

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

Pierre Bourhis and Michael Morak and Andreas Pieris

Department of Computer Science  
 University of Oxford, United Kingdom  
*firstname.lastname@cs.ox.ac.uk*

## Abstract

We study 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 (IDs). Our main result states that conjunctive query answering under a *fixed* set of disjunctive IDs is 2EXPTIME-hard. This quite surprising result together with a 2EXPTIME upper bound for weakly-frontier-guarded disjunctive rules, obtained by exploiting recent results on guarded negation first-order logic, gives us a complete picture of the computational complexity of our problem. We also consider a natural subclass of disjunctive IDs, namely frontier-one (only one variable is propagated), for which the combined complexity decreases to EXPTIME. Finally, we show that frontier-guarded rules, combined with negative constraints, are strictly more expressive than  $DL\text{-}Lite_{bool}^{\exists}$ , one of the most expressive languages of the DL-Lite family. We also show that query answering under  $DL\text{-}Lite_{bool}^{\exists}$  is 2EXPTIME-complete in combined complexity.

## 1 Introduction

Rule-based languages have a prominent presence in the areas of artificial intelligence 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., [Baget *et al.*, 2011a; 2011b; Krötzsch and Rudolph, 2011; Thomazo *et al.*, 2012; Cali *et al.*, 2012a]. This interest stems from the inability of plain Datalog to infer the existence of new objects which are not already in the extensional database [Patel-Schneider and Horrocks, 2007]. The obtained rules are known under a variety of names such as *existential rules*, *tuple-generating dependencies (TGDs)*, and *Datalog<sup>±</sup> rules*. Unfortunately, the addition of existential quantification as explained above can easily lead to undecidability of the main reasoning tasks, and in particular of conjunctive query answering [Beeri and Vardi, 1981]. Therefore, several concrete languages which guarantee decidability have been pro-

posed, see, e.g., [Fagin *et al.*, 2005; Cali *et al.*, 2008; Baget *et al.*, 2011a; Krötzsch and Rudolph, 2011; Cali *et al.*, 2012a; 2012b; Leone *et al.*, 2012]. Nevertheless, TGDs in general are not powerful enough for nondeterministic reasoning.

**Example 1** Consider the following set of rules:

$$\begin{aligned} \sigma_1 & : \text{parentOf}(X, Y), \text{father}(Y) \rightarrow \exists Z \text{gparentOf}(X, Z) \\ \sigma_2 & : \text{gparentOf}(X, Y) \rightarrow \text{child}(Y) \\ \sigma_3 & : \text{child}(X) \rightarrow \text{boy}(X) \vee \text{girl}(X) \end{aligned}$$

which expresses a simple and natural statement: “a parent of a father is a grandparent of some child, which in turn is a boy or a girl!”. Although  $\sigma_1$  and  $\sigma_2$  are TGDs,  $\sigma_3$  is not, due to the presence of disjunction in the head. ■

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)* [Deutsch and Tannen, 2003]. Such an extension of plain Datalog (a.k.a. *full TGDs*), called *disjunctive Datalog*, has been studied in [Eiter *et al.*, 1997]. More recently, special cases of the problem of query answering under guarded-based DTGDs have been investigated [Alviano *et al.*, 2012; Gottlob *et al.*, 2012]. 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. Guardedness is a well-accepted paradigm, giving rise to robust languages that capture important lightweight description logics such as DL-Lite [Calvanese *et al.*, 2007] and  $\mathcal{EL}$  [Baader, 2003].

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 [Cali *et al.*, 2008; Baget *et al.*, 2011a], and their main subclasses, i.e., linear DTGDs and disjunctive IDs (DIDs) [Cali *et al.*, 2012a]? 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. Indeed, we establish 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 easily obtained by exploiting results on more expressive languages such as guarded negation first-order logic [Bárány *et al.*, 2011], while the lower bounds are inherited from existing results on TGDs.
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 surprisingly strong lower bounds are established by a reduction from an appropriate variant of the validity problem of CQs w.r.t. a Büchi automaton [Björklund *et al.*, 2008]. Together with the 2EXPTIME upper bound discussed above, this gives us the complete picture for the computational 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  $DL\text{-}Lite_{bool}^{\mathcal{H}}$  [Artale *et al.*, 2009], one of the most expressive languages of the DL-Lite family. This allows us to show that query answering under  $DL\text{-}Lite_{bool}^{\mathcal{H}}$  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/ijcai2013-techrep.pdf>.

