Walking the Complexity Lines for Generalized Guarded Existential Rules

Jean-François Baget
INRIA
France
baget@lirmm.fr

Marie-Laure Mugnier
Univ. Montpellier II
France
mugnier@lirmm.fr

Sebastian Rudolph
KIT
Germany
rudolph@kit.edu

Michaël Thomazo
Univ. Montpellier II
France
thomazo@lirmm.fr

Abstract

We establish complexities of the conjunctive query entailment problem for classes of existential rules (*i.e.* Tuple-Generating Dependencies or Datalog+/- rules). Our contribution is twofold. First, we introduce the class of *greedy bounded treewidth sets (gbts)*, which covers guarded rules, and their known generalizations, namely (weakly) frontier-guarded rules. We provide a generic algorithm for query entailment with *gbts*, which is worst-case optimal for combined complexity with bounded predicate arity, as well as for data complexity. Second, we classify several *gbts* classes, whose complexity was unknown, namely frontier-one, frontier-guarded and weakly frontier-guarded rules, with respect to combined complexity (with bounded and unbounded predicate arity) and data complexity.

1 Introduction

First-order Horn rules (without function symbols except constants) have long been used in artificial intelligence, as well as in databases under name Datalog. We consider here an extension of these rules that allows to create existentially quantified variables. More precisely, these extended rules are of the form *Body* → *Head*, where *Body* and *Head* are conjunctions of atoms, and variables occurring only in the *Head* are existentially quantified. E.g., $\forall x(Human(x) \rightarrow \exists y(Parent(y, x) \wedge Human(y)))$. Such rules are known in databases as Tuple-Generating Dependencies (TGDs) and have been extensively used, e.g. for data exchange [Fagin *et al.*, 2005]. Recently, the corresponding logical fragment has gained new interest in the context of ontological knowledge representation. It has been introduced as the Datalog+/- framework in [Calì *et al.*, 2008; Calì *et al.*, 2009; Calì *et al.*, 2010], and independently, stemming from graph-based knowledge representation formalisms [Chein and Mugnier, 2009], as ∀∃-rules [Baget *et al.*, 2009; Baget *et al.*, 2010]. This rule-based framework is particularly well-suited to the topical ontological query answering problem, which consists of querying data while taking ontological knowledge into account. In particular, it generalizes the core of new description logics (DL) tailored for conjunctive query answering [Calì *et al.*, 2009; Baget *et al.*, 2010].

The ability to generate existential variables, associated with arbitrarily complex conjunctions of atoms, makes entailment with these rules undecidable [Beeri and Vardi, 1981; Chandra *et al.*, 1981]. Since the birth of TGDs, and recently within the Datalog+/- and ∀∃-rule frameworks, various conditions of decidability have been exhibited. Three "abstract" classes have been introduced in [Baget *et al.*, 2010] to describe known decidable behaviours: an obvious condition of decidability is the finiteness of the forward chaining (known as the *chase* in the TGD framework [Johnson and Klug, 1984]); sets of rules ensuring this condition are called *finite expansion sets* (*fes*); a more general condition introduced in [Calì *et al.*, 2008] accepts infinite forward chaining provided that the facts generated have a bounded treewidth (when seen as graphs); such sets of rules are called bounded treewidth sets (*bts*); then decidability follows from the decidability of first-order logic (FOL) classes with the bounded treewidth model property [Courcelle, 1990]. The third condition, giving rise to *finite unification sets* (*fus*), relies on the finiteness of (a kind of) backward chaining mechanism. None of these three abstract classes is recognizable, i.e., the associated membership problem is undecidable [Baget *et al.*, 2010].

In this paper, we focus on the *bts* paradigm and its main "concrete" classes. (Pure) *Datalog* rules (*i.e.* without existential variables) are *fes* (thus *bts*). *Guarded* rules [Calì *et al.*, 2008] are inspired by the guarded fragment of FOL. Their body has an atom (the guard) that contains all variables from the body. Guarded rules are *bts* (and not *fes*), they are generalized by *weakly guarded rules* (wg), in which the guarding condition is relaxed: only "affected" variables need to be guarded; intuitively, affected variables are variables that are possibly mapped, during the forward chaining process, to newly created variables [Calì *et al.*, 2008]. *wg*-rules include Datalog rules (in which there are no affected variables). Other decidable classes rely on the notion of the *frontier* of a rule (the set of variables shared between the body and the head of a rule). In a *frontier-one* rule *(fr1)*, the frontier is restricted to a single variable [Baget *et al.*, 2009]. In a frontier-guarded rule *(fg)*, an atom in the body guards the frontier [Baget *et al.*, 2010]. Hence, *fg*-rules generalize both *guarded* rules and *fr1*-rules. When only affected variables from the frontier need to be guarded, we obtain the still decidable class of *weakly frontier guarded rules (wfg)*, which generalizes both *fg* and *wg* classes [Baget *et al.*, 2010]. Of all known recognizable *bts* classes, *wfg* is the class subsuming the most of the others.

