Extending Decidable Existential Rules by Joining Acyclicity and Guardedness

Markus Krötzsch
University of Oxford, UK
markus.kroetzsch@comlab.ox.ac.uk

Sebastian Rudolph
Karlsruhe Institute of Technology, Germany
sebastian.rudolph@kit.edu

Abstract

Existential rules, i.e. Datalog extended with existential quantifiers in rule heads, are currently studied under a variety of names such as *Datalog+/–*, *∀∃-rules*, and *tuple-generating dependencies*. The renewed interest in this formalism is fuelled by a wealth of recently discovered language fragments for which query answering is decidable. This paper extends and consolidates two of the main approaches in this field – acyclicity and guardedness – by providing (1) complexity-preserving generalisations of weakly acyclic and weakly (frontier-)guarded rules, and (2) a novel formalism of *glut-(frontier-)guarded rules* that subsumes both. This builds on an insight that acyclicity can be used to extend any existential rule language while retaining decidability. Besides decidability, combined query complexities are established in all cases.

1 Introduction

Rule-based knowledge representation has a long-standing history in AI and related areas such as databases and information systems. Function-free first-order Horn logic (also referred to as Datalog) as one of the central paradigms, however, has been criticised for its inability of stating or inferring the existence of domain entities not previously introduced as constants [Patel-Schneider and Horrocks, 2007]. Existential rules, i.e. Datalog extended by *value invention* capabilities realised by existential quantifiers in rule heads, overcome this restriction and are currently studied under a variety of names such as *Datalog+/–*, *∀∃-rules*, and – primarily in the database community – *tuple-generating dependencies* (TGDs) [Baget *et al.*, 2010; 2009; Calì *et al.*, 2010a; 2010b; 2009; 2008; Fagin *et al.*, 2005; Deutsch and Tannen, 2003]. The recent interest in this formalism marks the convergence of two paradigms of knowledge representation research that used to be rather separated: rule-based approaches and ontology languages.

This new ground was found to be very fertile, as witnessed by the above works' discoveries of many new rule languages for which query answering is decidable. Widely varying data and combined complexities underline the richness of the field. Examples of application areas for this new family of knowledge representation languages range from data exchange and data integration [Fagin *et al.*, 2005] to ontological data access in the spirit of the ontology languages of the DL-Lite family [Calì *et al.*, 2009; Calvanese *et al.*, 2007]. The wealth of recent contributions supports the development of such applications, but also calls for a more unified view on the existing proposals, their exact relationships, and formal properties. This is the general incentive for this work.

Concretely, we extend and consolidate two of the main notions commonly employed to ensure decidability: acyclicity and guardedness. The main contributions are as follows.

1. We extend weak acyclicity and weak (frontier-) guardedness to obtain *joint acyclicity* and *joint (frontier-) guardedness*. Both extensions use the observation that the existing notions over-estimate how far values can be passed on within a rule set, and that there is a refined criterion that still can be checked in polynomial time.
2. We present a new method of eliminating existential quantifiers from jointly acyclic rule sets. The approach incurs an exponential blow-up but is still worst-case optimal. The relevance of the method stems from the insight that a partial application of the procedure can also simplify rule sets that are not jointly acyclic.
3. We apply this observation to combine guardedness and acyclicity in the language of *glut-(frontier-)guarded rules*, based on identifying *glut variables* that may represent an overabundance of "existentially invented" values. Only glut variables remain affected by existential quantifiers after applying the elimination method introduced for jointly acyclic rules.

An important insight of this work therefore is that a very general notion of acyclicity can be combined "modularly" with existing rule languages without losing decidability. Jointly frontier-guarded rules serve us as an example for this construction, and illustrate that further studies are needed to determine the exact complexity of reasoning in each case. We determine exact combined worst-case complexities for all rule languages introduced herein.

Section 2 provides the preliminaries and reviews the existing results in the field. We then motivate and introduce the notion of joint acyclicity in Section 3, and present a generic way of eliminating jointly acyclic variables in Section 4. Section 5 introduces jointly frontier-guarded rules, and Section 6 combines all previous ideas to obtain *glut-(frontier-)guarded rules* for which the combined complexity of query answering is shown to be 3ExpTime-complete. Section 7 concludes. An extended version of this paper that contains detailed proofs is available as technical report [Krötzsch and Rudolph, 2011].

2 Existential Rules

We now provide the basic notions of the logical framework we consider, followed by an overview of a number of important approaches in this area.