## 2 Preliminaries

**Technical Definitions.** We define the following pairwise disjoint (infinite) sets: a set  $\Gamma$  of *constants*, a set  $\Gamma_N$  of *labeled nulls*, and a set  $\Gamma_V$  of *regular variables*. We denote by  $\mathbf{X}$  sequences (or sets) of variables  $X_1, \dots, X_k$ . A *relational schema*  $\mathcal{R}$  is a set of *relational symbols* (or *predicates*). A position  $r[i]$  in  $\mathcal{R}$  is identified by a predicate  $r \in \mathcal{R}$  and its  $i$ -th argument. A *term*  $t$  is a constant, null, or variable. An *atom* has the form  $r(t_1, \dots, t_n)$ , where  $r$  is a relation, and  $t_1, \dots, t_n$  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  $\text{terms}(D) \subset \Gamma$ . We assume the reader is familiar with (*unions of*) *conjunctive queries* ((U)CQs). The answer to a

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

A DTGD  $\sigma$  is *guarded* if there exists an atom  $\underline{a} \in \text{body}(\sigma)$ , called *guard*, which contains all the variables occurring in  $\text{body}(\sigma)$ . *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 [Cali *et al.*, 2008]. The concept of frontier can be used to generalize (weakly-)guarded DTGDs. The *frontier* of a DTGD  $\sigma$  is the set of variables  $\text{var}(\text{body}(\sigma)) \cap \text{var}(\text{head}(\sigma))$ .  $\sigma$  is *frontier-guarded* if there exists an atom  $\underline{a} \in \text{body}(\sigma)$  which contains all the variables occurring in its frontier. The class of *weakly-frontier-guarded* DTGDs is defined analogously. A DTGD  $\sigma$  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  $\Sigma$ , denoted  $\text{mods}(D, \Sigma)$ , 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  $\Sigma$ , denoted  $\text{ans}(q, D, \Sigma)$ , is the set of tuples of constants  $\bigcap_{I \in \text{mods}(D, \Sigma)} \{\mathbf{t} \mid \mathbf{t} \in q(I)\}$ . The answer to a Boolean CQ  $q$  w.r.t.  $D$  and  $\Sigma$  is *positive*, denoted  $D \cup \Sigma \models q$ , if  $\langle \rangle \in \text{ans}(q, D, \Sigma)$ . The answer to a UCQ w.r.t.  $D$  and  $\Sigma$  is defined analogously. The problem, called CQAns, tackled in this work is defined as follows: given a CQ  $q$ , a database  $D$ , a set  $\Sigma$  of DTGDs, and a tuple of constants  $\mathbf{t}$ , decide whether  $\mathbf{t} \in \text{ans}(q, D, \Sigma)$ . 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 [Deutsch and Tannen, 2003]. 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  $\sigma$  is *applicable* to  $I$  if there exists a homomorphism  $h$  such that  $h(\varphi(\mathbf{X})) \subseteq I$ , and the result of applying  $\sigma$  to  $I$  with  $h$  is the set  $\{I_1, \dots, I_n\}$ , where  $I_i = I \cup h'(\psi_i(\mathbf{X}, \mathbf{Y}))$ , for each  $i \in \{1, \dots, n\}$ , 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  $I \langle \sigma, h \rangle \{I_1, \dots, I_n\}$ . A *disjunctive chase tree*

of a database  $D$  and a set  $\Sigma$  of DTGDs is a (possibly infinite) tree such that the root is  $D$ , and for every node  $I$ , assuming that  $\{I_1, \dots, I_n\}$  are the children of  $I$ , there exists  $\sigma \in \Sigma$  and a homomorphism  $h$  such that  $I(\sigma, h)\{I_1, \dots, I_n\}$ . The disjunctive chase algorithm for  $D$  and  $\Sigma$  consists of an exhaustive application of DTGD chase steps in a fair fashion, which leads to a disjunctive chase tree  $T$  of  $D$  and  $\Sigma$ ; we denote by  $\text{chase}(D, \Sigma)$  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  $\text{chase}(D, \Sigma)$  is a model of  $D$  and  $\Sigma$ . Interestingly,  $\text{chase}(D, \Sigma)$  is a *universal set model* of  $D$  and  $\Sigma$ , i.e., for each  $I \in \text{mods}(D, \Sigma)$ , there exists  $J \in \text{chase}(D, \Sigma)$  and a homomorphism  $h_J$  such that  $h_J(J) \subseteq I$  [Deutsch *et al.*, 2008]. This implies that, given a UCQ  $Q$ ,  $D \cup \Sigma \models Q$  iff  $I \models Q$ , for each  $I \in \text{chase}(D, \Sigma)$ .

