Finite Controllability for Ontology-Mediated Query Answering of CRPQ

Diego Figueira[1] , **Santiago Figueira**[2,3] , **Edwin Pin Baque**[2,3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, France
[2]University of Buenos Aires, Argentina
[3]CONICET, Argentina

Abstract

Finite ontology mediated query answering (FOMQA) is the variant of ontology mediated query answering (OMQA) where the represented world is assumed to be finite, and thus only finite models of the ontology are considered. We study the property of finite-controllability, that is, whether FOMQA and OMQA are equivalent, for fragments of C2RPQ.

C2RPQ is the language of conjunctive two-way regular path queries, which can be regarded as the result of adding simple recursion to Conjunctive Queries. For graph classes $\mathcal{S}$, we consider fragments C2RPQ($\mathcal{S}$) of C2RPQ as the queries whose underlying graph structure is in $\mathcal{S}$.

We completely classify the finitely controllable and non-finitely controllable fragments under: inclusion dependencies, (frontier-)guarded rules, frontier-one rules (either with or without constants), and more generally under guarded-negation first-order constraints.

For the finitely controllable fragments, we show a reduction to the satisfiability problem for guarded-negation first-order logic, yielding a 2EXPTIME algorithm (in combined complexity) for the corresponding (F)OMQA problem.

1 Introduction

In ontology-based reasoning, a knowledge base K represents a partial view of the world which, although unknown to us, it is assumed to satisfy a given set of rules and constraints (the ontology). The ontology mediated query answering (OMQA) problem consists in deciding whether, for a given tuple and query, the tuple belongs to the answer of the query in every plausible world (finite or infinite) which extends K and satisfies the constraints. OMQA problem was formally stated in (Dolby et al. 2008; Heymans et al. 2008; Poggi et al. 2008) and largely studied in the context of Description Logics (Ortiz and Simkus 2012; Kontchakov, Rodriguez-Muro, and Zakharyaschev 2013; Bienvenu and Ortiz 2015) and existential rules (Calì, Gottlob, and Lukasiewicz 2009; Mugnier and Thomazo 2014).

The *finite* ontology mediated query answering (FOMQA) is the variant of ontology mediated query answering (OMQA) where the represented world is assumed to be finite, and thus only *finite* models of the ontology are considered. This is sometimes called finite ontology-based data access (finite OBDA) or finite open-world query answering (finite OWQA) in database theory jargon, as a form of query answering under integrity constraints or, more generally, background knowledge. The finiteness assumption allows, often but not always, to infer more properties.

Two parameters are of relevance in (F)OMQA: the ontology language $\mathcal{O}$ to specify the ontology Γ, and the query language $\mathcal{Q}$ to describe the property q of the answers. When OMQA and FOMQA problems for $\mathcal{O}$ and $\mathcal{Q}$ coincide, we say that OMQA of $\mathcal{Q}$ under $\mathcal{O}$ is *finitely controllable*. This means that if $q \in \mathcal{Q}$ holds in every *finite* extension of K satisfying $\Gamma \in \mathcal{O}$ then q holds also in every *infinite* extension satisfying Γ. We study finite-controllability for:

- *Guarded Negation fragment of First Order logic* (GNFO) in the role of $\mathcal{O}$, as studied by Bárány, ten Cate, and Segoufin (2015), which encompasses (frontier-) guarded existential rules, Guarded FO (Andréka, Németi, and van Benthem 1998), and some expressive DL language families such as $\mathcal{ALC}$, possibly with inverse roles ($\mathcal{I}$) and role inclusions ($\mathcal{H}$). OMQA of UCQ under GNFO is finitely controllable and decidable. This result is a consequence of the finite model property and decidability of the satisfiability problem, respectively, for GNFO (Bárány, ten Cate, and Segoufin 2015) via the fact that: for a finite model $\mathcal{G}$, a UCQ q, a tuple $\bar{a}$, and a GNFO sentence γ, the property stating that the model extends $\mathcal{G}$, satisfies γ, and $\bar{a}$ is not an answer to q, is effectively expressible in GNFO.
- *Conjunctive regular path queries* (CRPQ) possibly with inverse navigation roles (C2RPQ) and unions thereof (UC2RPQ) in the role of $\mathcal{Q}$. These are basic extensions of CQs and UCQs (the most studied query languages in OMQA) with a simple form of recursion, and an integral part of the W3C standard for querying RDF data (SPARQL 1.1) and often popular for querying ontologies, as revealed by recent studies (Malyshev et al. 2018; Bonifati, Martens, and Timm 2019).