Definition 1 *Consider a* signature $\langle \mathbf{C}, \mathbf{P}, \mathbf{V} \rangle$ *consisting of a finite set of* constant symbols $\mathbf{C}$*, a finite set of* predicates $\mathbf{P}$*, and an infinite set of* variables $\mathbf{V}$*, all of which are mutually disjoint. A function* $\mathsf{ar} : \mathbf{P} \to \mathbb{N}$ *associates a natural number* $\mathsf{ar}(r)$ *with each predicate* $r \in \mathbf{P}$ *that defines the* arity *of* r*. The set of* positions *of a predicate* r *is the set* $\Pi_r = \{\langle r, 1\rangle, \ldots, \langle r, \mathsf{ar}(r)\rangle\}$.

- *A* term *is a variable* $x \in \mathbf{V}$ *or a constant* $c \in \mathbf{C}$.
- *An* atom *is a formula of the form* $r(t_1, \ldots, t_n)$ *if* $t_1, \ldots, t_n$ *are terms, and* $r \in \mathbf{P}$ *is a predicate with* $\mathsf{ar}(r) = n$.
- *An* existential rule *(or simply* rule *in the context of this paper) is a formula of the form*

$$\forall \mathbf{x}.(B_1 \wedge \ldots \wedge B_k \to \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_l),$$

  where $B_1, \ldots, B_k, H_1, \ldots, H_l$ *are atoms all of whose variables are in the scope of some quantifier, and where no variable occurs more than once in* $\mathbf{x}, \mathbf{y}$.[1] *We use sets of atoms as a convenient notation for conjunctions of atoms. A* Datalog rule *is a rule with no existential quantifiers. A rule with* $k = 0$ *is called a* fact *(a conclusion that is unconditionally true), and a rule with* $l = 0$ *is called a* constraint *(a premise that must never be true).*

The premise of a rule is called the body *while the conclusion is called the* head*. Since all variables in rules are quantified, we will often omit the explicit preceding universal quantifier.*

A rule set Σ *is* renamed apart *if each variable name is bound in at most one quantifier in* Σ.

The rule language hereby introduced is a syntactic fragment of first-order predicate logic, and we consider it under the according semantics. This also means that every rule set is semantically equivalent to one that is renamed apart. Moreover, note that we do not exclude *non-safe rules*, i.e. rules with universally quantified variables that occur in the head but not in the body; all of our results apply in any case.

Definition 2 *Let* Σ *be a set of rules. We call* Σ satisfiable *if it has a model according to the standard semantics of first-order logic. Two rule sets* Σ *and* Σ' *are* equisatisfiable *if either both or none of them is satisfiable. A* boolean conjunctive query *(BCQ) is a formula* $\exists \mathbf{v}.Q$ *where* Q *is a conjunction of atoms and* $\mathbf{v}$ *contains all variables in* Q*. A BCQ* $\exists \mathbf{v}.Q$ *is entailed by* Σ *if it is entailed under standard first-order logic semantics.*

[1] We freely use $\mathbf{x}$, $\mathbf{t}$, etc. to denote vectors of the form $\langle x_1, \ldots, x_n\rangle$, $\langle t_1, \ldots, t_n\rangle$, etc. throughout this paper.

Checking satisfiability and BCQ entailment for unrestricted existential rules is undecidable [Chandra *et al.*, 1981b; Beeri and Vardi, 1981] even with very strong restrictions on the vocabulary or the number of rules [Baget *et al.*, 2010]. Therefore, a large body of work has been devoted to the identification of restricted rule languages which retain decidability and still allow for sufficient expressiveness. A generic tool for establishing decidability results is the *chase* introduced by Maier *et al.* [1979] and extended to query containment by Johnson and Klug [1982]. Intuitively the chase procedure starts with a given set of factual data (ground facts) and "applies" rules in a production rule style by introducing new domain elements whenever required by an existentially quantified variable in a rule head. In general, termination of this procedure cannot be guaranteed, and an infinite set of new domain elements and facts may be created.

Many of the decidable rule classes come about by establishing properties about the chase they create. Finiteness of the chase is a straightforward criterion for ensuring decidability, and rule sets with this property are called *finite extension sets* [Baget *et al.*, 2010]. This criterion is undecidable in general, but several sufficient conditions on rule sets for chase-finiteness have been identified. Pure Datalog (also known as *full implicational dependencies* [Chandra *et al.*, 1981b] or *total TGDs* [Beeri and Vardi, 1981]) is an immediate case, as no new domain elements are created at all. A more elaborate concept is (weak) acyclicity [Deutsch and Tannen, 2003; Fagin *et al.*, 2005] which we review and extend in Section 3. Another approach that pursues a similar goal by different means is to require acyclicity of the *graph of rule dependencies* introduced by Baget *et al.* [2009].