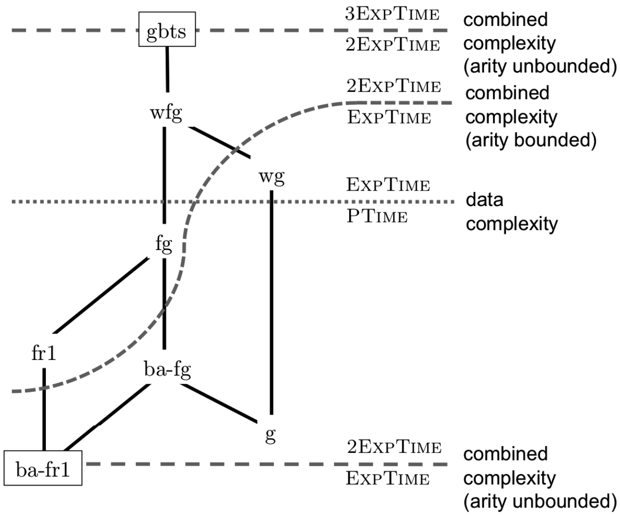

Figure 1: Complexity Boundaries. Tight bounds for gbts and ba-fr1 are yet unknown, we conjecture 2ExpTime-completeness and ExpTime-completeness, respectively.

Example 1 $\mathit{Parent}(x, y), \mathit{Parent}(y, z) \rightarrow \mathit{isGdParent}(x)$ (with simplified syntax) is Datalog and *fr1* but not guarded; while $\mathit{WorksOn}(x, z), \mathit{WorksOn}(y, z), \mathit{StudentTandem}(x, y) \rightarrow \mathit{Grade}(x, t), \mathit{Grade}(y, t)$ is *fg*, but is neither *fr1*, nor guarded, nor Datalog.

Contrarily to *fes* and *fus*, the definition of *bts* does not provide a constructive entailment procedure. Some of its subclasses, namely guarded and *wg*, are provided with an algorithm and their complexity is known [Calì *et al.*, 2008; Calì *et al.*, 2009]. However, this is not the case for the *fr1*, *fg* and *wfg* classes. The aim of this paper is to solve these algorithmic and complexity issues.

Our contribution is twofold. First, by imposing a restriction on the allowed derivation sequences, we define a subclass of *bts*, namely greedy *bts (gbts)*, which has the nice property of covering the *wfg* class (thus all *bts* classes cited above). We provide a generic algorithm for this class, which is worst-case optimal for data complexity, as well as for combined complexity in the case where predicate arity is bounded. Second, we classify the *wfg*, *fg* and *fr1* classes with respect to both combined (with and without bound on the predicate arity) and data complexities. We also consider the case of rules with a hypergraph-acyclic body (notation *ba*) and point out that body-acyclic fg-rules coincide with guarded rules from an expressivity and complexity perspective.

Fig. 1 shows the complexity lines for these classes of rules with three complexity measures, i.e., combined complexity without or with bound on the predicate arity, and data complexity. Notice that data complexity and bounded-arity combined complexity are not strictly layered. While *fg*-rules are much easier for data complexity (PTime) than for bounded-arity combined complexity (2ExpTime), wg-rules are in ExpTime for both. Precise complexity results obtained are given in Tab. 1. New results are indicated by a star. Detailed proofs can be found in the accompanying research report [Baget *et al.*, 2011].

Class	arity unbounded	arity bounded	Data Complexity
gbts	in 3ExpTime $\star$	2ExpTime-c $\star$	ExpTime-c$\star$
wfg	2ExpTime-c $\star$	2ExpTime-c $\star$	ExpTime-c$\star$
fg	2ExpTime-c $\star$	2ExpTime-c $\star$	PTime-c $\star$
fr1	2ExpTime-c $\star$	2ExpTime-c $\star$	PTime-c $\star$
wg	2ExpTime-c	ExpTime-c	ExpTime-c
guarded	2ExpTime-c	ExpTime-c	PTime-c
ba-fg	2ExpTime-c $\star$	ExpTime-c $\star$	PTime-c $\star$
ba-fr1	ExpTime-hard $\star^{(1)}$	ExpTime-c $\star$	PTime-c $\star$

(1) ExpTime-c if no constants in rules

Table 1: Combined and Data Complexities

2 Preliminaries

As usual, an *atom* is of the form $p(t_1, \ldots, t_k)$ where p is a predicate with arity k, and the t_i are terms, *i.e.* variables or constants. A *conjunct* $C[\mathbf{x}]$ is a finite conjunction of atoms, where $\mathbf{x}$ is the set of variables occurring in C. A *fact* is the existential closure of a conjunct.[1] A (boolean) *conjunctive query* (CQ) has the same form as a fact, thus we identify both notions. We also see conjuncts, facts and CQ as sets of atoms. Given an atom or a set of atoms A, $vars(A)$ and $terms(A)$ denote its set of variables and of terms, respectively. Given conjuncts F and Q, a *homomorphism* π from Q to F is a substitution of $vars(Q)$ by terms of F such that $\pi(Q) \subseteq F$ (we say that π maps Q to F). It is well-known that, given two facts F and Q, $F \models Q$ iff there is a homomorphism from Q to F.

Definition 1 (∀∃-Rule) *A* ∀∃-rule *(*existential rule*, or simply* rule *when not ambiguous) is a formula* $R = \forall \mathbf{x} \forall \mathbf{y} (B[\mathbf{x}, \mathbf{y}] \rightarrow (\exists \mathbf{z} H[\mathbf{y}, \mathbf{z}]))$ *where* $B = body(R)$ *and* $H = head(R)$ *are conjuncts, resp. called the* body *and the* head *of R. The* frontier *of R, noted* $fr(R)$*, is the set of variables* $vars(B) \cap vars(H) = \mathbf{y}$.