Contributions. For a class of graphs $\mathcal{C}$ we consider the fragment UC2RPQ($\mathcal{C}$) to be the set of all UC2RPQ whose underlying graph structure is in $\mathcal{C}$. This is a standard approach to define syntactic subclasses of queries, as done extensively for conjunctive queries (*e.g.*, α-acyclicity, treewidth-k). (Actually, we consider graphs enriched with information on whether a node represents a free variable, and whether an edge represents an infinite language.) We show that there is a computable class $\mathcal{S}_1$ of graphs such that OMQA of UC2RPQ($\mathcal{C}$) under GNFO is finitely controllable if, and only if, $\mathcal{C} \subseteq \mathcal{S}_1$. Moreover, this characterization holds also for ontology languages weaker than GNFO. These languages contain certain "*existential rules*" (*a.k.a.* Datalog$^{\pm}$ rules, or tuple-generating dependencies), which are first-order sentences of the form $\forall \bar{x}\bar{y}(\varphi(\bar{x}\bar{y}) \Rightarrow \exists \bar{z}\psi(\bar{x}\bar{z}))$, where φ, ψ are conjunctions of atoms. Let ID be the class of inclusion dependencies (*i.e.*, φ and ψ are atoms), and let F1 stand for frontier-one rules (*i.e.*, $\bar{x}$ consists of one single variable).

Theorem 1. *For every class $\mathcal{C}$ of graphs, query language* CRPQ($\mathcal{C}$) $\subseteq \mathcal{Q} \subseteq$ UC2RPQ($\mathcal{C}$), $\mathcal{O}_0 \in \{\text{ID}, \text{F1}\}$*, and set of constraints* $\mathcal{O}_0 \subseteq \mathcal{O} \subseteq$ GNFO *we have that* OMQA *of* $\mathcal{Q}$ *under* $\mathcal{O}$ *is finitely controllable if, and only if,* $\mathcal{C} \subseteq \mathcal{S}_1$.

In the statement above, we write $\mathcal{L}_1 \subseteq \mathcal{L}_2$ to denote that properties/queries expressible in $\mathcal{L}_1$ are also expressible in $\mathcal{L}_2$. As by-product of our proofs, we obtain the decidability for OMQA of UC2RPQ($\mathcal{S}_1$) under GNFO constraints.

Corollary 1. *The* (F)OMQA *problem for* UC2RPQ($\mathcal{S}_1$) *under* GNFO *constraints is decidable, 2*EXPTIME*-complete in combined complexity.*

We also identify a larger class $\mathcal{S}_2 \supsetneq \mathcal{S}_1$ which characterizes the finitely controllable cases for frontier-one existential rules which have no constants (actually, for a slight generalization thereof), which we denote here by TF1.

Theorem 2. *For every class $\mathcal{C}$ of graphs and query language* CRPQ($\mathcal{C}$) $\subseteq \mathcal{Q} \subseteq$ UC2RPQ($\mathcal{C}$)*, we have that* OMQA *of* $\mathcal{Q}$ *under* TF1 *is finitely controllable if, and only if,* $\mathcal{C} \subseteq \mathcal{S}_2$.

Corollary 2. *The* (F)OMQA *problem for* UC2RPQ($\mathcal{S}_2$) *under* TF1 *is decidable in 2*EXPTIME *in combined complexity.*

Organization. We define the ontologies and query languages with which we work in the next Section 2. In Section 3 we formally state our main result (a slightly more general result than Theorems 1 and 2 above). One direction (the fact that some instances are finitely controllable) is shown in Section 4 and the other direction (that some other instances are non-finitely controllable) in Section 5. We discuss related work in Section 6 and we conclude in Section 7.