An even more relaxed condition than finiteness of the chase is that the (possibly infinite) chase enjoys a variant of the bounded treewidth property, leading to *bounded treewidth sets* [Baget *et al.*, 2010]. Decidability of BCQ entailment follows from known decidability results for first-order logic theories with the bounded treewidth model property [Courcelle, 1990]. Again rules with this property are not recognisable in general, but a variety of sufficient conditions has been established. The most prominent examples are a number of *guardedness* conditions that we review and extend in Section 5.

Independently of the chase, other decidability criteria can be established by considering rewritings of the query in a backward-chaining manner. In analogy to the finite chase condition, one can define *finite unification sets* where this rewriting procedure terminates and yields a finite set of rewritten queries [Baget *et al.*, 2010]. First-order rewritability also implies a sub-polynomial AC_0 data complexity for BCQ entailment checking. Again, recognising finite unification sets is undecidable, and various decidable sublanguages are known. Examples include *atomic-hypothesis rules* and *domain restricted rules* [Baget *et al.*, 2010], *linear Datalog+/–* [Calì *et al.*, 2009], *sticky sets of TGDs*, and *sticky-join sets of TGDs* [Calì *et al.*, 2010a; 2010b].

3 Joint Acyclicity

This section introduces *joint acyclicity*, which is a proper generalisation of the following notion of *weak acyclicity*

[Deutsch and Tannen, 2003; Fagin *et al.*, 2005].

Definition 3 *For a set of rules Σ, the* dependency graph *is a directed graph that has the positions of predicates in Σ as its nodes. For every rule $\rho \in \Sigma$, and every variable x at position $\langle r, p\rangle$ in the head of ρ, the graph contains edges as follows:*

- *If x is universally quantified, and x occurs in a body atom at position $\langle s, q\rangle$, there is an edge from $\langle s, q\rangle$ to $\langle r, p\rangle$.*
- *If x is existentially quantified, and the body of ρ contains a (necessarily universally quantified) variable y at $\langle s, q\rangle$, then there is a* special edge *from $\langle s, q\rangle$ to $\langle r, p\rangle$.*[2]

Σ *is* weakly acyclic *if its dependency graph has no cycle going through a special edge.*

Intuitively, non-special edges encode the possible passing of values in bottom-up reasoning, whereas special edges encode the dependency between the premise that a rule was applied to and the new individuals that the application of this rule entails. A cycle over special edges may indicate that newly invented values can recursively be used in premises which require the invention of further values *ad infinitum*. For instance, the rule

$$r(x, y) \to \exists z.r(y, z) \tag{1}$$

may lead to the construction of an infinite r-chain of new elements, and indeed the dependency graph has a special edge from $\langle r, 2\rangle$ to itself. But weak acyclicity also excludes cases where no infinite recursion would occur:

$$r(x, y) \land c(y) \to \exists z.r(y, z) \tag{2}$$

The dependency graph contains the same cycle as before, yet the rule cannot be applied recursively since invented values are not required to belong to c. Note that this remains true even if there are other rules with existentially quantified variables at $\langle c, 1\rangle$. We capture this by shifting our focus from positions to variables (which can occur in multiple positions):

Definition 4 *Consider a renamed apart set of rules Σ. For a variable x, let Π^B_x (Π^H_x) be the set of all positions where x occurs in the body (head) of a – necessarily unique – rule. Now for any existentially quantified variable, let Ω_x be the smallest set of positions such that (1) $\Pi^H_x \subseteq \Omega_x$, and (2) $\Pi^H_y \subseteq \Omega_x$ for every universally quantified variable y with $\Pi^B_y \subseteq \Omega_x$.*

The existential dependency graph *of Σ has the existentially quantified variables of Σ as its nodes. There is an edge from x to y if the rule where y occurs contains a universally quantified (body) variable z with $\Pi^B_z \subseteq \Omega_x$. Σ is* jointly acyclic *if its existential dependency graph is acyclic.*

Thus Ω_x contains the positions in which values invented for x may appear. This captures the effect of non-special edges in Definition 3, whereas special edges correspond to edges in the existential dependency graph. Definition 3 is obtained by modifying condition (2) in Definition 4 to require $\Pi^B_y \cap \Omega_x \neq \emptyset$ instead of $\Pi^B_y \subseteq \Omega_x$. This states that a value is propagated by a rule if it satisfies *some* – instead of *all* – of the rule's premises. Joint acyclicity therefore appears to be more natural.