Definition 2 (Application of a Rule) *A rule R is* applicable *to a fact F if there is a homomorphism* π *from* $body(R)$ *to F; the result of the* application of R on F w.r.t. π *is a fact* $\alpha(F, R, \pi) = F \cup \pi^{safe}(head(R))$ *where* π^{safe} *is a substitution of* $head(R)$*, which replaces each* $x \in fr(R)$ *with* $\pi(x)$*, and other variables with fresh variables. As* α *only depends on* $\pi_{|fr(R)}$ *(the restriction of* π *to* $fr(R)$*), we also write* $\alpha(F, R, \pi_{|fr(R)})$.

Definition 3 (Derivation) *Let F be a fact, and* $\mathcal{R}$ *be a set of rules. An* $\mathcal{R}$*-derivation of F is a finite sequence* $(F_0 = F), \ldots, F_k$ *s.t. for all* $0 \leq i < k$*, there is* $R_i \in \mathcal{R}$ *and a homomorphism* π_i *from* $body(R_i)$ *to* F_i *s.t.* $F_{i+1} = \alpha(F_i, R_i, \pi_i)$.

Theorem 1 (Forward Chaining) *Let F and Q be two facts, and* $\mathcal{R}$ *be a set of rules. Then* $F, \mathcal{R} \models Q$ *iff there exists an* $\mathcal{R}$*-derivation* $(F_0 = F), \ldots, F_k$ *such that* $F_k \models Q$.

A *knowledge base* (KB) $\mathcal{K} = (F, \mathcal{R})$ is composed of a finite set of facts (seen as a single fact) F and a finite set of rules $\mathcal{R}$. We denote by C the set of constants occurring in $(F, \mathcal{R})$ and by $\mathcal{P}$ the set of predicates occurring in $\mathcal{R}$. The *(boolean) CQ*

[1] Note that thereby we generalize the traditional notion of a fact in order to take existentially quantified variables into account.

entailment problem is the following: given a KB $\mathcal{K} = (F, \mathcal{R})$ and a (boolean) CQ Q, does $F, \mathcal{R} \models Q$ hold ?

We now specify some already introduced notions. A fact can naturally be seen as a hypergraph whose nodes are the terms in the fact and whose hyperedges encode atoms. The primal graph of this hypergraph has the same set of nodes and there is an edge between two nodes if they belong to the same hyperedge. The treewidth of a fact is defined as the treewidth[2] of its associated primal graph. A set of rules $\mathcal{R}$ is called a *bounded treewidth set (*bts*)* if for any fact F there exists an integer b such that, for any fact F' that can be $\mathcal{R}$-derived from F, $\mathit{treewidth}(F') \leq b$. A rule R is guarded if there is an atom $a \in body(R)$ (called a guard) with $vars(body(R)) \subseteq vars(a)$. R is weakly guarded *(wg)* if there is $a \in body(R)$ (called a weak guard) that contains all affected variables from $body(R)$. The notion of affected variable is relative to the rule set: a variable is affected if it occurs only in affected predicate positions, which are positions that may contain a new variable generated by forward chaining (see [Fagin *et al.*, 2005] for a precise definition). The only property of affected variables used in this paper is that a rule application necessarily maps non-affected variables only to terms from the initial fact F. R is frontier-one *(fr1)* if $|fr(R)| = 1$. R is frontier-guarded *(fg)* if there is $a \in body(R)$ with $vars(fr(R)) \subseteq vars(a)$. R is weakly-frontier guarded *(wfg)* if there is $a \in body(R)$ that contains all affected variables from $fr(R)$.

3 Greedy Bounded-Treewidth Sets of Rules

In a greedy derivation, every rule application maps the frontier of the rule into terms added by a *single* previous rule application or occurring in the initial fact:

Definition 4 (Greedy Derivation) *An $\mathcal{R}$-derivation $(F_0 = F), \ldots, F_k$ is said to be* greedy *if, for all i with $0 \leq i < k$, there is $j < i$ such that $\pi_i(fr(R_i)) \subseteq vars(A_j) \cup vars(F_0) \cup C$, where $A_j = \pi_j^{safe}(head(R_j))$.*

Example 2. Let $\mathcal{R} = \{R_0, R_1\}$ where: $R_0 = r_1(x,y) \rightarrow r_2(y,z)$ and $R_1 = r_1(x,y), r_2(x,z), r_2(y,t) \rightarrow r_2(z,t)$.

Let $F_0 = \{r_1(a,b), r_1(b,c)\}$ and $S = F_0, \ldots, F_3$ with $F_1 = \alpha(F_0, R_0, \{(y,b)\}), A_0 = \{r_2(b,x_1)\}$, $F_2 = \alpha(F_1, R_0, \{(y,c)\})$, $A_1 = \{r_2(c,x_2)\}$, $F_3 = \alpha(F_2, R_1, \pi_2)$, with $\pi_2 = \{(z,x_1),(t,x_2)\}$; there is no A_j s.t. $\{\pi_2(z), \pi_2(t)\} \subseteq terms(A_j)$, thus S is not greedy.

To any greedy derivation S of F can be assigned a unique *derivation tree* $DT(S)$ built iteratively as follows: the root is a node x_0 with $terms(x_0) = vars(F) \cup C$ and $atoms(x_0) = atoms(F)$, and $\forall 0 \leq i < k$, we add a node x_{i+1} with $terms(x_{i+1}) = vars(A_{i+1}) \cup vars(F) \cup C$ and $atoms(x_{i+1}) = atoms(A_{i+1})$. Since S is greedy, there is at least a j such that $\pi_i(fr(R_i)) \subseteq terms(x_j)$. We add an edge between x_{i+1} and $x_{j'}$, j' being the smallest having this property. The nodes of $DT(S)$ are also called "bags".