2 Preliminaries

Models. Let Var be a countably infinite set of variables. We consider a **vocabulary** composed of a finite set of predicates Pred and an infinite set of constants Const (*a.k.a.* individual names). The subset of *unary* predicates are usually called 'concepts' and the subset of *binary* predicates are called 'roles', which we denote here by $\mathsf{N_R}$. By $\mathsf{N_R^{\pm}}$ we denote the set of all roles $\mathsf{N_R}$ union the set of all **inverse roles** r^- for $r \in \mathsf{N_R}$. A **term** is either a variable or a constant. An **atom** α is of the form $P(\bar{t})$ where P is a predicate of arity k and $\bar{t}$ is a k-tuple of **terms**. We denote by *terms*(α) the set of terms in α and extend the notation to a conjunction of atoms. For $r \in \mathsf{N_R}$, we usually write $t \xrightarrow{r} t'$ instead of $r(t, t')$. We will often write $x, y, z, \dots$ [resp. $\bar{x}, \bar{y}, \bar{z}, \dots$] to denote variables [resp. tuples of variables]; $a, b, c, \dots$ [resp. $\bar{a}, \bar{b}, \bar{c}, \dots$] to denote constants [resp. tuples of constants]; and $t, t', \dots$ [resp. $\bar{t}, \bar{t}', \dots$] to denote terms [resp. tuples of terms]. A **ground atom** contains only constants. A **model** is a (possibly infinite) set of ground atoms. The **active domain** of a model $\mathcal{G}$, noted *adom*($\mathcal{G}$), is the set of all constants it contains in its ground atoms. For a model $\mathcal{G}$, by $\mathcal{G}^{\pm}$ we denote the extension of $\mathcal{G}$ where we add a ground atom $r^-(b, a)$ for every $r(a, b) \in \mathcal{G}$, $r \in \mathsf{N_R}$. A **homomorphism** from a model $\mathcal{A}$ to a model $\mathcal{B}$ is a function $h : \mathit{adom}(\mathcal{A}) \to \mathit{adom}(\mathcal{B})$ such that if $R(a_1, \dots, a_n) \in \mathcal{A}$ then $R(h(a_1), \dots, h(a_n)) \in \mathcal{B}$. We say that h **preserves** a set $C \subseteq \mathsf{Const}$, if $h(c) = c$ for every $c \in C$. A homomorphism from a conjunction of atoms φ to a model $\mathcal{B}$ is a function $h : \mathit{terms}(\varphi) \to \mathit{adom}(\mathcal{B})$ that preserves all constants of φ, such that if $R(t_1, \dots, t_n)$ is an atom of φ, then $R(h(t_1), \dots, h(t_n)) \in \mathcal{B}$.

Ontology-mediated querying. In general, we consider a **constraint** to be a property of a model, and a **query** to be a mapping from models to sets of vectors over the active domain. Given a query language $\mathcal{Q}$ and a constraint language $\mathcal{O}$, the **ontology-mediated query answering** (OMQA) problem of $\mathcal{Q}$ under $\mathcal{O}$ is the problem of, given a finite model $\mathcal{G}$, a finite set of constraints $\Gamma \subseteq_{\text{fin}} \mathcal{O}$, a k-ary query $q(\bar{x}) \in \mathcal{Q}$, and a k-ary tuple of constants $\bar{a}$ from *adom*($\mathcal{G}$), whether for every model $\mathcal{G}'$ that extends $\mathcal{G}$ and satisfies all the properties Γ we have that $\bar{a}$ is in the set of answers $q(\mathcal{G}')$. We also study a more general version of this problem, which will be handy for our reductions. The **generalized OMQA** (gOMQA) of $\mathcal{Q}$ under $\mathcal{O}$ is the problem of, given $\mathcal{G}$, $\Gamma \subseteq_{\text{fin}} \mathcal{O}$, and $q(\bar{x}) \in \mathcal{Q}$ as before, and given a *finite set* of k-ary tuples of constants X, whether for every $\mathcal{G}' \supseteq \mathcal{G}$ satisfying Γ there exists $\bar{a} \in X$ such that $\bar{a} \in q(\mathcal{G}')$. Note that gOMQA restricted to singleton sets is equivalent to OMQA, hence the name of 'generalized' OMQA. The **finite ontology-mediated query answering** (FOMQA) problem and its generalized version (gFOMQA) are defined analogously, except that now $\mathcal{G}'$ spans over all *finite* extensions of $\mathcal{G}$. In inconsistent database querying, FOMQA is equivalent to *Consistent Query Answering* under $\supseteq$-repairs (Arenas, Bertossi, and Chomicki 1999). We write OMQA$_{\mathcal{Q},\mathcal{O}}$ [resp. FOMQA$_{\mathcal{Q},\mathcal{O}}$, gOMQA$_{\mathcal{Q},\mathcal{O}}$, gFOMQA$_{\mathcal{Q},\mathcal{O}}$] to denote the set of all 4-tuple inputs such that the OMQA [resp. FOMQA, gOMQA, gFOMQA] problem of $\mathcal{Q}$ under $\mathcal{O}$ yields a positive answer. Observe that OMQA$_{\mathcal{Q},\mathcal{O}} \subseteq$ FOMQA$_{\mathcal{Q},\mathcal{O}}$, but the converse is not necessarily true. If OMQA$_{\mathcal{Q},\mathcal{O}} =$ FOMQA$_{\mathcal{Q},\mathcal{O}}$ [resp. gOMQA$_{\mathcal{Q},\mathcal{O}} =$ gFOMQA$_{\mathcal{Q},\mathcal{O}}$], we say that OMQA [resp. gOMQA] of $\mathcal{Q}$ under $\mathcal{O}$ has the **finite-controllability** property.