The following rule is jointly acyclic (as a singleton set) but not weakly acyclic: its existential dependency graph has no edges while its dependency graph is a clique of special edges.

$$r(x, y) \land s(x, y) \to \exists v,w.r(x,v) \land r(w, y) \land s(x,w) \land s(v, y) \tag{3}$$

In spite of this generalisation, joint acyclicity is easy to recognise. Detecting cycles in a directed graph and checking inclusion of a position in Ω_x is possible in polynomial time. The latter problem is also hard for P since propositional Horn logic entailment can be expressed using unary predicates with a single variable to encode propositions.

Another generalisation of weak acyclicity, called *Super-weak acyclicity* (SwA), has been proposed in [Marnette, 2009]. SwA is more general than joint acyclicity as it uses function symbols and unification to exclude some additional cases of value propagation. It remains open how our results can be extended to SwA.

4 Reducing Jointly Acyclic Variables

We now present a method for eliminating existential quantifiers from rule sets. Applied iteratively to jointly acyclic rules, this procedure yields a Datalog program that faithfully represents all consequences of the original rule set. This establishes decidability and optimal complexity bounds for jointly acyclic rules. For the general case, the procedure still allows semantically faithful simplifications of rules that can be used to extend other decidable rule languages as in Section 6.

Our transformation simulates Skolemisation, the replacement of existentially quantified variables with Skolem terms, where we "flatten" function terms to represent them in Datalog. For example, Skolemising the rule $r(x, y) \to \exists v.s(x, v)$ yields $r(x, y) \to s(x, f(x, y))$ where f is a fresh function symbol. We express this without functions by considering f as a constant and replacing s by a predicate s' of higher arity: $r(x, y) \to s'(x, f, x, y)$. Other predicates may need to be extended analogously in positions where the Skolem term might be relevant; those are exactly the positions in Ω_v. Conversely, some uses of s may not require all the new positions, and we use a special symbol $\Box$ as a filler. For example, a fact $s(a, b)$ is represented as $s'(a, b, \Box, \Box)$.

Definition 5 *Consider a renamed apart rule set Σ, such that there is an existentially quantified variable x that does not have incoming edges in the existential dependency graph.*

Let k be the number of universally quantified variables in the rule containing x. For a predicate r define n_r to be the cardinality of the set $\{\langle r, p\rangle \in \Omega_x \mid 1 \leq p \leq \mathsf{ar}(r)\}$. If $n_r > 0$ let $\hat{r}$ denote a fresh predicate of arity $\mathsf{ar}(\hat{r}) = \mathsf{ar}(r) + n_r k$; if $n_r = 0$ let $\hat{r}$ denote r. Let f and $\Box$ be fresh constant symbols.

Σ_x is the set of rules that contains, for each rule $\rho \in \Sigma$, the rule ρ_x that is obtained by replacing each atom $r(t_1, \ldots, t_{\mathsf{ar}(r)})$ in ρ by the atom $\hat{r}(\mathbf{s_1}, \ldots, \mathbf{s}_{\mathsf{ar}(r)})$ where the term vectors $\mathbf{s_i}$ are defined as follows:

- *If $\langle r, i\rangle \notin \Omega_x$ then $\mathbf{s_i} := t_i$.*

  For the remaining cases, assume that $\langle r, i\rangle \in \Omega_x$.
- *If $t_i = x$ then $\mathbf{s_i} := \langle f, y_1, \ldots, y_k\rangle$ where $y_1, \ldots, y_k$ are all universally quantified variables in the rule.*

[2] The definition of Fagin *et al.* [2005] is slightly more general by requiring y to also occur in the head. Here we want to focus on the main idea of acyclicity. Our glut-frontier-guarded rules in Section 6 then properly generalise the original form of weak acyclicity.

- *If $t_i = y$ is universally quantified and occurs only in positions in Ω_x, then $\mathbf{s_i} \coloneqq \langle y_0, y_1, \ldots, y_k\rangle$ where the same fresh universally quantified variable names y_j are used in all replacements of y but nowhere else.*
- *In all other cases, $\mathbf{s_i} \coloneqq \langle t_i, \square, \ldots, \square\rangle$ where this is a vector of length $k+1$.*

Quantifiers for ρ are updated accordingly: new universal quantifiers are introduced for all variables of the form y_j, and the existential quantifier for x is deleted.

For a boolean conjunctive query $\exists\mathbf{v}.Q$ over the signature of Σ, the BCQ $\exists\mathbf{v}.Q_x$ is defined as the body of the rule $Q_x \rightarrow$ obtained by applying the above transformation to the rule $Q \rightarrow$.