Property 2 *Let $S = F_0 \ldots, F_k$ be a greedy derivation. Then $DT(S)$ is a tree decomposition of F_k of width bounded by $|vars(F)| + |C| + max(|vars(head(R))|_{R \in \mathcal{R}})$.*

[2] We assume that the reader is familiar with this notion.

Definition 5 (greedy bounded-treewidth set of rules (gbts)) *$\mathcal{R}$ is said to be a greedy bounded-treewidth set (gbts) if (for any fact F) any $\mathcal{R}$-derivation (of F) is greedy.*

The class *gbts* is a strict subclass of *bts* (e.g. in Example 2: $\mathcal{R}$ is *fes* but not *gbts*). It is nevertheless an expressive subclass of *bts* since it contains *wfg*:

Property 3 *wfg-rules are gbts.*

Proof: Let R be a rule from a *wfg* rule set $\mathcal{R}$ and let g be a weak frontier guard of R. If R is applied by π and let $\pi(g) = a$, then $a \in F$ or $a \in A_i$ for some i. Thus $\mathcal{R}$ is gbts. □

Note that *gbts* strictly contains *wfg*. Indeed, $\{R\}$ is *gbts*, but not *wfg* (nor *fes*), with $R = r_1(x,y), r_2(y,z) \rightarrow r(x,x'), r(y,y'), r(z,z'), r_1(x',y'), r_2(y',z')$.

4 An algorithm for gbts

Greediness ensures that, at each step of a derivation, the current derivation tree can be extended to a bounded-width tree decomposition of any to-be-derived fact. It follows that rule applications create only a bounded number of "relevant patterns". When we have created all possible patterns, "large-enough" to map Q, we can halt the derivation process. In [Calì *et al.*, 2009], a specific notion of *type* is used for that purpose. However, to take non-guarded rules into account, we need to generalize it. We thus define the ultimate applicability of a rule, and the related notion of an oracle. Sets of oracles generalize types. To simplify the presentation, we translate Q into a rule $R_Q = Q \rightarrow \mathit{match}$ where match is a fresh nullary predicate (note that $fr(R_Q)$ is empty). The question is now $\mathcal{R} \cup \{R_Q\} \models \mathit{match}$.

Definition 6 (Ultimate applicability, oracle) *Let F be a fact, $\mathcal{R}$ be a set of gbts rules, and $S = F_0 (= F), \ldots, F_{i+1}$ be an $\mathcal{R}$-derivation of F (with $F_{j+1} = F_j \cup A_j, \forall 0 \leq j \leq i$), and let x_j be the bag of $DT(S)$ associated with A_j. We say that $R \in \mathcal{R}$ is* ultimately applicable *to x_j if there is an $\mathcal{R}$-derivation $F_0 (= F), \ldots, F_j, \ldots, F_{i+1}, \ldots, F_l$ with a homomorphism π from $body(R)$ to F_l and $\pi(fr(R)) \subseteq vars(A_j) \cup vars(F) \cup C$. We say that $\pi_{|fr(R)}$ is an* oracle *for the ultimate applicability of R on x_j. An* ultimate $\mathcal{R}$-derivation *is a sequence $F_0, \ldots, F_k$ where $\forall 0 < i < k$, there is $R \in \mathcal{R}$ and an oracle $\pi_{fr(R)}$ for the ultimate applicability of R on some x_j with $j < i$ such that $F_i = \alpha(F_{i-1}, R, \pi_{|fr(R)})$.*

Any derivation is an ultimate derivation and an ultimate derivation $F_0, F_1, \ldots, F_k$ can always be extended to a derivation $F_0, F_0^1, \ldots, F_0^{i_1}, F'_1, \ldots, F'_{k-1}, F_{k-1}^1, \ldots, F_{k-1}^{i_k}, F'_k$, where F'_i contains F_i.

We now define an equivalence relation $\sim_Q$ on the bags of the derivation tree with the following informal meaning: $x_i \sim_Q x_j$ means that a rule body can be ultimately mapped in x_i iff if can be mapped similarly in x_j.

Definition 7 ($\sim_Q$) *Let T be a derivation tree, x and y two bags of T. $x \sim_Q y$ if there is a bijective substitution ψ of $terms(x)$ by $terms(y)$ s.t. $\forall R \in \mathcal{R}$, π is an oracle for R on x iff $\psi \circ \pi$ is an oracle of R on y.*

Algorithm 1 behaves as a classical breadth-first forward chaining (also called "chase" in databases) with two main differences. First, instead of looking for homomorphisms to check the applicability of rules, it uses oracles for ultimate applicability, thus building an ultimate derivation tree. Second, $\sim_Q$ allows to prune the ultimate derivation tree.[3]

Algorithm 1: Ultimate Saturation

```
Data: Two facts F, Q,gbts rule set R
Result: YES if F,R ⊨ Q, NO otherwise.
R ← R ∪ {R_Q};
T ← newTree(x_0) ;                                   // x_0 is the root
terms(x_0) ← vars(F) ∪ C;
Continue ← True;  Depth ← 0 ;                        // depth of T
while Continue do
    Continue ← False;
    for x ∈ leavesAtDepth(T, Depth) do
        for R ∈ R do
            for π ∈oracles(R, x)\oracles(R, parent(x)) do
                if R = R_Q then
                    return YES;
                y ← newNode();
                terms(y)←vars(π^safe(head(R)))∪vars(F)∪C;
                if ¬∃z ∈ nodes(T) such that y ~_Q z then
                    addEdge(T, (x, y));
                    Continue ← True;
    Depth++;
return NO;
```