Regular path queries. We consider **regular languages** over any subset of $\mathsf{N_R^{\pm}}$ to be defined by regular expressions, defined as usual —we use standard notation ε, $\cdot \mid \cdot$, $\cdot^*$ and $\cdot^+$. Henceforth, we simply write "regular language" to denote any regular language over a finite subset of $\mathsf{N_R^{\pm}}$, and we intend it to be represented as a regular expression. A **two-way regular path query** (2RPQ) is a binary query of the form $x \xrightarrow{L} x'$, where L is a regular language, and x, x' are variables. A **conjunctive two-way regular path query**

(C2RPQ) is of the form $\gamma(\bar{x}) = \exists \bar{y}\ \varphi$, where φ is a conjunction of 2RPQ, and every variable of $\bar{x}$ occurs in φ but not in $\bar{y}$. The **arity** of $\gamma(\bar{x})$ is the dimension of $\bar{x}$, $\bar{x}$ is the tuple of **free variables**, and $\bar{y}$ is the set of **bound variables**. A **union of C2RPQ** (UC2RPQ) $\gamma(\bar{x})$ is a finite disjunction $\gamma_1(\bar{x}) \vee \cdots \vee \gamma_n(\bar{x})$ of C2RPQ, all sharing the same free variables $\bar{x}$. A UC2RPQ is **Boolean** if it has no free variables. Given a 2RPQ $\gamma = x \xrightarrow{L} x'$, a model $\mathcal{G}$, and an assignment ν from the free variables of γ to $adom(\mathcal{G})$, we write $\mathcal{G}, \nu \models \gamma$ if there is a directed path from $\nu(x)$ to $\nu(x')$ in $\mathcal{G}^{\pm}$, labeled with a word from L. Observe that for every regular language L there is another regular language L^{-} such that $\mathcal{G}, \nu \models x \xrightarrow{L} x'$ if, and only if, $\mathcal{G}, \nu \models x' \xrightarrow{L^{-}} x$. We write $()$ for the 0-ary tuple and $\emptyset$ for the empty valuation. We extend this definition to UC2RPQ in the standard way: $\mathcal{G}, \nu \models \exists x\ \gamma$ if there is some $a \in adom(\mathcal{G})$ such that $\mathcal{G}, \nu \dot{\cup} \{x \mapsto a\} \models \gamma$; $\mathcal{G}, \nu \models \gamma_1 \vee \gamma_2$ if either $\mathcal{G}, \nu \models \gamma_1$ or $\mathcal{G}, \nu \models \gamma_2$; and $\mathcal{G}, \nu \models \gamma_1 \wedge \gamma_2$ if both $\mathcal{G}, \nu \models \gamma_1$ and $\mathcal{G}, \nu \models \gamma_2$. Hence, for instance if $\mathcal{G}, \nu \models \gamma(\bar{x})$ then $\mathcal{G}, \emptyset \models \exists \bar{x}\ \gamma$. For a UC2RPQ $\gamma(\bar{x})$ of arity k we write $\gamma(\mathcal{G})$ to denote the set of all tuples $\bar{a} \in (adom(\mathcal{G}))^k$ such that $\mathcal{G}, \{\bar{x} \mapsto \bar{a}\} \models \gamma$. Observe that if γ is Boolean, then $\gamma(\mathcal{G})$ is either $\{()\}$ (*i.e.*, *true*) or $\{\}$ (*i.e.*, *false*).

Constraint languages. We define here the constraint languages on which we focus, starting with the most expressive one we consider: guarded-negation. **Guarded-negation first-order logic** (GNFO) is a fragment of first-order logic with equality, given by the following grammar, where $P \in \mathsf{Pred}$ and $\alpha \in \mathsf{Pred} \dot{\cup} \{=\}$:

$$\varphi ::= P(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\ \varphi \mid \alpha(\bar{x}\bar{y}) \wedge \neg\varphi(\bar{y})$$

Here $P(\bar{x})$ and $\alpha(\bar{x}\bar{y})$ are atoms, possibly containing constants, whose free variables are $\bar{x}$ and $\bar{x}\bar{y}$, respectively. Observe that in GNFO, formulas with 0 or 1 free variables are closed under negation —the latter through the equivalence $\neg\varphi(x) \equiv (x = x \wedge \neg\varphi(x))$. GNFO enjoys many desirable properties (Segoufin 2017). In particular, it has the finite-model property (if $\varphi \in$ GNFO is satisfiable, it is also satisfiable in a finite model) and decidable, 2EXPTIME-c, satisfiability problem (Bárány, ten Cate, and Segoufin 2015). We will use these facts to prove results on finite-controllability.