Note that this definition is well. In particular, for each r we find that n_r of the vectors $\mathbf{s_i}$ are of length $k+1$, and all others are of length 1, yielding the required $\mathsf{ar}(r) + n_r k$ arguments of $\hat{r}$. Applying this transformation to v in rule (3), we have $k = 2$ and $\Omega_v = \{\langle r, 2\rangle, \langle s, 1\rangle\}$, and so obtain:

$$\hat{r}(x, y, \square, \square) \wedge \hat{s}(x, \square, \square, y) \rightarrow \exists w.\hat{r}(x, f, x, y) \wedge \hat{r}(w, y, \square, \square) \wedge \hat{s}(x, \square, \square, w) \wedge \hat{s}(f, x, y, y) \quad (4)$$

Next, we state the main correctness result for this transformation. The respective proof in [Krötzsch and Rudolph, 2011] directly shows equisatisfiability using suitable model transformations. This is not hard to formalise after observing the correspondence of domain elements in models of Σ on the one hand, and vectors of such elements – corresponding to term vectors $\mathbf{s_i}$ in Definition 5 – in models of Σ_x on the other.

Theorem 1 *Given a set of rules Σ and a variable x as in Definition 5, Σ is satisfiable if and only if Σ_x is satisfiable. Moreover, a BCQ $\exists\mathbf{v}.Q$ over the signature of Σ is entailed by Σ if and only if $\exists\mathbf{v}.Q_x$ is entailed by Σ_x.*

We can thus apply Definition 5 iteratively, where Theorem 1 ensures that correctness is preserved. It is important that the iterative reduction also preserves joint acyclicity:

Theorem 2 *Consider a rule set Σ, and a variable x as in Definition 5. The variables $y \neq x$ without incoming edges in the existential dependency graph of Σ do not have incoming edges in the existential dependency graph of Σ_x either. Moreover, Σ is jointly acyclic if and only if Σ_x is jointly acyclic.*

The previous theorem ensures that the set of variables that can be eliminated by applying Definition 5 iteratively is not affected by the order in which variables are reduced in case there is more than one variable without incoming edges. Yet, iterative reductions may yield syntactically different results depending on the order of application. This non-determinism is inessential for our considerations, so we use $\mathsf{ja}(\Sigma)$ to denote an arbitrary but fixed rule set obtained by iteratively applying Definition 5 until it is no longer applicable.

Theorem 3 *If Σ is a jointly acyclic, renamed apart set of rules Σ then $\mathsf{ja}(\Sigma)$ is a Datalog program.*

Before stating the main complexity result of this section, we provide a more precise estimate of the increase in size that is caused by the transformation. Importantly, the exponential blow-up is caused by chains of dependencies in the existential dependency graph, not by the size of the rule set in general.

Theorem 4 *Given a renamed apart rule set Σ, the set $\mathsf{ja}(\Sigma)$ contains the same number of rules as Σ, and the same number of head and body atoms in each rule. The number of variables per rule in $\mathsf{ja}(\Sigma)$ is bounded by a function that is exponential in the maximum directed path length in the existential dependency graph of Σ, and polynomial in the size of Σ.*

Theorem 5 *Deciding whether a BCQ is entailed by a jointly acyclic set of rules is* 2ExpTime*-complete for combined complexity,* ExpTime*-complete if the maximal length of a path in the existential dependency graph is bounded, and* P*-complete in data complexity.*

5 Jointly Frontier-Guarded Rules

A large class of existential rules for which query answering is decidable are based on the idea of *guardedness* [Andréka *et al.*, 1998], the requirement that all or some of the universally quantified variables of a rule appear together in a single "guard" atom. Requiring guards only for variables that also appear in the head (the "frontier") yields *frontier-guarded rules* [Baget *et al.*, 2010]. Both notions can be generalised by not requiring guards for variables that cannot possibly represent existentially introduced elements. This idea has been used to arrive at *weakly guarded rules* [Calì *et al.*, 2008] and *weakly frontier-guarded rules* [Baget *et al.*, 2010]. In this section, we generalise the latter to fit more naturally to our definitions in Section 3, and we establish basic complexity results.