**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  $\varphi$  [Bárány *et al.*, 2011]. The formulas of GNFO are generated by the recursive definition  $\varphi ::= r(t_1, \dots, t_n) \mid t_1 = t_2 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \exists X \varphi \mid \underline{a} \wedge \neg\varphi$ . GNFO is strictly more expressive than *guarded first-order logic (GFO)* [Andréka *et al.*, 1998].

### 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 [Alviano *et al.*, 2012] and [Gottlob *et al.*, 2012]. 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 [Calvanese *et al.*, 2006], 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<sub>0</sub> (improving the LOGSPACE upper bound mentioned above). Notice that the AC<sub>0</sub> 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 [Cali *et al.*, 2008], (ii) 2EXPTIME for

	<b>Combined compl.</b>	<b>Bounded arity</b>	<b>Fixed rules</b>	<b>Data compl.</b>
<b>L/DID</b>	EXP-h	$\Pi_p^2$ -h	$\Pi_p^2$ -h	<b>coNP-c</b>
<b>G</b>	2EXP-h	EXP-h	$\Pi_p^2$ -h	<b>coNP-c</b>
<b>W-G</b>	2EXP-h	EXP-h	EXP-h	<b>EXP-c</b>
<b>F-G</b>	2EXP-h	2EXP-h	$\Pi_p^2$ -h	coNP-h
<b>W-F-G</b>	2EXP-h	2EXP-h	EXP-h	EXP-h

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

frontier-guarded DTGDs in case of bounded arity [Baget *et al.*, 2011b; Lutz, 2008], (iii) EXPTIME for DIDs in combined complexity; this holds since the rules employed in [Gottlob *et al.*, 2012] to prove an analogous result for linear DTGDs are DIDs, and (iv)  $\Pi_p^2$  for fixed sets of DIDs; this follows from a result in [Bárány *et al.*, 2010] which states that query answering under fixed universal GFO sentences is  $\Pi_p^2$ -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 picture of the complexity of the problem studied here.

### 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 [Bárány *et al.*, 2011]. 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 [Baget *et al.*, 2011b]. Thus, given a UCQ  $Q$ , a database  $D$ , and set  $\Sigma$  of weakly-frontier-guarded DTGDs, there exists a polynomial translation  $\tau$  such that  $D \cup \Sigma \models Q$  iff  $\tau(D) \cup \tau(\Sigma) \models \tau(Q)$ , where  $\tau(Q)$  is a set of frontier-guarded DTGDs. It is easy to see that  $\tau(\Sigma)$  can be equivalently rewritten as a GNFO formula [Bárány *et al.*, 2012]. 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.  $\square$

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 [Bárány *et al.*, 2010].

**Lower Bound.** Recall that CQAns for guarded TGDs is 2EXPTIME-hard [Calì *et al.*, 2008] in combined complexity, while for frontier-guarded TGDs remains 2EXPTIME-hard even in the case of bounded arity [Baget *et al.*, 2011b]. 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., [Grädel *et al.*, 2002]).

**Theorem 2** *CQAns under DIDs is 2EXPTIME-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  $\Lambda$ , we define a schema  $\mathcal{S} = \{child, parentorchild\} \cup \Lambda$ . 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  $\mathcal{S}' = \{child, descendent\} \cup \Lambda$  in [Björklund *et al.*, 2008].