An **existential rule** (*a.k.a.* Datalog$^{\pm}$ rules, or tuple-generating dependencies) is a first-order sentence of the form $\forall \bar{x}\bar{y}\ (\varphi(\bar{x}\bar{y}) \Rightarrow \exists \bar{z}\ \psi(\bar{x}\bar{z}))$, where φ and ψ are conjunctions of atoms. We call φ and ψ the **body** and **head** of the rule, respectively, and $\bar{x}$ the **frontier variables**. Such an existential rule is **frontier-guarded** if there is an atom in the body containing all frontier variables; we denote by FG the set of all frontier-guarded existential rules. An existential rule is **frontier-one** if it has at most one frontier variable; we denote by F1 the set of all frontier-one existential rules. A frontier-one rule is **term-frontier-one** if its body and atom have at most one term in common (*i.e.*, it could be a variable or a constant). In particular, a rule $(\forall x) R(c, x) \Rightarrow S(c, x)$ where c is a constant, is frontier-one but not term-frontier-one. We denote by TF1 the set of all term-frontier-one rules. An **inclusion dependency** is an existential rule with no constants whose head and body consist of only one atom; we denote by ID the set of all inclusion dependencies. It is easy to see that these classes ID, F1, FG are definable in GNFO, through polynomial-time translations.

The Chase. We will resort to the *Chase* for existential rules as defined, *e.g.*, in (Calì, Gottlob, and Lukasiewicz 2012). Given a finite set Γ of existential rules, $Chase(\mathcal{G}, \Gamma)$ is a possibly infinite model defined as the infinitary union $\bigcup_{i \geq 0} \mathcal{G}_i$ such that $\mathcal{G}_0 = \mathcal{G}$, and for every $i > 0$, $\mathcal{G}_i$ is, roughly speaking, a model extending $\mathcal{G}_{i-1}$ with finitely many new ground atoms that witness the head of a rule, whenever its body is satisfied. Concretely, let S be the set of all pairs (ψ, h) such that there exists a rule $\forall \bar{x}\bar{y}\ (\varphi(\bar{x}\bar{y}) \Rightarrow \exists \bar{z}\ \psi(\bar{y}\bar{z}))$ in Γ and a homomorphism h from $\varphi(\bar{x}\bar{y})$ to $\mathcal{G}_{i-1}$ such that $\mathcal{G}_{i-1} \not\models \exists \bar{z}\ \psi(h(\bar{y})\bar{z})$. For every such pair of S, let $g_{\psi,h} : \bar{z} \to \mathsf{Const} \setminus adom(\mathcal{G}_{i-1})$ be any injective mapping from the variables $\bar{z}$ to constants so that the image of any two such functions have empty intersection. We define $\mathcal{G}_i$ as $\mathcal{G}_{i-1} \cup \bigcup_{(\psi,h) \in S} F_{\psi,h}$, where $F_{\psi,h}$ is the set of all facts obtained from the atoms of ψ by replacing variables with constants according to h and $g_{\psi,h}$. Observe that $\mathcal{G}_{i-1} = \mathcal{G}_i$ iff $\mathcal{G}_{i-1} \models \Gamma$. Further, every $\mathcal{G}_i$ is unique modulo renaming of constants from $\mathsf{Const} \setminus adom(\mathcal{G})$, and hence so is $Chase(\mathcal{G}, \Gamma)$. We call $\mathcal{G}_i$ the ***i*-th step of the chase**, denoted by $Chase^i(\mathcal{G}, \Gamma)$. This is usually called the 'standard' or 'restrictive' Chase. There are three fundamental properties of $Chase(\mathcal{G}, \Gamma)$ that we will use in our proofs:

(a) $Chase(\mathcal{G}, \Gamma)$ satisfies Γ.

(b) It is a *universal model*: if $\mathcal{G}' \supseteq \mathcal{G}$ and $\mathcal{G}'$ satisfies Γ then there is a homomorphism from $Chase(\mathcal{G}, \Gamma)$ to $\mathcal{G}'$ preserving $adom(\mathcal{G})$ (Fagin et al. 2005).

(c) It can be shown that if $\Gamma \subseteq$ TF1, then for every $a \in adom(Chase^i(\mathcal{G}, \Gamma))$, $i \geq 0$, there is $j \geq i$ such that all facts containing a in $Chase(\mathcal{G}, \Gamma)$ are already present in $Chase^j(\mathcal{G}, \Gamma)$; in particular, $Chase(\mathcal{G}, \Gamma)$ is *finitely branching*.

OMQA of UC2RPQ under GNFO. The OMQA problem for CRPQ is decidable under GNFO even when extended with guarded least fixpoint (GNFP). Remember that GNFO can express guarded and frontier guarded existential rules.

