Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Combining Existential Rules and Description Logics

Antoine Amarilli
Télécom ParisTech; Institut Mines–Télécom; CNRS LTCI
antoine.amarilli@telecom-paristech.fr

Michael Benedikt
University of Oxford
michael.benedikt@cs.ox.ac.uk

Abstract

Query answering under *existential rules* — implications with existential quantifiers in the head — is known to be decidable when imposing restrictions on the rule bodies such as frontier-guardedness [Baget *et al.*, 2010; 2011a]. Query answering is also decidable for *description logics* [Baader, 2003], which further allow disjunction and *functionality constraints* (assert that certain relations are functions); however, they are focused on ER-type schemas, where relations have arity two.

This work investigates how to get the best of both worlds: having decidable existential rules on arbitrary arity relations, while allowing rich description logics, including functionality constraints, on arity-two relations. We first show negative results on combining such decidable languages. Second, we introduce an expressive set of existential rules (frontier-one rules with a certain restriction) which can be combined with powerful constraints on arity-two relations (e.g. $\mathsf{GC}^2, \mathcal{ALCQIb}$) while retaining decidable query answering. Further, we provide conditions to add functionality constraints on the higher-arity relations.

1 Introduction

Recent years have seen an explosion of techniques for solving the *query answering problem*: given a query q, a conjunction F of atoms, and a set of logical constraints Σ, determine whether q follows from F and Σ. In databases this is called *querying under constraints* or the *certain answer problem*, seeing F as an incomplete database, and Σ as restrictions on the possible completions. For researchers working on description logics, F is referred to as the *A-box* and Σ the *T-box*. In both communities q is usually a *conjunctive query*, an existential quantification of conjunctions of atoms, equivalent to a basic SQL SELECT. We will make this assumption throughout this work, referring for simplicity to the problem as just "query answering" (QA).

QA is undecidable when Σ ranges over arbitrary first-order logic constraints. This motivates the search for restricted constraint languages with decidable QA. Within the description logic community, powerful such languages were developed to express constraints on vocabularies of arity two. The unary relations are referred to as *concepts* while the binary ones are the *roles*. The languages can build new concepts and roles from basic ones via Boolean operations and (limited) quantification, and many of them, such as DL-Lite [Calvanese *et al.*, 2005] or $\mathcal{ALCQIb}$ [Tobies, 2001], may restrict the input roles $R(x,y)$ to be *functional* – for all x there is at most one y such that $R(x,y)$. Functionality constraints are crucial to faithfully model many real-world relationships: the relationship of a person to their birthdate, the relationship of an event to its starting time, etc. Hence, description logics are very powerful languages for *arity-two vocabularies*.

In parallel, the AI and database communities have developed rich constraint languages on arbitrary arity via *existential rules* or *tuple-generating dependencies* (TGDs). Existential rules are constraints of the form $\forall \mathbf{x}\ (\phi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}', \mathbf{y}))$ where $\mathbf{x}' \subseteq \mathbf{x}$ and ϕ and ψ are conjunctions of atoms. They generalize the well-known *inclusion dependencies* or *referential constraints* in databases [Abiteboul *et al.*, 1995], and can also express mapping relationships used in data exchange [Fagin *et al.*, 2005] and data integration [Lenzerini, 2002]. Although QA over general rules is undecidable, important subclasses are decidable. First, decidability holds whenever the *chase* procedure [Abiteboul *et al.*, 1995] is guaranteed to terminate, which is ensured by a number of conditions on the rules, e.g., weak acyclicity [Fagin *et al.*, 2005], joint acyclicity [Krötzsch and Rudolph, 2011], or the very restricted class of source-to-target TGDs. See [Grau *et al.*, 2013] for a survey and [Baget *et al.*, 2014] for a recent study. A second class of tame constraints are those that admit bounded-treewidth models. There are several such classes, such as *guarded TGDs* [Calì *et al.*, 2012a], *frontier-guarded TGDs* [Baget *et al.*, 2010], or the more general *greedy bounded-treewidth sets* [Baget *et al.*, 2011b]. However, many features of description logics, such as disjunction or functionality restrictions, cannot be expressed by existential rules.

Could we then enjoy the best of both worlds, by allowing both description logic constraints and existential rules, while maintaining the decidability of QA? This paper studies to what extent both paradigms can be combined, by looking for classes of constraints with decidable QA over relational schemas of *arbitrary arity* that can 1. express non-trivial existential rules over any relation in the schema and 2. assert expressive constraints (e.g., in $\mathcal{ALCQIb}$) on the *arity-two subschema* — the subset of the relations of arity one and two within the schema

Our first results (Section 3) are negative: we show that arity-two languages featuring functionality constraints on the arity-two subschema may lead to undecidable QA when combined with even very simple acyclic rules (source-to-target TGDs, S2T), or with the simplest existential rules that export two variables (frontier-two inclusion dependencies, ID[2]). More surprisingly, undecidability can occur with rules exporting only a single variable, the class of *frontier-one dependencies* FR[1] of [Baget *et al.*, 2009]. We say the existential rule languages S2T, ID[2], FR[1] are *destructive of arity-two QA*.

We then show (Section 4) that by restricting FR[1] slightly, imposing that the head of the rules have a certain tree shape (denoted "non-looping"), we can obtain a class of existential rules that can be combined with expressive constraints on the arity-two schema while maintaining decidable QA (we call this *not destructive*). The reduction proceeds in two steps. We first handle rules with tree-shaped bodies, via a direct rewriting technique to constraints on an arity-two encoding of the schema. Second, we handle rules with non-tree-shaped bodies, showing that the bodies can be soundly replaced by a tree-shaped approximation. Soundness is proven by extending the technique of "treeification" used previously in many modal and guarded logics (e.g., [Bárány *et al.*, 2014]), showing that models of the constraints can be "unraveled" to be tree-shaped.

We go on to study (Section 5) the addition of functional dependencies (FDs), a well-known generalization of description logic functionality constraints to arbitrary arity. QA with existential rules and FDs is generally undecidable unless their interaction with the existential rules is controlled, e.g., by imposing the non-conflicting condition [Calì *et al.*, 2012b]. We show that FDs can be added to our existential rules while maintaining decidable QA with the arity-two constraints, as long as the non-conflicting condition is satisfied. As in the standard non-conflicting setting, we show that the FDs can always be satisfied unless the initial facts violate them. We prove this by modifying the unraveling argument.