Definition 6 *Consider a set of rules Σ. A position $\langle r, i\rangle$ is* affected *if (1) Σ contains an existentially quantified variable on position $\langle r, i\rangle$, or (2) Σ contains a universally quantified variable x on position $\langle r, i\rangle$ in the head of a rule where x occurs on an affected position in its body. A position $\langle r, i\rangle$ is* jointly affected *if $\langle r, i\rangle \in \Omega_x$ for a variable x in Σ (see Definition 4).*

A variable x in a rule $\rho = \forall\mathbf{x}.\varphi \rightarrow \exists\mathbf{y}.\psi \in \Sigma$ is universal *if it occurs in $\mathbf{x}$,* affected *if it occurs on some affected position in φ,* jointly affected *if it occurs only on jointly affected positions in φ,* frontier *if it occurs φ and in ψ. The sets of all such variables are denoted X^u_ρ, X^a_ρ, X^{ja}_ρ, X^f_ρ.*

The rule ρ is X-guarded for a set X of variables, if all $x \in X$ occur together in one atom in φ. Relevant notions are: guarded *($X = X^u_\rho$),* frontier-guarded *($X = X^f_\rho$),* weakly guarded *($X = X^a_\rho$),* weakly frontier-guarded *($X = X^a_\rho \cap X^f_\rho$),* jointly guarded *($X = X^{ja}_\rho$),* jointly frontier-guarded *($X = X^{ja}_\rho \cap X^f_\rho$). The set Σ is X-guarded if all rules $\rho \in \Sigma$ are.*

The relation of these notions follows from the observation that $X^u_\rho \supseteq X^f_\rho$ and $X^u_\rho \supseteq X^a_\rho \supseteq X^{ja}_\rho$, e.g. every weakly guarded rule is also jointly frontier-guarded. The combined complexity of BCQ answering for guarded and weakly guarded rules is known to be 2ExpTime-complete [Calì *et al.*, 2008]. Hardness carries over to the frontier-guarded cases, but upper complexity bounds for these languages have been open until very recently. We cite the following result from Baget *et al.* [2011].

Proposition 1 *Deciding whether a BCQ is entailed by a frontier-guarded set of rules is* 2ExpTime*-complete for combined complexity.*

Baget *et al.* [2011] further show that BCQ answering for weakly frontier-guarded rules is in 2ExpTime. Here, we extend this result to our new notion of jointly guarded and jointly frontier-guarded rules. We observe that variables that are not jointly affected may never represent elements that are introduced existentially. Hence, their assignments correspond to constant symbols that could be substituted instead. A naive use of this idea yields exponentially many *partially grounded* rules with constants used in all possible combinations.

A polynomial reduction is possible by extending the arguments of all predicates to contain parameters for all variables that are not jointly affected. These parameters then guard all such variables in rules. Bindings for the added parameters can only be inferred by auxiliary rules that allow arbitrary constants to be substituted for variables. These ideas are combined to the following definition.

Definition 7 *For a renamed apart rule set Σ, let $\mathbf{z} = \langle z_1, \ldots, z_n \rangle$ be a list of all variables in Σ that are not jointly affected, and let $\tilde{r}$ be a fresh predicate of arity $\mathsf{ar}(r) + n$ for each predicate r of Σ. The rule set $\mathsf{guard}(\Sigma)$ consists of:*

(1) for each rule $\rho \in \Sigma$ with non-empty body, a rule $\rho' \in \mathsf{guard}(\Sigma)$ obtained by replacing each atom $r(t_1, \ldots, t_{\mathsf{ar}(r)})$ (with terms t_i) by $\tilde{r}(t_1, \ldots, t_{\mathsf{ar}(r)}, z_1, \ldots, z_n)$, where all variables z_i are universally quantified,

(2) for each rule $\rho \in \Sigma$ with empty body (i.e. generalised fact), a rule $\rho' \in \mathsf{guard}(\Sigma)$ obtained by replacing each atom $r(t_1, \ldots, t_{\mathsf{ar}(r)})$ (with terms t_i) by $\tilde{r}(t_1, \ldots, t_{\mathsf{ar}(r)}, c, \ldots, c)$ where c is an arbitrary constant,

(3) for each predicate r of Σ, each $i \in \{1, \ldots, n\}$, and each constant symbol c, a rule

$$\tilde{r}(x_1, \ldots, x_{\mathsf{ar}(r)}, z_1, \ldots, z_i, \ldots, z_n) \\ \to \tilde{r}(x_1, \ldots, x_{\mathsf{ar}(r)}, z_1, \ldots, c, \ldots, z_n),$$

(4) for each predicate r of Σ, a rule

$$\tilde{r}(x_1, \ldots, x_{\mathsf{ar}(r)}, z_1, \ldots, z_n) \to r(x_1, \ldots, x_{\mathsf{ar}(r)}),$$

where all variable names x_i are fresh.

The next theorem shows the correctness of this transformation. The proof in [Krötzsch and Rudolph, 2011] directly transforms models of Σ into models of $\mathsf{guard}(\Sigma)$, and vice versa, restricting to minimal models in the latter case.