Proposition 1. *The* OMQA *of* UC2RPQ *under* GNFO *is decidable.*

Proof. This follows from a result of Benedikt, Bourhis, and Vanden Boom (2016), stating that there is an extension GNFP-UP of GNFP (GNFO extended with guarded least fixpoint), which captures UC2RPQ and has a decidable satisfiability problem. □

However, the respective FOMQA problem remains open —indeed, while the finite satisfiability problem for GNFP is decidable (Bárány and Bojańczyk 2012), it is not clear whether it is also decidable for GNFO-UP.

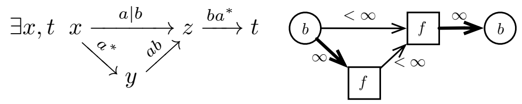


Figure 1: A query and its skeleton.

For the case of constraints defined by guarded existential rules, Baget et al. (2017) show that OMQA is in 2ExpTime.

Proposition 2. (Baget et al. 2017) *The* OMQA *of* UC2RPQ *under guarded existential rules is* 2ExpTime*-complete in combined complexity,* PTime*-complete in data complexity.*

3 Main Results

As anticipated in Section 1, our results characterize the classes of finitely controllable UC2RPQ based on the shape of its underlying graph. However, we want our results to be more fine-grained than that, and we distinguish, in the underlying graph, which nodes come from a free variable and which edges come from finite languages. We call this the *skeleton* of a C2RPQ. We define a class of skeletons $\mathcal{S}$ such that (i) every query with a skeleton from $\mathcal{S}$ is finite controllable and (ii) for every skeleton $s \notin \mathcal{S}$ there is a query which is not finitely controllable and has skeleton s. We show that there is a class $\mathcal{S}$ with such a property for every language of constraints containing inclusion dependencies and contained in GNFO, which we denote by $\mathcal{S}_1$; and that there is a strictly larger class satisfying the property for frontier-one existential rules, which we denote by $\mathcal{S}_2$.

A **multigraph** is a tuple $M = (V, E, \eta)$ where V is a finite set of vertices, E is a finite set of edges, and $\eta : E \to V \times V$ associates every edge with its source and target vertices. The **underlying undirected graph** G_M of M is the simple undirected graph having V as set of vertices and $\{\{v, v'\} : e \in E, \eta(e) = (v, v')\}$ as set of edges. An **undirected simple path** [resp. **undirected cycle**] of M is a (possibly empty) sequence of edges which induces a simple path [resp. a cycle] in G_M.

A **skeleton** is a triple (M, ν, μ) where $M = (V, E, \eta)$ is a multigraph, $\nu : V \to \{b, f\}$, and $\mu : E \to \{\infty, <\infty\}$. In this context, we say that a vertex $v \in V$ is either *free* or *bound* depending on wether $\nu(v) = f$ or $\nu(v) = b$. The **distance** between two vertices v, v' is **finite** if there is an undirected path $e_1, \dots, e_n$ so that $\mu(e_i) = <\infty$ for all i. An undirected cycle $e_1, \dots, e_n$ of M is **infinite** if $\mu(e_i) = \infty$ for some i, and it is **bound** if for every i such that $\eta(e_i) = (v_1, v_2)$ we have $\nu(v_1) = \nu(v_2) = b$.

Given a C2RPQ q and a skeleton $sk = (M, \nu, \mu)$, we say that sk **is the skeleton of** q if for $M = (V, E, \eta)$ we have V is the set of variables of q, E is the set of atoms of q, $\eta(e) = (v_1, v_2)$ iff the atom e is of the form $v_1 \xrightarrow{L} v_2$ for some L, $\nu(v) = b$ iff v is bound in q, and $\mu(e) = \infty$ iff the atom e is of the form $v_1 \xrightarrow{L} v_2$ for some *infinite* language L.

See Figure 1 for an example of a query and its skeleton. We often refer to the elements of V and E as vertices and atoms of sk. The skeleton of a UC2RPQ is the set of skeletons of the C2RPQ queries therein. Given a set of skeletons $\mathcal{S}$, we define the class **UC2RPQ**($\mathcal{S}$) of UC2RPQ as the set of all UC2RPQ queries whose skeletons are in $\mathcal{S}$.

We define $\mathcal{S}_1$ to be the set of all skeletons such that every undirected infinite cycle contains a free node. We define $\mathcal{S}_2$ to be the set of all skeletons such that every infinite bound cycle contains a node at finite distance from a free node.

Our results can be summarized as follows.