Theorem 4 *Ultimate Saturation is sound and complete for CQ entailment with $\mathcal{R}$ gbts.*

Proof: (sketch). Without the $\sim_Q$ check, soundness and completeness would follow from the correspondence between standard and ultimate derivations. To prove that the pruning performed by the $\sim_Q$ check does not prevent completeness, we show that: "if $x \sim_Q y$ and x parent of x', then there is a child y' of y s.t. $x' \sim_Q y'$". We first build $y' = \mathit{copy}_{x \leftarrow y}(x')$ the node obtained from y "as x' is obtained from x". Obviously $x'' = \mathit{copy}_{y \leftarrow x}(y') \sim_Q x'$. It remains to prove that if π is an oracle of $R = (B, H)$ on x', then it is also an oracle of R on y'. For that, we generalize the *copy* notion: instead of copying a child of a node x under a $\sim_Q$ equivalent node y, we *copy* a whole subtree of x under y. We point out that if π is an oracle of R on x' (and not on x), then there is a finite subtree $T_{x'}$ rooted in x' s.t. B is mapped to the fact associated with a minimal derivation generating $T_{x'}$. Finally, we show that B is also mapped to the fact associated with a minimal derivation generating the *copy* of $T_{x'}$ rooted in the $\sim_Q$ equivalent node y. The proof is by induction on the depth of $T_{x'}$. □

Let us focus on the complexity of Algorithm 1. If T_{max} denotes the maximum number of non-equivalent nodes that can be generated, the complexity of Algorithm 1 is $O(T_{max} \times |\mathcal{R}| \times [(\text{cost of a call to oracles } (R, x)) + |\text{oracles}(R, x)| \times (\text{cost of checking the existence of a } \sim_Q \text{ equivalent node})])$.

Property 5 *Let $\mathcal{R}$ be gbts, Q a query, F a fact. Let $q = \max(|\mathrm{terms}(\mathrm{body}(R_i))|)$, $b = |\mathrm{terms}(F)| + \max_i(|\mathrm{terms}(\mathrm{head}(R_i))|) + |C|$ and w the maximum predicate arity. Let S be an ultimate derivation of F, x a bag of $DT(S)$. The cost of a call to $\mathrm{oracles}(R, x)$ is in the order of $\mathrm{poly}(b^b, 2^{b^{q+1} 2^{|\mathcal{P}|.q^{w+1}}})$.*

[3]This is necessary to guarantee termination, resembling the *blocking* techniques applied in DL tableaux algorithms.

Proof: See [Baget *et al.*, 2011]. □

Checking $y \sim_Q z$ has a cost $O(b^b \times (|\mathcal{R}| \times q^b)^2)$ and T_{max} is upper-bounded by $2^{|\mathcal{R}| \times b^q}$, hence the following theorem (hardness results stemming from *wfg* subclass):

Theorem 6 *CQ entailment for gbts is in* 3ExpTime *for combined complexity,* 2ExpTime*-complete for predicate with bounded arity and* ExpTime*-complete for data complexity.*

5 Weakly Frontier-Guarded Rules

First, the ExpTime-complete data complexity of wfg-rules directly follows from ExpTime membership of gbts (Sect. 4) and ExpTime-hardness of wg-rules [Calì *et al.*, 2008].

We now prove that w(f)g-rules can be polynomially translated into (f)g-rules. In particular, this allows us to exploit the 2ExpTime membership result established in the next section for fg-rules. W.l.o.g. we assume here that the initial fact F does not contain any variable. Then, a homomorphism from a rule body to a derived fact necessarily maps non-affected variables to constants in C. Thus, by replacing non-affected variables in rules with all possible constants, we obtain an equivalent set of rules. However, this partial grounding produces a worst-case exponential blow-up in the number of non-affected variables per rule. We thus provide a way to simulate partial groundings with only polynomial blow-up.

Let $\mathsf{nav}(R)$ denote the non-affected variables in $R \in \mathcal{R}$. For convenience, we fix bijections $\#_R : \mathsf{nav}(R) \to \{1, ..., |\mathsf{nav}(R)|\}$ which for every R, assign numbers to all the non-affected variables. Now let $v_1, \ldots, v_m$ and $v'_1, \ldots, v'_s$ be variable symbols not used in $\mathcal{R}$, where $m = |C|$ and $s = \max_{R \in \mathcal{R}} |\mathsf{nav}(R)|$. We now define the function τ, mapping rules from $\mathcal{R}$ to fg-rules as follows: For terms t, let $\tau_R(t) = v'_{\#_R(t)}$ if $t \in \mathsf{nav}(R)$ and $\tau_R(t) = t$ otherwise. We extend τ_R to atoms by letting $\tau_R\big(p(t_1, ..., t_l)\big) = p(v_1, ..., v_m, v'_1, ..., v'_s, \tau_R(t_1), ..., \tau_R(t_l))$, which is lifted to sets of atoms in the usual way. Finally, for a rule $R : \mathit{body}(R) \to \mathit{head}(R)$, we let $\tau(R) = \tau_R(B) \to \tau_R(H)$