Theorem 6 *A BCQ $\exists \mathbf{v}.Q$ is entailed by a renamed apart rule set Σ iff $\exists \mathbf{v}.Q$ is entailed by $\mathsf{guard}(\Sigma)$.*

The following theorem is easily obtained by summing up the above results.

Theorem 7 *Deciding whether a BCQ is entailed by a jointly guarded or jointly frontier-guarded set of rules is* 2ExpTime*-complete for combined complexity.*

6 Joining Acyclicity and Guardedness

The iterative reduction in Section 4 hints at a much wider applicability of the idea of joint acyclicity, since it allows for the elimination of some existential quantifiers even in rule sets that are not jointly acyclic. This is useful if the reduced rule set belongs to a rule language for which decidability of reasoning has been established on other grounds. In this section, we illustrate this idea by combining acyclicity with joint (frontier-)guardedness, and establish tight complexity bounds for related reasoning tasks.

Using the terminology of Section 5, we can say that Definition 5 eliminates jointly affected variables. To be more precise, we say that a variable in a renamed apart rule set Σ is a *glut variable* if it occurs in a set Ω_x as in Definition 5 for a variable x that is part of a cycle in the existential dependency graph. Intuitively, glut variables may represent an overabundance of values, as opposed to the remaining, *non-glut variables* that can only represent finitely many values. Clearly, the iterative application of Definition 5 then turns non-glut variables into variables that are not jointly affected. This leads to a further generalisation of guardedness:

Definition 8 *A renamed apart rule set Σ is* glut-guarded *(*glut-frontier-guarded*) if each rule of Σ has a body atom that contains all glut variables (that also occur in the head).*

This definition is illustrated in the following example of a glut-frontier-guarded rule set, where c, intuitively speaking, marks persons that are "specifically important" for us:

$$c(x) \wedge ancestor(x, \dot{y}) \wedge ancestor(\dot{y}, \dot{z}) \to ancestor(x, \dot{z}) \quad (5)$$
$$parent(\dot{x}, \dot{y}) \to ancestor(\dot{x}, \dot{y}) \quad (6)$$
$$c(x) \to person(x) \quad (7)$$
$$person(\dot{x}) \to \exists \dot{w}.parent(\dot{x}, \dot{w}) \wedge person(\dot{w}) \quad (8)$$
$$sibling(x, y) \to \exists v.parent(x,v) \wedge parent(y,v) \wedge c(v) \quad (9)$$
$$parent(\dot{x}, y) \wedge sibling(y, z) \to uncle(\dot{x}, z) \quad (10)$$

Information about c, *parent*, and *sibling* would be given in facts, while the remaining predicates are derived only. The existential dependency graph has two edges $v \to w$ and $w \to w$, where the latter cycle follows from (8). Glut variables thus are those occurring only on positions of Ω_w; they are marked by a dot in the example. It is easy to verify that the example is glut-frontier-guarded. Note how c is used to make x in rule (5) non-glut, thus allowing a form of transitivity – a typical counter-example for all common types of guardedness. Furthermore, transitivity is not first-order rewritable, thus excluding the example from all types of finite unification sets reviewed in Section 2. Rule (10) is another illustration of the increased expressive power, since it is neither jointly frontier-guarded nor glut-guarded. Indeed, since all positions other than those of *sibling* are in Ω_v, almost all variables in the example are jointly affected.

Theorem 8 *Deciding whether a BCQ is entailed by a glut-guarded or glut-frontier-guarded set of rules Σ is* 3ExpTime*-complete for combined complexity.*

Inclusion is shown by applying Theorems 1 and 4 to obtain that $\mathsf{ja}(\Sigma)$ is an exponentially large rule set that can be used for BCQ entailment checking. Clearly, $\mathsf{ja}(\Sigma)$ is jointly frontier-guarded, so the result follows from Theorem 7.

For hardness, one simulates an Alternating Turing Machine (ATM) with doubly exponential space. Such ATMs can accept all languages that a Turing Machine can accept given triply exponential time [Chandra *et al.*, 1981a]. The ATM acceptance conditions as such can be formulated using frontier-guarded rules, but the efficient encoding of a doubly exponential storage tape requires additional existential quantifiers. This leads to further variables being jointly affected, but not glut. The tape construction adapts a method for constructing doubly exponential chains proposed by Calì *et al.* [2010b]. Details are given in [Krötzsch and Rudolph, 2011].

7 Conclusion

We have extended the notions of weak acyclicity and weak (frontier-)guardedness, introduced a versatile new method for eliminating existential quantifiers, and applied these insights to define glut-frontier-guarded rules as one of the most expressive known existential rule languages for which query answering is decidable. Yet, a wide range of open issues still needs to be tackled for developing both the foundations of the field and applications to use these novel approaches.