Our results have the advantage that QA for our combined constraints reduces to QA on an arity-two schema; hence, existing QA algorithms for rich description logics could be extended to arbitrary arity signatures with expressive constraints.

Related work. A great deal of research has centered around the integration of DLs with Datalog-style rules, including work as early as the 1990's, when the languages AL-Log [Donini *et al.*, 1991] and CARIN [Levy and Rousset, 1998] were introduced. AL-Log links Horn rules with concepts from a description logic terminology, while the later language CARIN provides a broader framework allowing both concepts and roles from a terminology to appear in rules. [Levy and Rousset, 1998] provides both entailment algorithms for CARIN and undecidability results exploring the borderline for combining rules and DLs.

Datalog rules, however, unlike the existential rules that we consider in this work, do not allow existential quantification in the head, so they cannot assert the existence of higher-arity facts on fresh elements.

Another approach to combination are description logics that support higher-arity relations directly. Languages such as $\mathcal{DLR}_{reg}$ [Calvanese *et al.*, 2008] give some support for higher arity while retaining a DL-style syntax. Unlike them, we support existential rules with cyclic bodies that cannot be encoded in $\mathcal{DLR}_{reg}$, as well as arbitrary higher-arity functional dependencies that go beyond DL-expressible functionality assertions. On the other hand, we do not support some features of $\mathcal{DLR}_{reg}$, such as regular expression on role paths. Indeed, we do not consider the interaction of rules with DLs supporting transitivity and other recursion mechanisms [Glimm *et al.*, 2008], focusing instead only on first-order-expressible constraints given by decidable DLs and existential rules.

2 Preliminaries

Signatures, facts, queries. A *signature* σ consists of *relation names* (e.g. R) and an associated *arity* (e.g. $|R|$). We write σ as $\sigma_{\leq 2} \sqcup \sigma_{>2}$, containing respectively the relations of arity ≤ 2 and the *higher-arity* relations with arity > 2. An *atom* $R(\mathbf{x})$ consists of a relation name R and an $|R|$-tuple $\mathbf{x}$ of variables. A σ-*fact* (or just *fact* when σ is clear from context) is a conjunction of atoms using relations in σ. A Boolean *conjunctive query* (or CQ) is an existentially quantified conjunction of atoms. In this paper we assume for simplicity that CQs are Boolean, i.e., have no free variables, and we disallow constants. This is without loss of generality: for non-Boolean queries we can enumerate all possible assignments, and constants can be encoded with fresh unary relations.

Constraints, QA. We consider constraints that are formulae in function-free and constant-free first-order logic (FO), on the signature σ. A σ-*interpretation* $\mathcal{I}$ (or just *interpretation*) consists of a *domain* $\mathrm{dom}(\mathcal{I})$ and an *interpretation function* $\cdot^{\mathcal{I}}$ mapping each relation R of σ to a set $R^{\mathcal{I}}$ of $|R|$-tuples of $\mathrm{dom}(\mathcal{I})$. The definition of $\mathcal{I}$ *satisfying* a FO formula ϕ, written $\mathcal{I} \models \phi$, is standard. A *witness* $\mathcal{W}$ of F in $\mathcal{I}$ is an interpretation that maps each relation R to the tuples in $R^{\mathcal{I}}$ obtained by substituting the atoms of F using some variable binding $\mathbf{w}$ such that $\mathcal{I} \models F(\mathbf{w})$.

We study the *query answering* problem (QA): given a fact F, a set of constraints Σ, and a CQ q, decide the validity of $\forall \mathbf{x}\ (F(\mathbf{x}) \wedge \Sigma \to q)$; that is, whether F and Σ entail q. In this case, we write $F \wedge \Sigma \models q$. The *combined complexity* of QA, for a fixed class of constraints, is the complexity of deciding it when all of F, Σ (in the constraint class) and q are given as input. If we assume that Σ and q are fixed, and only F is given as input, then we define instead the *data complexity*.

The QA problem above allows arbitrary FO constraint classes. Below we present two kinds of integrity constraints that are known to enjoy decidable QA.

Existential rules. An *existential rule* (or *tuple-generating dependency*, or TGD) is a logical constraint of the form $\forall \mathbf{x}\ (\phi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}', \mathbf{y}))$, with $\mathbf{x}' \subseteq \mathbf{x}$, where the *body* ϕ and *head* ψ are conjunctions of atoms. Equality atoms and constants are disallowed. For brevity, in rules we often omit the quantification on $\mathbf{x}$ and write '$\wedge$' as a comma. A rule is *single-head* if its head consists of only one atom.

QA is undecidable for general rules (following from [Beeri and Vardi, 1981]). One class of rules with decidable QA are those satisfying *acyclicity* conditions. We will show negative results for one of the most restrictive classes, the class $\mathsf{S2T}$ of *source-to-target TGDs*, where σ is partitioned as $\sigma = \sigma_S \sqcup \sigma_T$, the bodies of all rules only use relations in σ_S, and the heads only use relations in σ_T. Our results on $\mathsf{S2T}$ extend to more permissive acyclicity conditions, such as those mentioned in the introduction.

A second class of decidable rules guarantees that it suffices to consider bounded-treewidth interpretations, usually because of constraints on the rule bodies. We focus on the class $\mathsf{FR}[1]$ of *frontier-one rules*, following [Baget *et al.*, 2009]: the *frontier* of a rule is the set $\mathbf{x}'$ of variables that occur both in the body and the head, and a rule is frontier-one if $|\mathbf{x}'| = 1$. The class of *inclusion dependencies* ID imposes that the head and body are single atoms where each variable is used only once and that the frontier is not empty, and we will focus on the class $\mathsf{ID}[2]$ of the inclusion dependencies with frontier size 2. QA is decidable for $\mathsf{FR}[1]$ [Baget *et al.*, 2009]. For ID it is decidable and has PTIME data complexity [Calì *et al.*, 2003b].