Note that thereby the arity of all predicates is increased by $m + s$. The first m positions will be used to permanently hold all constants $c_1, \ldots, c_m$ and the next s positions will serve as a pool for special non-affected variables which will be used for our implicit grounding. Now we let $\tau(\mathcal{R}) := \{\tau(R) \mid R \in \mathcal{R}\} \cup \mathcal{S}$, where $\mathcal{S}$ contains for every predicate $p \in \mathcal{P}$ (let its arity be l) the rules $S^p_{c_i \mapsto v'_j}$

$$p(v_1, .., v_m, v'_1, .., v'_s, x_1, .., x_l) \to p(v_1, .., v_m, v^*_1, .., v^*_s, x_1, .., x_l)$$

where $v^*_j = v_i$ and $v^*_k = v'_k$ for all $k \neq j$. Thereby, the rule $S_{c_i \mapsto v'_j}$ is used to realize the bindings of the constant c_i to the non-affected variable v'_j. Now let $\tau(F) = \{\tau'(a) \mid a \in F\}$ and $\tau(Q) = \{\tau'(a) \mid a \in Q\}$ where

$$\tau'(p(e_1, \ldots, e_l)) = p(c_1, \ldots, c_m, \underbrace{c_1, \ldots, c_1}_{s}, e_1, \ldots, e_l).$$

Note that the choice of c_1 at positions $m + 1, \ldots, m + s$ is just an arbitrary one; the rules from $\mathcal{S}$ allow to "put" any combination of constants at these positions. We have arrived at a translation that suits our needs.

Theorem 7 *Any instance of CQ entailment with w(f)g-rules can be polynomially translated into an instance of the same problem with (f)g-rules.*

Proof: The size of $\tau(F)$ (resp. $\tau(\mathcal{R})$, $\tau(Q)$) is polynomially bounded by the combined size of F and $\mathcal{R}$. By adding $\mathsf{nav}(R)$ to each atom in R, τ transforms each weak guard into a guard. The rules from $\mathcal{S}$ are guarded. Hence, if $\mathcal{R}$ is w(f)g, then $\tau(\mathcal{R})$ is (f)g. It is easy to check that, given F, $\mathcal{R}$ (*wfg*) and Q, we have that $F, \mathcal{R} \models Q$ iff $\tau(F), \tau(\mathcal{R}) \models \tau(Q)$. □

6 Frontier-Guarded and Frontier-One Rules

In this section, we show that fg- and fr1-rules are both PTime-complete for data complexity and 2ExpTime-complete for combined complexity no matter whether predicate arity is bounded or not. Bárány *et al.* (2010) showed that deciding entailment of unions of boolean CQ in the guarded fragment (GF) of FOL is 2ExpTime-complete. This result can be used to prove the following theorem.

Theorem 8 *CQ entailment for fg-rules is in* 2ExpTime.

Proof: We observe that every fg-rule R can be translated into two rules one of which is guarded and the other is Datalog. Given the frontier guard $p(t_1, \ldots, t_n) \in \mathit{body}(R)$, we introduce a new n-ary predicate p_R and let $\mathit{separate}(R)$ be the set containing the two rules $S_R : \ \mathit{body}(R) \to p_R(t_1, \ldots, t_n)$ and $T_R : p_R(t_i, \ldots, t_n) \to \mathit{head}(R)$. It is immediate that for any fg-rule set $\mathcal{R}$, we have $F, \mathcal{R} \models Q$ exactly if $F, \bigcup_{R \in \mathcal{R}} \mathit{separate}(R) \models Q$.

Obviously, T_R is guarded (and hence also lies in GF). Now we transform S_R as follows (abbreviating $(t_1, \ldots, t_n)$ by $\mathbf{t}$ and $\mathit{vars}(\mathit{body}(R))$ by $\mathbf{x}$ as well as introducing a new predicate p'_R):

$$\begin{aligned} &\forall \mathbf{x}(\mathit{body}(R) \to p_R(\mathbf{t})) \ \Leftrightarrow \ \neg\exists \mathbf{x}(\mathit{body}(R) \wedge \neg p_R(\mathbf{t})) \\ \Leftrightarrow &\Big(\neg\exists \mathbf{x}(\mathit{body}(R) \wedge p'_R(\mathbf{t}))\Big) \wedge \Big(\forall \mathbf{t}(p(\mathbf{t}) \wedge \neg p_R(\mathbf{t}) \to p'_R(\mathbf{t}))\Big) \\ \Leftrightarrow &\underbrace{\Big(\neg\exists \mathbf{x}(\mathit{body}(R) \wedge p'_R(\mathbf{t}))\Big)}_{=:S'_R} \wedge \underbrace{\Big(\forall \mathbf{t}(p(\mathbf{t}) \to p'_R(\mathbf{t}) \vee p_R(\mathbf{t}))\Big)}_{=:S''_R} \end{aligned}$$

Hence $F, \mathcal{R} \models Q$ iff

$$F \cup \{T_R, S''_R \mid R \in \mathcal{R}\} \cup \{S'_R \mid R \in \mathcal{R}\} \models Q \qquad (\dagger)$$

where the first two sets are in GF and the third consists of negated existentially quantified conjuncts. Hence we can conceive every S'_R as a negated CQ $\neg Q_R$. Consequently we have

$$\{S'_R \mid R \in \mathcal{R}\} \equiv \{\neg Q_R \mid R \in \mathcal{R}\} \equiv \bigwedge_{R \in \mathcal{R}} \neg Q_R \equiv \neg \bigvee_{R \in \mathcal{R}} Q_R$$

which allows to rephrase (†) as

$$F \cup \{T_R, S''_R \mid R \in \mathcal{R}\} \models Q \vee \bigvee_{R \in \mathcal{R}} Q_R$$

leaving us with a GF theory on the lhs and a union of boolean CQ on the rhs. This translation is clearly linear. □

To prove the 2ExpTime-hardness of fr1-rules we adopt and adapt the construction used to show 2ExpTime-hardness for CQ entailment in the DL $\mathcal{ALCI}$ from Lutz (2007).