Main Theorem. *For every class $\mathcal{S}$ of skeletons, for every $\mathcal{O}_0 \in \{\text{ID}, \text{F1}\}$, for every query language* CRPQ($\mathcal{S}$) $\subseteq \mathcal{Q} \subseteq$ UC2RPQ($\mathcal{S}$)*, we have that*

1. *If $\mathcal{O}_0 \subseteq \mathcal{O} \subseteq$* GNFO*, then* OMQA *[resp.* gOMQA*] of $\mathcal{Q}$ under $\mathcal{O}$ is finitely controllable if, and only if, $\mathcal{S} \subseteq \mathcal{S}_1$;*
2. OMQA *[resp.* gOMQA*] of $\mathcal{Q}$ under* TF1 *is finitely controllable if, and only if, $\mathcal{S} \subseteq \mathcal{S}_2$.*

As a corollary we obtain Theorems 1 and 2 as stated in Section 1, even for *generalized* OMQA. Observe that the Main Theorem is stated as a general *equivalence*, and this is the reason for $\mathcal{O}$ having both a lower bound $\{\text{ID}, \text{F1}\}$ and an upper bound GNFO in the hypothesis of item 1. Alternatively, item 1 above could be restated as: (1.a) If $\mathcal{O} \subseteq$ GNFO and $\mathcal{S} \subseteq \mathcal{S}_1$, then OMQA [resp. gOMQA] of $\mathcal{Q}$ under $\mathcal{O}$ is finitely-controllable. (1.b) If $\mathcal{O}_0 \subseteq \mathcal{O}$ and $\mathcal{S} \not\subseteq \mathcal{S}_1$, then OMQA [resp. gOMQA] of $\mathcal{Q}$ is not finitely-controllable.

Proof strategy. The right-to-left implications of Main Theorem (which are in Section 4), will follow from reductions to the finite-satisfiability problem for GNFO. Since GNFO has the finite-model property, it follows that it is finitely controllable. These reductions, however, are not completely straightforward, they are shown through the following sequence of reductions, for some class of skeletons $\mathcal{S}_0$ such that $\mathcal{S}_0 \subsetneq \mathcal{S}_1 \subsetneq \mathcal{S}_2$:

(i) gOMQA of UC2RPQ($\mathcal{S}_2$) under TF1 reduces to gOMQA of UC2RPQ($\mathcal{S}_1$) under TF1 (Proposition 4);

(ii) gOMQA of UC2RPQ($\mathcal{S}_1$) under GNFO reduces to gOMQA of UC2RPQ($\mathcal{S}_0$) under GNFO (Prop. 3);

(iii) gOMQA of UC2RPQ($\mathcal{S}_0$) under GNFO reduces to the finite satisfiability problem for GNFO (Lemma 1), and hence it is finitely controllable.

In particular, for the first reduction (i) we need to work with the most general problem of *generalized* OMQA. In fact, it is not clear to us how to reduce UC2RPQ($\mathcal{S}_2$) to UC2RPQ($\mathcal{S}_1$) without going to this more general setting. This is the reason why we work with the generalized version of the problem.

The complexity statement of Corollary 1 of the introduction is a direct consequence of the fact that reductions (ii) and (iii) are polynomial-time, combined with 2ExpTime-completeness of the satisfiability problem for GNFO (Bárány, ten Cate, and Segoufin 2015). On the other hand, Corollary 2 follows from Proposition 2, the fact that UC2RPQ($\mathcal{S}_2$) under TF1 is finitely controllable (item 2 of Main Theorem), and the fact that guarded rules contain TF1.

The left-to-right implications of Main Theorem (which are in Section 5) follow, as expected, from counterexamples. That is, for $(\mathcal{S}, \mathcal{O}) \in \{(\mathcal{S}_1, \text{F1}), (\mathcal{S}_1, \text{ID}), (\mathcal{S}_2, \text{TF1})\}$, and a skeleton $sk \notin \mathcal{S}$, we exhibit a model $\mathcal{G}$, a tuple of elements $\bar{t}$, a query $q \in$ CRPQ with skeleton sk, and a set of constraints Γ in $\mathcal{O}$ for which $\bar{t}$ is in the answer set of q on $\mathcal{G}'$, for every finite extension $\mathcal{G}' \supseteq \mathcal{G}$ verifying Γ, but $\bar{t}$ is not in the answer of q on some infinite extension of $\mathcal{G}$ verifying Γ (in particular, on *Chase*$(\mathcal{G}, \Gamma)$).

4 Reductions to GNFO

In this section, we show the right-to-left implications of both items of the main theorem. Observe that for this direction it suffices to prove finite-controllability of gOMQA for UC2RPQ$(\mathcal{S}_1)$ [resp. UC2RPQ$(\mathcal{S}_2)$] under GNFO [resp. TF1]. For the first implication, we proceed in steps. First, we show the implication for a restricted class of skeletons. Let $\mathcal{S}_0 \subseteq \mathcal{S}_1$ be the set of all skeletons that do not contain any undirected infinite cycle.