Existential rules can be augmented with *functional dependencies* (FDs), which are variants of existential rules that impose equalities. Writing $\forall \mathbf{x} = \forall x_1 \cdots \forall x_n$ and similarly for $\mathbf{y}$, an FD on the relation R is of the form:

$$\forall \mathbf{x}\mathbf{y}\ (R(x_1,\ldots,x_n) \wedge R(y_1,\ldots,y_n) \wedge \textstyle\bigwedge_{l \in L} x_l = y_l) \rightarrow x_r = y_r$$

for some $1 \le r \le |R|$ and some subset $L \subseteq \{1,\ldots,|R|\}$ which we call the *determiner* of the FD. QA is undecidable when combining existential rules and arbitrary FDs, for instance it is undecidable for $\mathsf{ID}[2]$ and FDs [Calì *et al.*, 2003a].

Arity-two constraints. The second kind of tame constraints are *arity-two constraints*, which are *only defined on* $\sigma_{\le 2}$. The most general such language that we study is the *two-variable guarded fragment with counting quantifiers*, GC^2 [Kazakov, 2004]. GC^2 is the smallest class of constant-free FO formulae with at most two variables, containing all atoms for $\sigma_{\le 2}$ relations, closed under Boolean connectives, under *guarded* universal and existential quantification, and under *number quantifications*: if $\phi(x,y)$ is a GC^2 formula and $A(x,y)$ is an arity-two atom with two free variables (the *guard*), then $\exists^{\ge n} y\, A(x,y) \wedge \phi(x,y)$ and $\exists^{< n} y\, A(x,y) \wedge \phi(x,y)$ are formulae, where n is an integer. QA for GC^2 is decidable and its data complexity is in co-NP [Pratt-Hartmann, 2009].

Description logics (DLs) are arity-two constraint languages. Examples of DLs are DL-Lite [Calvanese *et al.*, 2005], a lightweight DL often used in the context of ontology-based data access, and $\mathcal{ALCQIb}$ [Tobies, 2001], a more expressive DL that can make full use of *number restrictions*, a useful feature in practice. Both DL-Lite and $\mathcal{ALCQIb}$ can assert *concept inclusions* like $C \sqsubseteq C'$, where C and C' are *concepts* (arity 1 relations), meaning that C' holds whenever C does; and *functionality assertions* $\mathrm{funct}(R)$, where R is a *role* (an arity 2 relation), corresponding to $\forall x\ \exists^{\le 1} y\ R(x,y)$ in GC^2, or to the FD: $\forall x_1 x_2 y_1 y_2\ R(x_1,x_2) \wedge R(y_1,y_2) \wedge x_1 = y_1 \rightarrow x_2 = y_2$. Despite its expressiveness, $\mathcal{ALCQIb}$ can still, as DL-Lite, be captured by GC^2, which implies decidable QA.

Roles and concepts can be *atomic* (i.e., from $\sigma_{\le 2}$) or defined using constructors; we give some examples from $\mathcal{ALCQIb}$. The *inverse* R^- of an atomic role R is such that $R^-(b,a)$ holds whenever $R(a,b)$ does. An *intersection* of roles, which is written $R_1 \sqcap \cdots \sqcap R_n$, holds for (a,b) whenever $R_i(a,b)$ holds for all $1 \le i \le n$. $\top$ and $\bot$ are the *true* and *false* concepts. The *intersection* of concepts $C_1,\ldots,C_n$, written $C_1 \sqcap \cdots \sqcap C_n$, holds whenever each of the C_i does. The *negation* $\neg C$ of a concept C holds for elements where C does not hold. An *existential concept* $\exists R.C$ for a role R and concept C holds for every element a such that $\exists b\ R(a,b) \wedge C(b)$ does. Note that many of these features (e.g., functionality assertions and negation) cannot be expressed as existential rules.

Combining constraint classes. For any class CL of existential rules, we call CL *non-destructive* (of arity-two QA) if QA is decidable for the class $\mathsf{CL} \wedge \mathsf{GC}^2$ of conjunctions of constraints of CL (on σ) and of constraints of GC^2 (on $\sigma_{\le 2}$). Otherwise, we call CL *destructive*.

3 Negative Results for Combination

We now present classes of existential rules which have decidable QA but are destructive. First, we observe that even the simplest class of rules that ensures decidability based on chase termination, the class $\mathsf{S2T}$ of source-to-target TGDs, is destructive. This is not so surprising, since the arbitrary constraints on the arity-two signature may add dependencies that are not source-to-target.

Theorem 3.1. $\mathsf{S2T}$ *is destructive of arity-two QA, even when the whole* σ *has arity two and there is no query (i.e., this is just the* satisfiability problem *asking whether the fact and constraints are satisfiable).*

Thus we move on to classes of existential rules that are decidable because of guardedness assumptions.

We first observe that the class $\mathsf{ID}[2]$ of frontier-two inclusion dependencies is destructive of arity-two QA. In fact, functionality assertions on the binary relations are sufficient to get undecidability, because they can be lifted to functionality assertions on higher-arity relations using $\mathsf{ID}[2]$. Thus, following a standard reduction from QA to entailment of dependencies as in [Calì *et al.*, 2003a], we can use the undecidability of entailment for $\mathsf{ID}[2]$ and FDs (Theorem 2 of [Mitchell, 1983], which we adapt slightly) and prove the following:

Theorem 3.2. $\mathsf{ID}[2]$ *is destructive of arity-two QA. In particular, QA is undecidable for* $\mathsf{ID}[2] \wedge \mathcal{D}$*, for any DL* $\mathcal{D}$ *(such as DL-Lite) featuring functionality assertions.*

More surprisingly, frontier-one rules $\mathsf{FR}[1]$ are destructive of arity-two QA, even though they can only export a single variable, and this holds even when the whole σ has arity two. The reason is that $\mathsf{FR}[1]$ may be more expressive than GC^2 as it can disobey the two-variable restriction.

Theorem 3.3. $\mathsf{FR}[1]$ *is destructive of arity-two QA, even when the whole* σ *has arity two and there is no query.*

This motivates the search for more restricted existential rule classes which could be non-destructive of arity-two QA.

4 From Existential Rules to Arity-Two

We will focus on the subclass of frontier-one rules whose heads do not contain non-trivial *Berge cycles* [Fagin, 1983].