Theorem 9 *CQ entailment for fr1-rules with bounded predicate arity is* 2ExpTime*-hard.*

Proof: (Sketch) We show that Lutz' knowledge base representation of the halting problem for exponentially space-bounded alternating Turing machines can be expressed by fr1-rules by applying the following modifications: (1) the acceptance condition is encoded in a "backward-manner" as in the encoding used to prove ExpTime-hardness for Horn-$\mathcal{FLE}$ in [Krötzsch *et al.*, 2007] thus enabling to formulate acceptance for existential states without disjunction; (2) each negated concept $\neg A$ is replaced by a concept B defined to be disjoint with A, which is possible because the *tertium non datur* part of the negation is not needed; (3) the query is modified and turned into Horn rules: this ensures that information about configuration changes can be propagated forward making the use of disjunction obsolete. We obtain a CQ entailment problem w.r.t. a set of fr1-rules with at most binary predicates which encodes the halting problem, therefore showing the claim. □

The proof for PTime membership for data complexity is based on a specific locality property of derivations for fg-rules which is established in the following lemma.

Lemma 1 *For every constant-free fg-rule set $\mathcal{R}$ there exists a natural number $w_{\mathcal{R}}$ satisfying the following: Suppose $F, \mathcal{R} \models a$ for some F and an atom $a = p(z_1, \ldots, z_l)$ with $\mathsf{terms}(a) \subseteq \mathsf{terms}(F)$ and suppose that the corresponding derivation $F = F_0, \ldots, F_k$ is such that for every atom a' with $a' \in F_{k-1}$ and $\mathsf{terms}(a') \subseteq \mathsf{terms}(F)$ holds $a' \in F$. Then there is a set $V \subseteq \mathsf{terms}(F)$ with $|V| \leq w_{\mathcal{R}}$ such that $F|_V, \mathcal{R} \models a$ where $F|_V := \{b \mid b \in F, \mathsf{terms}(b) \subseteq V\}$.*

In order to leverage the above lemma for arbitrary fg-rule sets containing constants, we need to transform the task of deciding $F, \mathcal{R} \models Q$ into a setting where constants are excluded. The following definition and lemma provide for this by applying a partial grounding and subsequently shifting positions taken by constants into predicates.

Definition 8 *Let $\mathcal{R}$ be an arbitrary fg-rule set and let Q be a CQ. Let A be the set of constants occurring in $\mathcal{R}$ and Q. For every predicate p of arity k occurring in $\mathcal{R}$ and Q and every partial mapping $\gamma : \{1, \ldots, k\} \to A$, we let p_γ denote a new $(k - |dom(\gamma)|)$-ary predicate. Let ξ_A map atoms from $\mathcal{R}$ and Q to new atoms by projecting out positions filled by constants from A.[4] We lift the function ξ_A to conjuncts and rules in the obvious way. Now, letting $PG_A(R)$ denote all* partial groundings *of R where some universally quantified variables are substituted by constants from A, we define the rule set* $\mathsf{cfree}(\mathcal{R}, Q) = \left\{\xi_A(R') \mid R' \in PG_A(R), R \in \mathcal{R} \cup \{Q \to \mathit{match}\}\right\}$

Lemma 2 *For $\mathcal{R}$ (*fg*), $\mathsf{cfree}(\mathcal{R}, Q)$ is* fg *and constant-free. Given a fact F and assuming fixed $\mathcal{R}$ and Q, the size of $\xi_A(F)$ and the time to compute it is polynomially bounded by $|F|$. Moreover $F, \mathcal{R} \models Q$ iff $\xi_A(F), \mathsf{cfree}(\mathcal{R}, Q) \models \mathit{match}$.*

Relying on Lemma 1 we next provide a translation of constant-free $\mathcal{R}$ and Q into a Datalog program. The main idea is to "compile away" existential variables introduced in rule heads by "precomputing" deduction sequences that finally result in a query match.

Definition 9 *Given a constant-free fg-rule set $\mathcal{R}$, we define the Datalog-program $\mathbf{P}(\mathcal{R})$ as follows: Let $\{y_1, \ldots, y_{w_{\mathcal{R}}}\}$ be a set of variable symbols. Let $\mathbb{G}$ denote the finite set of all atoms with predicates from $\mathcal{R}$ and terms from $\{y_1, \ldots, y_{w_{\mathcal{R}}}\}$.*

[4] For instance $\xi_{\{a,b\}}(p(x, a, b, c)) = p_{\{2 \mapsto a, 3 \mapsto b\}}(x, c)$.

Now let $\mathbf{P}(\mathcal{R})$ *be the set of Datalog rules containing every* $\forall y_1, \ldots, y_{w_\mathcal{R}}(B \to h)$ *(with* $B \subseteq \mathbb{G}$ *and* $h \in \mathbb{G}$*) for which* $B, \mathcal{R} \models h$*.*

Then coupling constant removal and the preceding Datalog translation, we can establish the following proposition, which gives rise to the expected theorem.