Some immediate questions raised by this work concern the query complexity for fixed non-ground rules (data complexity) or for fixed signatures (bounded arity). A concurrent anonymous submission to this conference addresses these issues for previously defined rule languages, and it will be interesting to lift the respective methods to our cases.

More generally, further efforts are needed to continue the consolidation of rule languages that was started herein. To this end, modular reduction techniques for simplifying rule sets can be of great utility for advancing towards a unified theory of decidable existential rules.

Acknowledgements We thank Despoina Magka for helpful comments. Markus Krötzsch was supported by the EPSRC project ExODA (EP/H051511/1). Sebastian Rudolph was supported by the project ExpresST funded by the German Research Foundation (DFG).

References

[Andréka *et al.*, 1998] Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. of Philosophical Logic*, 27(3):217–274, 1998.

[Baget *et al.*, 2009] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In Craig Boutilier, editor, *Proc. 21st Int. Conf. on Artificial Intelligence (IJCAI'09)*, pages 677–682. IJCAI, 2009.

[Baget *et al.*, 2010] Jean-François Baget, Michel Leclère, and Marie-Laure Mugnier. Walking the decidability line for rules with existential variables. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proc. 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 466–476. AAAI Press, 2010.

[Baget *et al.*, 2011] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In Toby Walsh, editor, *Proc. 22nd Int. Conf. on Artificial Intelligence (IJCAI'11)*. IJCAI, 2011.

[Beeri and Vardi, 1981] Catriel Beeri and Moshe Y. Vardi. The implication problem for data dependencies. In Shimon Even and Oded Kariv, editors, *Proc. 8th Colloquium on Automata, Languages and Programming (ICALP'81)*, volume 115 of *LNCS*, pages 73–85. Springer, 1981.

[Calì *et al.*, 2008] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In Gerhard Brewka and Jérôme Lang, editors, *Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 70–80. AAAI Press, 2008.

[Calì *et al.*, 2009] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In Jan Paredaens and Jianwen Su, editors, *Proc. 28th Symposium on Principles of Database Systems (PODS'09)*, pages 77–86. ACM, 2009.

[Calì *et al.*, 2010a] Andrea Calì, Georg Gottlob, and Andreas Pieris. Advanced processing for ontological queries. *Proceedings of VLDB 2010*, 3(1):554–565, 2010.

[Calì *et al.*, 2010b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Query answering under non-guarded rules in Datalog+/-. In Pascal Hitzler and Thomas Lukasiewicz, editors, *Proc. 4th Int. Conf. on Web Reasoning and Rule Systems (RR 2010)*, volume 6333 of *LNCS*, pages 1–17. Springer, 2010.

[Calvanese *et al.*, 2007] Diego Calvanese, Guiseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[Chandra *et al.*, 1981a] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. of the ACM*, 28(1):114–133, 1981.

[Chandra *et al.*, 1981b] Ashok K. Chandra, Harry R. Lewis, and Johann A. Makowsky. Embedded implicational dependencies and their inference problem. In *Proc. 13th Annual ACM Symposium on Theory of Computation (STOC'81)*, pages 342–354. ACM, 1981.

[Courcelle, 1990] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[Deutsch and Tannen, 2003] Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In Diego Calvanese, Maurizio Lenzerini, and Rajeev Motwani, editors, *Proc. 9th Int. Conf. on Database Theory (ICDT'03)*, volume 2572 of *LNCS*, pages 225–241. Springer, 2003.

[Fagin *et al.*, 2005] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

[Johnson and Klug, 1982] David S. Johnson and Anthony Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. In *Proc. 1st Symposium on Principles of Database Systems (PODS'82)*, pages 164–169. ACM, 1982.

[Krötzsch and Rudolph, 2011] Markus Krötzsch and Sebastian Rudolph. Revisiting acyclicity and guardedness criteria for decidability of existential rules. Technical Report 3011, Institute AIFB, Karlsruhe Institute of Technology, 2011.

[Maier *et al.*, 1979] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems*, 4:455–469, 1979.

[Marnette, 2009] Bruno Marnette. Generalized schema-mappings: from termination to tractability. In Jan Paredaens and Jianwen Su, editors, *Proc. 28th Symposium on Principles of Database Systems (PODS'09)*, pages 13–22. ACM, 2009.

[Patel-Schneider and Horrocks, 2007] Peter F. Patel-Schneider and Ian Horrocks. A comparison of two modelling paradigms in the Semantic Web. *J. of Web Semantics*, 5:240–250, 2007.