Definition 4.1. *A* Berge cycle *in a conjunction of atoms* Ψ *is a sequence* $A_1, x_1, A_2, x_2, \ldots, A_n, x_n$ *of length* $n > 1$ *where the* x_i *are pairwise distinct variables, the* A_i *are pairwise distinct atoms of* Ψ*, and every* x_i *occurs in atoms* A_i *and* A_{i+1} *(with addition modulo* n*, so* x_n *occurs in* A_1*).*

We say Ψ *is* non-looping *if there is no Berge cycle of length above* 2*, and no Berge cycle that contains an atom of* $\sigma_{>2}$*.*

We define the head-non-looping $\mathsf{FR}[1]^{\mathsf{Hnl}}$ *subclass of* $\mathsf{FR}[1]$ *rules whose heads are non-looping. In particular, single-head* $\mathsf{FR}[1]$ *rules are always head-non-looping.*

Example 4.2. *Rules* $A(x) \to \exists yz\ R(x,y), S(y,z), T(z,x)$ *and* $B(y) \to \exists yz\ R(x,y), U(x,y,z)$ *are not in* $\mathsf{FR}[1]^{\mathsf{Hnl}}$*. However,* $A(x) \to \exists y\ V(x,x,y,y)$ *and* $B(x) \to \exists y\ R(x,y), S(x,y), R(y,x)$ *are in* $\mathsf{FR}[1]^{\mathsf{Hnl}}$*.*

We claim that head-non-looping rules are non-destructive, in contrast with general frontier-one rules (Theorem 3.3):

Theorem 4.3. $\mathsf{FR}[1]^{\mathsf{Hnl}}$ *is not destructive of arity-two QA.*

Of course, this means that QA is decidable for $\mathsf{FR}[1]^{\mathsf{Hnl}} \wedge \mathcal{D}$, for any DL $\mathcal{D}$ expressible in GC^2, such as $\mathcal{ALCQIb}$. The rest of this section proves the theorem and addresses complexity.

Shredding. Our proof of Theorem 4.3 translates the $\mathsf{FR}[1]^{\mathsf{Hnl}}$ rules to arity-two constraints, using a common way to represent general relational databases in a binary relational store, which we call *shredding*: we represent an n-ary relation by a set of binary relations giving the link from each tuple (materialized as an element) to its attributes. We present first the translation of the signature σ to its shredded arity-two signature σ_{S}, and the constraints imposed on σ_{S}-interpretations to ensure that they can be decoded back to σ-interpretations. Second, we explain how to shred facts and CQs.

Definition 4.4. *The* shredded signature σ_{S} *of a signature* σ *consists of* $\sigma_{\leq 2}$*, a unary relation* Elt*, and, for each* $R \in \sigma_{>2}$*, a unary relation* A_R *and binary relations* R_i *for* $1 \leq i \leq |R|$*.*

The well-formedness constraints *of* σ_{S}*, written* $\mathsf{wf}(\sigma_{\mathrm{S}})$*, are the following DL constraints (they are* $\mathcal{ALCQIb}$*-expressible):*

- $C \sqsubseteq \mathrm{Elt}$ *for every unary relation* C *of* $\sigma_{\leq 2}$
- $\exists R.\top \sqsubseteq \mathrm{Elt}$ *and* $\exists R^-.\top \sqsubseteq \mathrm{Elt}$ *for all binary* R *of* $\sigma_{\leq 2}$

and the following, where $R \neq S$ *are in* $\sigma_{>2}$ *and* $1 \leq i \leq |R|$*:*

- $\exists R_i.\top \sqsubseteq A_R$ *and* $\exists R_i^-.\top \sqsubseteq \mathrm{Elt}$
- $\mathrm{Elt} \sqcap A_R \sqsubseteq \bot$ *and* $A_R \sqcap A_S \sqsubseteq \bot$
- $A_R \sqsubseteq \exists R_i.\top$ *and* $\mathrm{funct}(R_i)$

The *shredding* $\mathrm{SHR}(F)$ of a σ-fact F is the σ_{S}-fact obtained by adding the atom $\mathrm{Elt}(x)$ for each variable x of F and replacing each atom $R(\mathbf{x})$ of F when $R \in \sigma_{>2}$ by the atoms $A_R(t)$ and $R_i(t, x_i)$ for $1 \leq i \leq |R|$, for t a fresh variable. The *shredding* $\mathrm{SHR}(q)$ of a CQ q is similarly defined.

Example 4.5. *Considering CQ* $q : \exists xyz\ U(x), R(x,y), S(z,z,x)$*, we define* $\mathrm{SHR}(q)$ *as:* $\exists xyzt\ \mathrm{Elt}(x), \mathrm{Elt}(y), \mathrm{Elt}(z), U(x), R(x,y), A_R(t), S_1(t,z), S_2(t,z), S_3(t,x)$*.*

Fully-non-looping. The interesting part is to define the shredding of $\mathsf{FR}[1]^{\mathsf{Hnl}}$ rules. We first restrict to the class of *fully-non-looping* rules, $\mathsf{FR}[1]^{\mathsf{Fnl}}$, whose head *and body* are non-looping. We show that $\mathsf{FR}[1]^{\mathsf{Fnl}}$ can be directly shredded to GC^2. We will later move from $\mathsf{FR}[1]^{\mathsf{Fnl}}$ to $\mathsf{FR}[1]^{\mathsf{Hnl}}$.

For any existential rule $\tau : \forall \mathbf{x}\ \phi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}', \mathbf{y})$ with $\mathbf{x} \subseteq \mathbf{x}'$, we define its *shredding* $\mathrm{SHR}(\tau)$ as the existential rule $\forall \mathbf{xt}\ (\mathrm{SHR}(\phi(\mathbf{x}))) \to \exists \mathbf{yt}'\ (\mathrm{SHR}(\psi(\mathbf{x}', \mathbf{y})))$, where $\mathbf{t}$ and $\mathbf{t}'$ are the fresh elements introduced in the shredding of ϕ and ψ respectively. We claim the following:

Lemma 4.6. *For any* $\mathsf{FR}[1]^{\mathsf{Fnl}}$ *rule* τ*,* $\mathrm{SHR}(\tau)$ *can be translated in PTIME to a* GC^2 *sentence on* σ_{S}*.*

Example 4.7. *For brevity, this example ignores the* Elt *and* A_R *atoms when shredding. Consider the* $\mathsf{FR}[1]^{\mathsf{Fnl}}$ *rule:*

$U(u), T(u,x), S(x) \to \exists yz\ T(x,y), U(y), R(x,x,z,z)$

Its shredding is expressible in GC^2 *(and even in* $\mathcal{ALCQIb}$*):*

$(\exists T^-.U) \sqcap S \sqsubseteq (\exists T.U) \sqcap (\exists (R_1^- \sqcap R_2^-).(\exists (R_3 \sqcap R_4).\top))$

By contrast, consider the following rule in $\mathsf{FR}[1] \backslash \mathsf{FR}[1]^{\mathsf{Hnl}}$*:*

$U(x) \to \exists yz\ R(x,y), S(x,y,z)$

Its shredding is as follows; it is not GC^2*-expressible:*

$U(x) \to \exists yzt\ R(x,y), S_1(t,x), S_2(t,y), S_3(t,z)$

In the general case, the GC^2 rewriting of Lemma 4.6 is obtained in PTIME by seeing the body and head of $\mathrm{SHR}(\tau)$ as a tree, which is possible because τ is fully-non-looping.

