lite$^{\mathcal{H}}_{bool}$ knowledge bases can be reduced in polynomial time to UCQAns under frontier-guarded[$\bot$] DTGDs.*

Proof (sketch). Consider a *DL-Lite*$^{\mathcal{H}}_{bool}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. Let $D_{\mathcal{A}}$ be the database obtained from $\mathcal{A}$ by replacing each negated atom $\neg A(a)$ and $\neg R(a, b)$ with $A_{\neg}(a)$ and $R_{\neg}(a, b)$, respectively, where $A_{\neg}$ and $R_{\neg}$ are auxiliary predicates. Let $\Sigma_{\mathcal{T}} = \tau(\mathcal{T}_R) \cup \Sigma_C \cup \Sigma_{\bot}$, where $\Sigma_{\bot}$ contains an NC $A(X), A_{\neg}(X) \rightarrow \bot$ and $R(X, Y), R_{\neg}(X, Y) \rightarrow \bot$, for each concept A and role R in $\mathcal{T}$, respectively. It is not difficult to verify that $\mathcal{K} \models Q$ iff $D_{\mathcal{A}} \cup \Sigma_{\mathcal{T}} \models Q$, for every UCQ Q over $\mathcal{K}$. Since $\Sigma_{\mathcal{T}}$ is a set of frontier-guarded[$\bot$] DTGDs that can be constructed in polynomial time, the claim follows. $\Box$

It is interesting to observe that the rules employed in the proof of Theorem 2, can be easily rewritten as DL axioms. This immediately gives us the following lower bound.

Theorem 7. *Let $\mathcal{L}$ be a DL able to express inclusions of the form $C_1 \sqsubseteq C_2 \sqcup C_3$, $C \sqsubseteq \exists R$, $\exists R \sqsubseteq C$, $R_1 \sqsubseteq R_2$ and $R_1 \sqsubseteq R_2^-$, where C, C_i are concepts and R, R_i are roles. Then, CQAns under $\mathcal{L}$ is* 2EXPTIME*-hard.*

In [20] it was shown that query answering under *DL-Lite*$^{\mathcal{H}}_{bool}$ is coNP-complete in data complexity; however, the combined complexity was not investigated and left as an open problem. Since *DL-Lite*$^{\mathcal{H}}_{bool}$ is a description logic equipped with limited existential qunatification, role inverse and union, Theorems 2 and 7, together with Lemma 3, imply the next complexity result.

Corollary 3. *CQAns under DL-Lite$^{\mathcal{H}}_{bool}$ knowledge bases is* 2EXPTIME*-complete in combined complexity.*

Interestingly, the above corollary significantly strengthens a similar result for the $\mathcal{ALCI}$ DL in [24].

7 Conclusion

We studied the query answering problem under (weakly-)(frontier-)guarded disjunctive TGDs and their main subclasses. Interestingly, query answering under a fixed set of disjunctive IDs is already 2EXPTIME-hard. We also investigated the impact of our results on query answering under DL-based formalisms; in particular, we showed that this problem for DLs equipped with limited existential quantification, role inverse and union is 2EXPTIME-hard. Regarding future work, we intend to study the impact of the addition of disjunction to non-guarded-based classes of TGDs, in the same complete fashion as in this paper.

Acknowledgements. Pierre Bourhis acknowledges his EPSRC Grant EP/H017690/1 "Query-Driven Data Acquisition from Web-based Datasources", Michael Morak his DOC Fellowship of the Austrian Academy of Sciences, and Andreas Pieris his EPSRC Grant EP/G055114/1 "Constraint Satisfaction for Configuration: Logical Fundamentals, Algorithms and Complexity" and ERC Grant 246858 "DIADEM".

References

1. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artif. Intell. **175**(9-10) (2011) 1620–1654
2. Baget, J.F., Mugnier, M.L., Rudolph, S., Thomazo, M.: Walking the complexity lines for generalized guarded existential rules. In: Proc. of IJCAI. (2011) 712–717
3. Krötzsch, M., Rudolph, S.: Extending decidable existential rules by joining acyclicity and guardedness. In: Proc. of IJCAI. (2011) 963–968
4. Thomazo, M., Baget, J.F., Mugnier, M.L., Rudolph, S.: A generic querying algorithm for greedy sets of existential rules. In: Proc. of KR. (2012)
5. Calì, A., Gottlob, G., Lukasiewicz, T.: A general Datalog-based framework for tractable query answering over ontologies. J. Web Sem. **14** (2012) 57–83
6. Patel-Schneider, P.F., Horrocks, I.: A comparison of two modelling paradigms in the semantic web. J. Web Sem. **5**(4) (2007) 240–250
7. Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. In: Proc. of ICALP. (1981) 73–85
8. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. Theor. Comput. Sci. **336**(1) (2005) 89–124
9. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. In: Proc. of KR. (2008) 70–80
10. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: The query answering problem. Artif. Intell. **193** (2012) 87–128
11. Leone, N., Manna, M., Terracina, G., Veltri, P.: Efficiently computable Datalog$^\exists$ programs. In: Proc. of KR. (2012)
12. Deutsch, A., Tannen, V.: Reformulation of XML queries and constraints. In: Proc. of ICDT. (2003) 225–241
13. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive Datalog. ACM Trans. Database Syst. **22**(3) (1997) 364–418
14. Alviano, M., Faber, W., Leone, N., Manna, M.: Disjunctive Datalog with existential quantifiers: Semantics, decidability, and complexity issues. TPLP **12**(4-5) (2012) 701–718
15. Gottlob, G., Manna, M., Morak, M., Pieris, A.: On the complexity of ontological reasoning under disjunctive existential rules. In: Proc. of MFCS. (2012) 1–18
16. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. Autom. Reasoning **39**(3) (2007) 385–429
17. Baader, F.: Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In: Proc. of IJCAI. (2003) 319–324
18. Bárány, V., ten Cate, B., Segoufin, L.: Guarded negation. In: Proc. of ICALP. (2011) 356–367
19. Björklund, H., Martens, W., Schwentick, T.: Optimizing conjunctive queries over trees using schema information. In: Proc. of MFCS. (2008) 132–143
20. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. **36** (2009) 1–69
21. Deutsch, A., Nash, A., Remmel, J.B.: The chase revisisted. In: Proc. of PODS. (2008) 149–158
22. Andréka, H., van Benthem, J., Németi, I.: Modal languages and bounded fragments of predicate logic. J. Philosophical Logic **27** (1998) 217–274
23. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proc. of KR. (2006) 260–270
24. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. of IJCAR. (2008) 179–193

25. Bárány, V., Gottlob, G., Otto, M.: Querying the guarded fragment. In: Proc. of LICS. (2010) 1–10
26. Bárány, V., ten Cate, B., Otto, M.: Queries with guarded negation. PVLDB **5**(11) (2012) 1328–1339
27. Grädel, E., Thomas, W., Wilke, T., eds.: Automata, Logics, and Infinite Games: A Guide to Current Research. Springer (2002)
28. Benedikt, M., Bourhis, P., Senellart, P.: Monadic Datalog containment. In: Proc. of ICALP. (2012) 79–91
29. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. of IJCAI. (2005) 364–369