**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  $\Sigma$  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  $\delta$  of  $T$ , we have binary predicates  $child^i[(s, a), s_1, s_2]$ , for  $i \in \{1, 2\}$ , in  $\mathcal{R}$ . Intuitively,  $child^i[(s, a), s_1, s_2](X, Y)$  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  $\Sigma$  as follows:

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

The database  $D$  contains a single atom  $s_I(c)$ , where  $s_I$  is the initial state of  $T$ . For each instance  $I \in chase(D, \Sigma)$ , 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 *parentorchild*-predicate in  $I$  is semantically equivalent to the *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.  $\square$

Notice that the constructed set  $\Sigma$  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  $\Sigma$  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 in the following way: 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 easily be done by the DIDs  $next(X) \rightarrow \exists Y chain(X, Y)$  and  $chain(X, Y) \rightarrow next(Y) \vee end(Y)$ . 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  $T$ .  $\square$

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  $arity(\mathcal{R})$  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 4** *Let  $\mathcal{R}$  be a relational schema. Consider a UCQ  $Q$  over  $\mathcal{R}$ , a database  $D$  for  $\mathcal{R}$ , and a set  $\Sigma$  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  $\Sigma'$  of DTGDs such that  $arity(\mathcal{R}') = arity(\mathcal{R}) + 1$ , 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  $\Sigma$  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  $false(Z_1) \wedge_{q_i \in Q} q'_i[X_i] \wedge or(Z_i, X_i, Z_{i+1}) \wedge true(Z_{k+1})$ , 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 *true*( $t$ ) and *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 *or*( $t, t, t$ ), *or*( $f, t, t$ ), *or*( $t, f, t$ ) and *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  $chase(D', \Sigma')$ , such that  $X_i$  maps to  $t$ . Note that the only atoms in  $chase(D', \Sigma')$  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.  $\square$

Theorem 3 and Lemma 4 immediately imply the following:

**Corollary 5** *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 [Bárány *et al.*, 2010], 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 5 gives a 2EXPTIME-completeness result since query answering under GFO is in 2EXPTIME in general. By combining Theorem 1 and Corollary 5, we get the following.

**Corollary 6** *(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 7** *(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 [Baget *et al.*, 2011b]. The lower bound follows immediately since CQAns under DIDs is already coNP-hard. Consider now a UCQ  $Q$ , a database  $D$ , and set  $\Sigma$  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  $\Sigma'$  is of exponential size in the number of non-affected variables, but of polynomial size in  $|terms(D)|$ . As discussed above, a linear translation  $\tau$  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 [Bárány *et al.*, 2010], 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 [Calì *et al.*, 2008].  $\square$

## 5 Reducing the Complexity

In contrast to our previous results, 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 reduced to EXPTIME-complete. Notice that the class of

frontier-one TGDs has been proposed in [Baget *et al.*, 2011a]. Clearly, frontier-one formalisms are quite close to DL axioms since concept inclusions propagate only one object.

**Theorem 8** *(U)CQAns for frontier-one DIDs is EXPTIME-complete in combined complexity.*

**Proof (sketch).** Consider a database  $D$ , and a set  $\Sigma$  of frontier-one DIDs. It is possible to associate a tree structure with every instance  $I \in chase(D, \Sigma)$ :  $I$  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  $bag(t)$ , such that the atoms in  $bag(t)$  contain  $t$ ;  $t$  is called the *input value* of  $bag(t)$ . These bags are used as labels of the tree structure, such that the bag containing  $I'$  labels the root, and two bags  $bag(t_1), bag(t_2)$  are in a parent-child relation iff there exists an atom containing  $t_2$  in  $bag(t_1)$ . Due to the monadic nature of frontier-one DIDs, the number of atoms in each bag is polynomial in  $D$  and  $\Sigma$  and the number of isomorphic bags is exponential in the size of  $\Sigma$ .

The above tree structure was introduced in [Benedikt *et al.*, 2012] 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., [Grädel *et al.*, 2002]) that is empty iff no instance of  $chase(D, \Sigma)$  satisfies  $q$ . This tree automaton represents the tree structure of the instances of  $chase(D, \Sigma)$  that do not satisfy  $q$ . Moreover, the size of the automaton is exponential in  $D$  and  $\Sigma$  due to the fact that, as shown in [Benedikt *et al.*, 2012], for every instance  $I \in chase(D, \Sigma)$  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.  $\square$

Another formalism with a lower combined complexity is the class of *full-identity* DIDs, that is, rules of the form  $r(X_1, \dots, X_n) \rightarrow \bigvee_{i=1}^m p_i(X_1, \dots, X_n)$  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 permute 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 [Calvanese *et al.*, 2006], 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 observed in [Gottlob *et al.*, 2012], guarded DTGDs are strictly more expressive than  $\mathcal{ELU}$  [Baader *et al.*, 2005], 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 contain 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  $DL-Lite_{bool}^{\mathcal{H}}$  [Artale *et al.*, 2009], 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  $DL-Lite_{bool}^{\mathcal{H}}$ . Let us recall that this logical language contains object names  $a_0, a_1, \dots$ , concept names  $A_0, A_1, \dots$ , and role names  $P_0, P_1, \dots$ . Complex roles  $R$  and concepts  $C$  are defined as follows:  $R ::= P_k | P_k^-$ ,  $B ::= \perp | A_k | \exists R$  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  $\tau$ .