It is now easy to show the following general result:

Proposition 4.8 (Shredding). *For any fact* F*,* GC^2 *constraints* Σ*, existential rules* Δ *and CQ* q*, the following are equivalent:*

- $F \wedge \Sigma \wedge \Delta \models q$*;*
- $\mathrm{SHR}(F) \wedge \Sigma \wedge \mathrm{SHR}(\Delta) \wedge \mathsf{wf}(\sigma_{\mathrm{S}}) \models \mathrm{SHR}(q)$*.*

Thus, from Lemma 4.6, as $\mathrm{SHR}(F)$, $\mathrm{SHR}(\Delta)$, σ_{S}, $\mathsf{wf}(\sigma_{\mathrm{S}})$, and $\mathrm{SHR}(q)$ can be computed in PTIME following their definition, we deduce the following, in the case of $\mathsf{FR}[1]^{\mathsf{Fnl}}$:

Corollary 4.9. *QA for* GC^2 *and* $\mathsf{FR}[1]^{\mathsf{Fnl}}$ *constraints can be reduced to QA for* GC^2 *in PTIME; further, when the constraints and query are fixed in the input, they also are in the output, so data complexity bounds for* GC^2 *QA are preserved.*

This concludes the proof of Theorem 4.3 for $\mathsf{FR}[1]^{\mathsf{Fnl}}$ constraints. It further implies that QA for GC^2 and $\mathsf{FR}[1]^{\mathsf{Fnl}}$ has co-NP-complete data complexity, like GC^2, [Pratt-Hartmann, 2009], and the combined complexity is the same as for GC^2.

Note that, although QA for GC^2 is decidable, we know of no realistic implementations. Our translation could however reduce instead to arity-two QA with constraints in DLs such as $\mathcal{ALCQIb}$, if we impose impose additional minor restrictions on the $\mathsf{FR}[1]^{\mathsf{Fnl}}$ rules (e.g., no atom of the form $S(x,x)$). For simplicity, however, we focus in the sequel on reductions to decidable QA on arity-two (i.e., translating to GC^2) rather than investigating which restrictions would ensure that the output of our translations can be expressed in particular DLs.

Head-non-looping. We now extend the claim to $\mathsf{FR}[1]^{\mathsf{Hnl}}$ rather than $\mathsf{FR}[1]^{\mathsf{Fnl}}$. The idea is that we rewrite $\mathsf{FR}[1]^{\mathsf{Hnl}}$ rules to $\mathsf{FR}[1]^{\mathsf{Fnl}}$ by *treeifying them*, considering all possible fully-non-looping rules that they imply, and all possible ways that they can match on the parts of the interpretations that satisfy the fact. To formalize this, we assume that we have added to the fact F one atom $P_x(x)$ for each variable x of F, where each P_x is a fresh unary relation. We then define:

Definition 4.10. *The* treeification *on fact* F *of a* $\mathsf{FR}[1]^{\mathsf{Hnl}}$ *rule* $\tau : \forall \mathbf{x}\ (\phi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(x_{\mathrm{f}}, \mathbf{y}))$*, where* $x_{\mathrm{f}} \in \mathbf{x}$ *is the frontier variable, is the conjunction* $\mathrm{TR}_F(\tau)$ *of* $\mathsf{FR}[1]^{\mathsf{Fnl}}$ *rules defined as follows:*

- *consider every mapping f from $\mathbf{x}$ to itself, and let $f(\tau)$ be obtained from τ by renaming all variables in $\mathbf{x}$ with f;*
- *for every such $f(\tau)$, consider every $\mathbf{x}' \subseteq \mathbf{x}$ and every mapping g from $\mathbf{x}'$ to the variables of F, and construct $g(f(\tau))$ by replacing every occurrence of each $x \in \mathbf{x}'$ in $\phi(\mathbf{x})$ by fresh variables $x_1, \ldots, x_n$, and adding the facts $P_{g(x)}(x_i)$ for all $x \in \mathbf{x}'$ and all i (if $x_\mathrm{f} \in \mathbf{x}'$, also replace x_f in $\psi(x_\mathrm{f}, \mathbf{y})$ by one of its copies);*
- *if $g(f(\tau))$ is fully-non-looping, add it to $\mathrm{TR}_F(\tau)$.*

Example 4.11. *Consider a fact F and the following rule τ:*

$R(x,y), S(y,z), T(z,w), U(w,x) \to A(x)$

The treeification $\mathrm{TR}_F(\tau)$ contains the rule:

$R(x,y), S(y,z), T(z,y), U(y,x) \to A(x)$.

Consider the rule $\tau' : R(x,y), S(y,x,x) \to A(x)$, and a fact F containing variable z. Then $\mathrm{TR}_F(\tau')$ contains:

$R(x_1,y), S(y,x_2,x_3), P_z(x_1), P_z(x_2), P_z(x_3) \to A(x_1)$

We now claim:

Proposition 4.12. *For any fact F, GC^2 constraints Σ, $\mathsf{FR}[1]^{\mathrm{Hnl}}$ rules Δ and CQ q, the following are equivalent:*
- *$F \wedge \Sigma \wedge \Delta \models q$;*
- *$F \wedge \Sigma \wedge \mathrm{TR}_F(\Delta) \models q$.*

This proposition implies that QA for $\mathsf{FR}[1]^{\mathrm{Hnl}}$ and GC^2 can be reduced to QA for $\mathsf{FR}[1]^{\mathrm{Fnl}}$ and GC^2, which is decidable by the Shredding Proposition, proving Theorem 4.3.