Lemma 3 *For a set* $\mathcal{R}$ *of fg rules holds* $F, \mathcal{R} \models Q$ *iff* $\xi_A(F), \mathbf{P}(\text{cfree}(\mathcal{R}, Q)) \models$ *match.*

Theorem 10 *CQ entailment for fg- and fr1-rules is* PTime*-complete for data complexity.*

Proof: Thanks to Lemma 3, we have reduced the problem to atom entailment in Datalog. Noting that $\mathbf{P}(cfree(\mathcal{R}, Q))$ is independent from F and (w.l.o.g. assuming that F contains only constants) that $\xi_A(F)$ consists only of ground atoms, PTime data complexity membership follows from the PTime data complexity of entailment in Datalog [Dantsin *et al.*, 2001]. PTime-hardness for data complexity is a direct consequence of the same result for propositional Horn logic. □

Tree-like structures often lead to lower complexity. Hence, let us focus on fg-rules with an acyclic body *(ba)*, in the sense that the hypergraph associated with their body is acyclic.[5] First, note that guarded rules are trivially ba-fg-rules. In turn, a KB with ba-fg-rules can be polynomially translated into a KB with guarded rules while preserving the predicate arity.[6] Thus, previous complexity results on guarded rules apply to ba-fg-rules (in particular they are ExpTime-complete for bounded-arity combined complexity, while fg-rules are 2ExpTime-complete). Concerning ba-fr1-rules, PTime-complete data complexity follows from the proof of Th. 10; about combined complexity, ExpTime-hardness with bounded arity (thus with unbounded arity too) follows from the fact that standard reasoning in the weaker DL fragment Horn-$\mathcal{FLE}$ is already ExpTime-hard [Krötzsch *et al.*, 2007]; from ExpTime membership of guarded rules in the bounded arity case, we conclude that ba-fr1-rules are ExpTime-complete with bounded-arity. The only remaining question is whether they are simpler than ba-fg-rules in the unbounded arity case. We established ExpTime membership for the constant-free variant, but not for the general case.

7 Conclusion

We have introduced the notion of greedy bts of existential rules that subsumes guarded rules as well as their known generalizations and gives rise to a generic algorithm for deciding CQ entailment. Moreover, we have classified known gbts subclasses w.r.t. their combined and data complexities. Some interesting open issues remain, e.g. the exact complexity of gbts in the unbounded predicate arity case and the recognizability of gbts. We conjecture that the latter problem is decidable, however it can be shown that it is at least 2ExpTime-hard. Future work will aim at the integration of rules expressing equality and other properties such as transitivity into this framework, preserving decidability, and trying to keep the desirable PTime data complexity of fg-rules.

Acknowledgments. Sebastian Rudolph was supported by the ExpresST project funded by the German Research Foundation (DFG), as well as by LIRMM (Montpellier, France).

References

[Baget *et al.*, 2009] J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. Extending decidable cases for rules with existential variables. In *Proc. of IJCAI'09*, pages 677–682, 2009.

[Baget *et al.*, 2010] J.-F. Baget, M. Leclère, and M.-L. Mugnier. Walking the decidability line for rules with existential variables. In *Proc. of KR 2010*, 2010.

[Baget *et al.*, 2011] J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Complexity boundaries for generalized guarded existential rules. Research Report RR-11006, LIRMM, 2011.

[Bárány *et al.*, 2010] V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *Proc. of LICS 2010*, pages 1–10, 2010.

[Beeri and Vardi, 1981] C. Beeri and M. Vardi. The implication problem for data dependencies. In *Proc. of ICALP 1981*, volume 115 of *LNCS*, pages 73–85, 1981.

[Calì *et al.*, 2008] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In *Proc. of KR 2008*, pages 70–80, 2008.

[Calì *et al.*, 2009] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *Proc. of PODS 2009*, pages 77–86, 2009.

[Calì *et al.*, 2010] A. Calì, G. Gottlob, T. Lukasiewicz, B. Marnette, and A. Pieris. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proc. of LICS*, pages 228–242, 2010.

[Chandra *et al.*, 1981] A. K. Chandra, H. R. Lewis, and J. A. Makowsky. Embedded implicational dependencies and their inference problem. In *Proc. of STOC 1981*, pages 342–354, 1981.

[Chein and Mugnier, 2009] M. Chein and M.-L. Mugnier. *Graph-based Knowledge Representation and Reasoning—Computational Foundations of Conceptual Graphs*. Springer, 2009.

[Courcelle, 1990] B. Courcelle. The monadic second-order logic of graphs: I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

[Dantsin *et al.*, 2001] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comp. Surveys*, 33(3):374–425, 2001.

[Fagin *et al.*, 2005] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[Johnson and Klug, 1984] D.S. Johnson and A.C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.*, 28(1):167–189, 1984.

[Krötzsch *et al.*, 2007] M. Krötzsch, S. Rudolph, and P. Hitzler. Complexity boundaries for Horn description logics. In *Proc. of AAAI 2007*, pages 452–457, 2007.

[Lutz, 2007] C. Lutz. Inverse roles make conjunctive queries hard. In *Proc. of DL 2007*, 2007.

[5] A hypergraph is acyclic if each of its connected components can be decomposed into a "join tree" (classical notion); we slightly generalize this notion by ignoring constants in hyperedge intersections.

[6] Due to space restriction, this translation cannot be detailed. It relies on the join tree decomposition of each rule body.