**Expressive Power.** We establish that *frontier-guarded*[ $\perp$ ] DTGDs, that is, the formalism obtained by combining frontier-guarded DTGDs with *negative constraints* (NCs) of the form  $\forall \mathbf{X} \varphi(\mathbf{X}) \rightarrow \perp$ , where  $\varphi$  is a conjunction of atoms without any syntactic restrictions, are strictly more expressive than  $DL-Lite_{bool}^{\mathcal{H}}$ . Notice that the complexity of query answering under frontier-guarded[ $\perp$ ] 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., [Calì *et al.*, 2012a].

It is easy to see that role inclusions are translated by  $\tau$  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[ $\perp$ ] DTGDs in polynomial time.

**Lemma 9** *For every concept inclusion  $\alpha = C \sqsubseteq D$ , a set  $\Sigma_\alpha$  of frontier-guarded[ $\perp$ ] DTGDs can be constructed in polynomial time such that  $\tau(C \sqsubseteq D)$  and  $\Sigma_\alpha$  are equisatisfiable.*

**Proof (idea).** 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  $\varphi$  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 fix-point 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.  $\square$

Lemma 9 implies that, if  $\mathcal{T}_C$  are the concept inclusions of  $\mathcal{T}$ , then  $\Sigma_C = \bigcup_{\alpha \in \mathcal{T}_C} \Sigma_\alpha$  is a set of frontier-guarded[ $\perp$ ] DTGDs which can be constructed in polynomial time, and it is equisatisfiable with  $\mathcal{T}_C$ . Consequently,  $\tau(\mathcal{T}_R) \cup \Sigma_C$  is a set of frontier-guarded[ $\perp$ ] DTGDs which is equisatisfiable with  $\mathcal{T}$ . Since the simple rule  $s(X) \rightarrow r(X, X)$  is not expressible in  $DL-Lite_{bool}^{\mathcal{H}}$ , the next result follows.

**Theorem 10** *Frontier-guarded[ $\perp$ ] DTGDs are strictly more expressive than  $DL-Lite_{bool}^{\mathcal{H}}$ .*

**Complexity Results.** By exploiting the above construction, query answering under  $DL-Lite_{bool}^{\mathcal{H}}$  can be reduced in polynomial time to CQAns under frontier-guarded[ $\perp$ ] 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  $\tau$ . A TBox  $\mathcal{T}$  together with  $\mathcal{A}$  constitute a *knowledge base* (KB)  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ .

**Lemma 11** *UCQAns under  $DL-Lite_{bool}^{\mathcal{H}}$  knowledge bases can be reduced in polynomial time to UCQAns under frontier-guarded[ $\perp$ ] DTGDs.*

**Proof (sketch).** Consider a  $DL-Lite_{bool}^{\mathcal{H}}$  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  $A_-(a)$  and  $R_-(a, b)$ , respectively, where  $A_-$  and  $R_-$  are auxiliary predicates. Let  $\Sigma_{\mathcal{T}} = \tau(\mathcal{T}_R) \cup \Sigma_C \cup \Sigma_{\perp}$ , where  $\Sigma_{\perp}$  contains an NC  $A(X)$ ,  $A_-(X) \rightarrow \perp$  and  $R(X, Y)$ ,  $R_-(X, Y) \rightarrow \perp$ , 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[ $\perp$ ] DTGDs that can be constructed in polynomial time, the claim follows.  $\square$

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 12** *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 [Artale *et al.*, 2009] it was shown that query answering under  $DL-Lite_{bool}^{\mathcal{H}}$  is coNP-complete in data complexity; however, the combined complexity was not investigated and left as an open problem. Since  $DL-Lite_{bool}^{\mathcal{H}}$  is a description logic equipped with limited existential quantification, role inverse and union, Theorems 2 and 12, together with Lemma 11, imply the next complexity result.