To prove Proposition 4.12, for the first direction, if $F \wedge \Sigma \wedge \Delta \not\models q$, one can show that all of the fresh unary relations P_x in an interpretation of $F \wedge \Sigma \wedge \Delta \wedge \neg q$ can be assumed to be interpreted by one tuple. One then shows that Δ implies $\mathrm{TR}_F(\Delta)$ on such interpretations. For the other direction, assuming that $F \wedge \Sigma \wedge \mathrm{TR}_F(\Delta) \not\models q$, the Shredding Proposition implies that there is a σ_S-interpretation $\mathcal{J}$ of $\Theta := \Sigma \wedge \mathrm{SHR}(\mathrm{TR}_F(\Delta)) \wedge \mathsf{wf}(\sigma_\mathrm{S})$, $\neg q' := \neg\mathrm{SHR}(q)$, and the existential closure of $F' := \mathrm{SHR}(F)$. We apply an unraveling argument to show that $\mathcal{J}$ can be made *cycle-free*:

Definition 4.13. *The* Gaifman graph *$\mathcal{G}(\mathcal{I})$ of an interpretation $\mathcal{I}$ is the undirected graph on $\mathrm{dom}(\mathcal{I})$ connecting any two elements co-occurring in a tuple of $\mathcal{I}$. Given a fact F, an interpretation $\mathcal{I}$ is* cycle-free except for *F if F has a witness $\mathcal{W}$ in $\mathcal{I}$ such that any cycle of $\mathcal{G}(\mathcal{I})$ is only on elements of $\mathrm{dom}(\mathcal{W})$.*

Lemma 4.14 (Unraveling). *For any σ_S-fact F', GC^2 constraints Θ, and CQ q', if $(\exists \mathbf{x}\mathbf{t}\ F'(\mathbf{x},\mathbf{t})) \wedge \Theta \wedge \neg q'$ is satisfiable then it has an interpretation which is cycle-free except for F'.*

Letting $\mathcal{J}'$ be the *unraveling* of our interpretation $\mathcal{J}$ (obtained by the Unraveling Lemma), we can then "unshred" $\mathcal{J}'$ back to a σ-interpretation $\mathcal{I}$:

Definition 4.15. *The* unshredding *$\mathcal{I}$ of a σ_S-interpretation $\mathcal{J} \models \mathsf{wf}(\sigma_\mathrm{S})$ is obtained by setting $R^\mathcal{I} := R^\mathcal{J}$ for $R \in \sigma_{\leq 2}$, and, for all $R \in \sigma_{>2}$ and $t \in A_R^\mathcal{J}$, creating the tuple $\mathbf{a} \in R^\mathcal{I}$ such that $(t, a_i) \in R_i^\mathcal{J}$ for all $1 \leq i \leq |R|$.*

As in the proof of the Shredding Proposition, we can show that the unshredding $\mathcal{I}$ is well-defined and satisfies the unshredded constraints $(\exists \mathbf{x}\ F(\mathbf{x})) \wedge \Sigma \wedge \mathrm{TR}_F(\Delta) \wedge \neg q$. Further, we show that it satisfies Δ and not just $\mathrm{TR}_F(\Delta)$, because a match of a $\mathsf{FR}[1]^{\mathrm{Hnl}}$ rule τ in $\mathcal{I}$ must be a match of $\mathrm{TR}_F(\tau)$; otherwise the match would witness that $\mathcal{J}'$ was not cycle-free:

Lemma 4.16 (Soundness). *For a σ-fact F, $\mathsf{FR}[1]^{\mathrm{Hnl}}$ rule τ and σ_S-interpretation $\mathcal{J}$, if $\mathcal{J}$ satisfies $\mathrm{SHR}(\mathrm{TR}_F(\tau))$ and is cycle-free except for $\mathrm{SHR}(F)$, then the unshredding $\mathcal{I}$ of $\mathcal{J}$ satisfies τ.*

We conclude by sketching the proof of the Unraveling Lemma, which follows [Kazakov, 2004; Pratt-Hartmann, 2009]. From an interpretation $\mathcal{J}$ of $(\exists \mathbf{x}\mathbf{t}\ F'(\mathbf{x},\mathbf{t})) \wedge \Theta \wedge \neg q'$, for all $u \neq v$ in $\mathrm{dom}(\mathcal{J})$ co-occurring in some tuple of $\mathcal{J}$, we call a *bag* the interpretation with domain $\{u,v\}$ consisting of the tuples of $\mathcal{J}$ mentioning only u,v. We build a graph G over the bags by connecting bags whose domain shares one element. We pick a witness $\mathcal{W}$ of F' in $\mathcal{J}$ and merge in the *fact bag* all bags whose domain is included in $\mathrm{dom}(\mathcal{W})$.

An unraveling is a tree T of bags obtained by unfolding G starting at the fact bag, which is preserved as-is. Each bag b of T except the fact bag has a domain containing two elements: one of them occurs exactly in b, its siblings and its parent; the other occurs exactly in b and its children (it is *introduced* in b). We see T as an interpretation formed of the union of its bags.

We construct T from G inductively. For any bag b in T corresponding to a bag b' in G, construct the children of b as follows. For each bag b'' adjacent to b' in G, if b' and b'' share the element corresponding to the element u introduced in b, create an isomorphic copy of b'' as a child of b in T, whose domain is u plus a fresh element, and perform the unraveling process recursively on the children.

It can be shown that the unraveling operation preserves GC^2 constraints, the fact F', and the negated CQ $\neg q'$. As T is a tree, the interpretation it describes is cycle-free (except for the witness $\mathcal{W}$, because we copied the fact bag as-is).

Complexity. Proposition 4.12 gives a reduction from $\mathsf{FR}[1]^{\mathrm{Hnl}}$ and GC^2 QA to $\mathsf{FR}[1]^{\mathrm{Fnl}}$ to GC^2 QA, but its output is of exponential size in the input, because of treeification. Hence, letting $\mathfrak{f}(n)$ bound the size of the output of our reduction given an input of size n, and letting $\mathfrak{g}(n)$ bound the combined complexity of GC^2 QA, we have shown an upper bound of $\mathfrak{g}(\mathfrak{f}(n))$ for QA for $\mathsf{FR}[1]^{\mathrm{Hnl}}$ and GC^2.

Further, treeification rewrites the rules in a fact-dependent way, so, unlike the previous case of $\mathsf{FR}[1]^{\mathrm{Fnl}}$ and GC^2 QA, data complexity bounds for GC^2 QA do not imply data complexity bounds for $\mathsf{FR}[1]^{\mathrm{Hnl}}$ and GC^2 QA.

5 Adding Functional Dependencies

The previous section showed that the language of head-non-looping frontier-one rules is not destructive of GC^2 QA. However, another kind of rules that we would want to support on higher-arity relations are functional dependencies (FDs).

It is well-known that QA is undecidable for, e.g., $\mathsf{ID}[2]$ and arbitrary FDs [Calì *et al.*, 2003a], so such constraints are trivially destructive. As it turns out, undecidability also holds for $\mathsf{FR}[1]^{\mathrm{Hnl}}$ rules and FDs; in fact, even for single-head $\mathsf{FR}[1]$ rules and FDs:

Theorem 5.1. *QA is undecidable for FDs and single-head frontier-one rules, even if all FDs have a determiner of size* 1.

However, for certain kinds of existential rules and FDs, QA is known to be decidable: this is in particular the case of *non-conflicting* rules and FDs [Calì *et al.*, 2012b]:

Definition 5.2. *We say that a* single-head *existential rule* τ *is* non-conflicting *with respect to a set of FDs* Φ *if, letting* $A = R(\mathbf{z})$ *be the head atom of* τ*, letting* S *be the subset of* $\{1,\ldots,|R|\}$ *such that* z_i *is a frontier variable iff* $i \in S$*:*

- *No strict subset of* S *is the determiner of an FD in* Φ*;*
- *If* S *is exactly the determiner of an FD of* Φ*, then all existentially quantified variables in* A *occur only once.*

Note that this requires rules to be *single-head*, and thus head-non-looping. Our result with respect to adding FDs is:

Theorem 5.3. *Non-conflicting frontier-one rules and FDs are non-destructive of arity-two QA.*

In particular, single-head frontier-one rules and FDs are non-destructive of arity-two QA if all variables in the head atom of rules are assumed to have only one occurrence, as this simple sufficient condition implies the non-conflicting condition.

To prove the theorem, we assume without loss of generality that we only have FDs on higher-arity relations, as we can write them in GC^2 otherwise. We cannot shred the FDs, as they would translate to a functionality assertion for the path, e.g., $R_i^- \circ R_j$, which is not expressible in GC^2 (and not even in expressive DLs such as $\mathcal{SROIQ}$ [Horrocks *et al.*, 2006]). However, we can show that, thanks to the non-conflicting requirement, FDs can always be made to hold on interpretations, as long as they hold on a witness of the fact.

Proposition 5.4. *For any* GC^2 *constraints* Σ*, non-conflicting frontier-one rules* Δ*, FDs* Φ *on* $\sigma_{>2}$*,* σ*-fact* F*, and CQ* q*, if there is an interpretation* $\mathcal{I}$ *satisfying* $\Theta := (\exists \mathbf{x}\ F(\mathbf{x})) \wedge \Sigma \wedge \Delta \wedge \neg q$ *and there is a witness* $\mathcal{W}$ *of* F *in* $\mathcal{I}$ *satisfying* Φ*, then* $\Theta \wedge \Phi$ *is satisfiable.*

We first prove Proposition 5.4. As in Section 4, consider the treeification $\mathrm{TR}_F(\Delta)$: it is still non-conflicting as treeification only affects rule bodies. Use the Shredding Proposition to obtain an interpretation $\mathcal{J}$ of $\neg q' := \neg\mathrm{SHR}(q)$, $\Theta := \Sigma \wedge \mathrm{SHR}(\mathrm{TR}_F(\Delta)) \wedge \mathsf{wf}(\sigma_{\mathrm{S}})$, and the existential closure of $F' := \mathrm{SHR}(F)$. By our hypothesis about the existence of a witness, we can assume that $\mathcal{J}$ has a witness $\mathcal{W}$ of F' whose unshredding satisfies Φ.

In the previous section, we used the Unraveling Lemma to show that $\mathcal{J}$ could be assumed to be cycle-free. We now modify the lemma to additionally ensure the following property on $\mathcal{J}$, which will forbid FD violations in its unshredding:

Definition 5.5. *Given a set of FDs* Φ *on* $\sigma_{>2}$*, a* σ_{S}*-interpretation* $\mathcal{J}$*, and a witness* $\mathcal{W}$ *of a fact in* $\mathcal{J}$*, we call* $\mathcal{J}$ FD-safe except for $\mathcal{W}$ *if for every* $a \in \mathrm{dom}(\mathcal{J})$*, for any* $R \in \sigma_{>2}$ *and FD determiner* P *of* R *in* Φ*, considering each* $t \in \mathrm{dom}(\mathcal{J})$ *such that* $(t,a) \in R_i^{\mathcal{J}}$ *for every* $i \in P$*, either there is at most one such* t *or all are in* $\mathrm{dom}(\mathcal{W})$*.*

FD-safety is useful for the following reason:

Lemma 5.6. *For any set of FDs* Φ *on* $\sigma_{>2}$*, for any* σ_{S}*-interpretation* $\mathcal{J}$ *which is cycle-free and FD-safe except for a witness* $\mathcal{W}$*, if the unshredding of* $\mathcal{W}$ *satisfies* Φ*, then the unshredding of* $\mathcal{J}$ *satisfies* Φ*.*

We now claim a variant of the Unraveling Lemma:

Lemma 5.7 (FD-aware unraveling). *Let* Σ *be a* GC^2 *constraint,* F *a* σ*-fact,* q *a CQ,* Δ *non-conflicting frontier-one rules and* Φ *a set of FDs on* $\sigma_{>2}$*. Let* $\mathcal{J}$ *be an interpretation satisfying* $\Theta := (\exists \mathbf{xt}\ \mathrm{SHR}(F)(\mathbf{x},\mathbf{t})) \wedge \Sigma \wedge \mathrm{SHR}(\mathrm{TR}_F(\Delta)) \wedge \mathsf{wf}(\sigma_{\mathrm{S}}) \wedge \neg\mathrm{SHR}(q)$*, and* $\mathcal{W}$ *a witness of* $\mathrm{SHR}(F)$ *in* $\mathcal{J}$*. Then there is an interpretation* $\mathcal{J}'$ *satisfying* Θ *such that* $\mathcal{W}$ *is a witness of* $\mathrm{SHR}(F)$ *in* $\mathcal{J}'$*, and* $\mathcal{J}'$ *is cycle-free and FD-safe except for* $\mathcal{W}$*.*

We prove the lemma by tweaking the unraveling process to ensure FD-safety: when creating children of each bag b in the unraveling T for neighbors of its corresponding bag b' in the bag graph G, omit some neighbors that contain shreddings of higher-arity tuples if the shared element u occurs in a strict superset of an FD determiner of Φ, and unravel differently the neighbors where u occurs exactly at a determiner. This unraveling still satisfies Σ, $\neg q'$, and the existential closure of F', and satisfies $\mathrm{SHR}(\mathrm{TR}_F(\Delta))$: the non-conflicting condition ensures that the omitted facts were not required by a rule.