**Corollary 13** *CQAns under  $DL-Lite_{bool}^{\mathcal{H}}$  knowledge bases is 2EXPTIME-complete in combined complexity.*

Interestingly, the above corollary significantly strengthens a similar result for the  $\mathcal{ALCT}$  DL in [Lutz, 2008].

## 7 Conclusion

In this paper, we studied in depth 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 investigate the impact of the addition of disjunction on non-guarded-based classes of TGDs.

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

## References

- [Alviano *et al.*, 2012] Mario Alviano, Wolfgang Faber, Nicola Leone, and Marco Manna. Disjunctive Datalog with existential quantifiers: Semantics, decidability, and complexity issues. *TPLP*, 12(4-5):701–718, 2012.
- [Andréka *et al.*, 1998] Hajnal Andréka, Johan van Benthem, and István Németi. Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic*, 27:217–274, 1998.
- [Artale *et al.*, 2009] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyashev. The DL-Lite family and relations. *J. Artificial Intelligence Res.*, 36:1–69, September 2009.
- [Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *Proc. of IJCAI*, pages 364–369, 2005.
- [Baader, 2003] Franz Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. of IJCAI*, pages 319–324, 2003.
- [Baget *et al.*, 2011a] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
- [Baget *et al.*, 2011b] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proc. of IJCAI*, pages 712–717, 2011.
- [Bárány *et al.*, 2010] V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *Proc. of LICS*, pages 1–10, 2010.
- [Bárány *et al.*, 2011] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. In *Proc. of ICALP*, pages 356–367, 2011.
- [Bárány *et al.*, 2012] Vince Bárány, Balder ten Cate, and Martin Otto. Queries with guarded negation. *PVLDB*, 5(11):1328–1339, 2012.
- [Beeri and Vardi, 1981] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In *Proc. of ICALP*, pages 73–85, 1981.
- [Benedikt *et al.*, 2012] Michael Benedikt, Pierre Bourhis, and Pierre Senellart. Monadic Datalog containment. In *Proc. of ICALP*, pages 79–91, 2012.
- [Björklund *et al.*, 2008] Henrik Björklund, Wim Martens, and Thomas Schwentick. Optimizing conjunctive queries over trees using schema information. In *Proc. of MFCS*, pages 132–143, 2008.
- [Calì *et al.*, 2008] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In *Proc. of KR*, pages 70–80, 2008.
- [Calì *et al.*, 2012a] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.
- [Calì *et al.*, 2012b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, 193:87–128, 2012.
- [Calvanese *et al.*, 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of KR*, pages 260–270, 2006.
- [Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
- [Deutsch and Tannen, 2003] Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In *Proc. of ICDT*, pages 225–241, 2003.
- [Deutsch *et al.*, 2008] Alin Deutsch, Alan Nash, and Jeff B. Remmel. The chase revisited. In *Proc. of PODS*, pages 149–158, 2008.
- [Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive Datalog. *ACM Trans. Database Syst.*, 22(3):364–418, 1997.
- [Fagin *et al.*, 2005] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: Semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
- [Gottlob *et al.*, 2012] Georg Gottlob, Marco Manna, Michael Morak, and Andreas Pieris. On the complexity of ontological reasoning under disjunctive existential rules. In *Proc. of MFCS*, pages 1–18, 2012.
- [Grädel *et al.*, 2002] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*. Springer, 2002.
- [Krötzsch and Rudolph, 2011] Markus Krötzsch and Sebastian Rudolph. Extending decidable existential rules by joining acyclicity and guardedness. In *Proc. of IJCAI*, pages 963–968, 2011.
- [Leone *et al.*, 2012] Nicola Leone, Marco Manna, Giorgio Terracina, and Pierfrancesco Veltri. Efficiently computable Datalog<sup>∃</sup> programs. In *Proc. of KR*, 2012.
- [Lutz, 2008] C. Lutz. The complexity of conjunctive query answering in expressive description logics. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Automated Reasoning*, volume 5195 of *LNCS*, pages 179–193. Springer Berlin / Heidelberg, 2008.
- [Patel-Schneider and Horrocks, 2007] Peter F. Patel-Schneider and Ian Horrocks. A comparison of two modelling paradigms in the semantic web. *J. Web Semantics*, 5(4):240–250, 2007.
- [Thomazo *et al.*, 2012] Michaël Thomazo, Jean-François Baget, Marie-Laure Mugnier, and Sebastian Rudolph. A generic querying algorithm for greedy sets of existential rules. In *Proc. of KR*, 2012.