We then apply the FD-aware Unraveling Lemma to $\mathcal{J}$ and consider the unshredding $\mathcal{I}$ of the result; it satisfies all necessary constraints as in Section 4, including Φ by Lemma 5.6. This proves Proposition 5.4.

We conclude by proving Theorem 5.3. We first observe that the results of Section 4 extend to a more general notion of fact that allows inequality axioms ($x \neq y$); indeed, inequalities in the fact are preserved by shredding and unshredding, and by unraveling. So Theorem 4.3 holds for such facts with inequalities, with the same complexity. Second, we enumerate all possible equalities between variables of the fact F, and for each possibility, consider the fact $F_=$ where variables are merged following the equalities, and inequalities are asserted between the remaining variables. Proposition 5.4 implies that our original entailment holds iff all the derived entailments hold where F is replaced by some $F_=$ whose canonical interpretation satisfies Φ (this can be tested in PTIME for each $F_=$). Thus we have reduced to QA for $\mathsf{FR}[1]^{\mathrm{Fnl}}$ and GC^2.

In terms of complexity, as GC^2 QA is EXPTIME-hard in combined complexity (because satisfiability for the usual two-variable guarded fragment is EXPTIME-hard [Grädel, 1999]), the additional exponential factor (from all possible $F_=$) has no impact, so the bounds of Section 4 also apply to QA for GC^2 and non-conflicting frontier-one rules and FDs.

6 Conclusion

In this paper, we have studied the impact of existential rules on the decidability of query answering for classes of arity-two constraints. We also explained (in proving Theorem 5.3) how the decidability extends when inequalities are allowed in facts.

We have limited our arbitrary arity constraints to rules, i.e., dependencies. In future work we will study how to extend our results to arbitrary arity constraint languages with more features, e.g., disjunction. We will also study what happens in the presence of constants (or nominals), which are disallowed in GC^2 (and in the rule languages we consider), but are known not to break decidability in arity-two contexts [Rudolph and Glimm, 2010; Calvanese *et al.*, 2009]. This, however, would probably require different techniques, as unraveling may create multiple copies of constants. Another question that would probably require specific tools is the study of *finite QA*, i.e., QA restricted to finite interpretations.

We are very grateful to Boris Motik and Pierre Senellart for their helpful feedback. This work was partly supported by the Télécom ParisTech Research Chair on Big Data and Market Insights and by the Engineering and Physical Sciences Research Council, UK, grants EP/G004021/1 and EP/M005852/1.

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[Baader, 2003] Franz Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.

[Baget *et al.*, 2009] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In *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 *KR*, 2010.

[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 *IJCAI*, 2011.

[Baget *et al.*, 2014] Jean-François Baget, Fabien Garreau, Marie-Laure Mugnier, and Swan Rocher. Extending acyclicity notions for existential rules. In *ECAI*, 2014.

[Bárány *et al.*, 2014] Vince Bárány, Georg Gottlob, and Martin Otto. Querying the guarded fragment. *LMCS*, 10(2), 2014.

[Beeri and Vardi, 1981] Catriel Beeri and Moshe Y Vardi. The implication problem for data dependencies. In *ICALP*. 1981.

[Calì *et al.*, 2003a] Andrea Calì, Domenico Lembo, and Riccardo Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *PODS*, 2003.

[Calì *et al.*, 2003b] Andrea Calì, Domenico Lembo, and Riccardo Rosati. Query rewriting and answering under constraints in data integration systems. In *IJCAI*, 2003.

[Calì *et al.*, 2012a] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Semantics*, 14, 2012.

[Calì *et al.*, 2012b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intel.*, 193, 2012.

[Calvanese *et al.*, 2005] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. DL-Lite: Tractable description logics for ontologies. In *AAAI*, 2005.

[Calvanese *et al.*, 2008] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Conjunctive query containment and answering under description logic constraints. *TOCL*, 9(3), 2008.

[Calvanese *et al.*, 2009] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *IJCAI*, 2009.

[Donini *et al.*, 1991] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. A hybrid system with Datalog and concept languages. In *AI*IA*, 1991.

[Fagin *et al.*, 2005] Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian Popa. Data exchange: Semantics and query answering. *TCS*, 336(1), 2005.

[Fagin, 1983] Ronald Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *JACM*, 30(3), 1983.

[Glimm *et al.*, 2008] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic SHIQ. *JAIR*, 31, 2008.

[Grädel, 1999] Erich Grädel. On the restraining power of guards. *J. Symbolic Logic*, 1999.

[Grau *et al.*, 2013] Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity notions for existential rules and their application to query answering in ontologies. *JAIR*, 47, 2013.

[Horrocks *et al.*, 2006] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In *KR*, 2006.

[Kazakov, 2004] Yevgeny Kazakov. A polynomial translation from the two-variable guarded fragment with number restrictions to the guarded fragment. In *JELIA*. 2004.

[Krötzsch and Rudolph, 2011] Markus Krötzsch and Sebastian Rudolph. Extending decidable existential rules by joining acyclicity and guardedness. In *IJCAI*, 2011.

[Lenzerini, 2002] Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS*, 2002.

[Levy and Rousset, 1998] Alon Y. Levy and Marie-Christine Rousset. Combining Horn rules and description logics in CARIN. *Artif. Intell.*, 104(1-2):165–209, 1998.

[Mitchell, 1983] John C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56(3), 1983.

[Pratt-Hartmann, 2009] Ian Pratt-Hartmann. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.*, 207(8), 2009.

[Rudolph and Glimm, 2010] Sebastian Rudolph and Birte Glimm. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *JAIR*, 39, 2010.

[Tobies, 2001] Stephan Tobies. *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, 2001.