Lemma 1. gOMQA *of* UC2RPQ$(\mathcal{S}_0)$ *under* GNFO *is finitely controllable.*

Lemma 1 is our main technical lemma and its proof will be deferred to the end of the section. Using this lemma, the right-to-left implication of Main Theorem's item 1 follows:

Proposition 3. gOMQA *of* UC2RPQ$(\mathcal{S}_1)$ *under* GNFO *is finitely controllable.*

Proof. The idea is very simple: to break all cycles in the query by introducing new free variables and apply Lemma 1.

Let $q(\bar{x})$ in UC2RPQ$(\mathcal{S}_1)$ be a query with k free variables $\bar{x} = (x_1, \dots, x_k)$. Suppose that variable x_i occurs exactly m_i many times in $q(\bar{x})$. For $k' = \sum_i m_i$, and the k'-tuple $\bar{y} = (y_1^1, \dots, y_1^{m_1}, \dots, y_k^1, \dots, x_k^{m_k})$, we define $q'(\bar{y})$ as the result of replacing the j-th occurrence of x_i in $q(\bar{x})$ by y_i^j (which we assume is not occurring in q), for $i = 1 \dots k$. Any cycle in $q(\bar{x})$ has a free variable, and this cycle is absent in $q'(\bar{y})$, since in $q'(\bar{y})$ each free variable is used exactly once. Hence $q'(\bar{y}) \in$ UC2RPQ$(\mathcal{S}_0)$.

Observe that for any model $\mathcal{G}$ and valuation $\nu : \bar{x} \to$ *adom*$(\mathcal{G})$, we have $\mathcal{G}, \nu \models q$ iff $\mathcal{G}, \nu' \models q'$, where ν' is the valuation defined by $\nu'(y_i^j) = \nu(x_i)$. Therefore gOMQA for to $q(\bar{x})$ and X is equivalent to gOMQA for $q'(\bar{y})$ and X', for

$$X' = \{(\underbrace{c_1, \dots, c_1}_{m_1 \text{ times}}, \dots, \underbrace{c_k, \dots, c_k}_{m_k \text{ times}}) \mid (c_1, \dots, c_k) \in X\}.$$

Hence, by Lemma 1, the statement follows. □

The right-to-left implication of item 2 of the main theorem is shown by reduction to item 1, exploiting two facts: (i) that we are working with the *generalized* version gOMQA of OMQA; and (ii) that the Chase under TF1 rules enjoys property (c), as defined in Section 2.

Proposition 4. gOMQA *of* UC2RPQ$(\mathcal{S}_2)$ *under* TF1 *is finitely controllable.*

Proof. We reduce the gOMQA problem for UC2RPQ $(\mathcal{S}_2)$ to the gOMQA problem for UC2RPQ$(\mathcal{S}_1)$ (both under TF1 rules). Let $q(\bar{x}) \in$ UC2RPQ$(\mathcal{S}_1)$ have k free variables $\bar{x} = (x_1, \dots, x_k)$, let $\mathcal{G}$ be a finite model, let Γ be a finite set of TF1 rules, and let X be a finite set of k-ary constants.

Let Z be the smallest set of variables of $q(\bar{x})$ containing all free variables and satisfying that for every atom $x \xrightarrow{L} y$ of q where L is finite, we have $x \in Z$ if and only if $y \in Z$. Suppose Z contains r bound variables $y_1, \dots, y_r$ of $q(\bar{x})$. Consider now the query $q'(\bar{x}\bar{y})$, where $\bar{y} = (y_1, \dots, y_r)$, which results from making all variables of $Z \setminus \bar{x}$ free. Observe that $q'(\bar{x}\bar{y}) \in$ UC2RPQ$(\mathcal{S}_1)$.

Let m be a number which is larger than the length of any word in any finite language in $q(\bar{x})$. Let U be the set of all constants of *Chase*$(\mathcal{G}, \Gamma)$ at distance $\leq m$ from a constant in $\mathcal{G}$. By the property (c) discussed in section 2, U is finite. Let $\mathcal{G}^+ = Chase^i(\mathcal{G}, \Gamma)$, for i large so such that $Chase^i(\mathcal{G}, \Gamma)$ contains U. Note that $\mathcal{G}^+$ contains $\mathcal{G}$. Consider now the set of tuples of arity $k + r$ of atoms in $\mathcal{G}^+$, defined as $X' = X \times U^r$. We next show that $\mathcal{G}, \Gamma, q(\bar{x}), X$ is a positive instance of gOMQA if, and only if, so is $\mathcal{G}^+, \Gamma, q'(\bar{x}\bar{y}), X'$. We then conclude the proof, by Proposition 3, by the fact that $q'(\bar{x}\bar{y}) \in$ C2RPQ$(\mathcal{S}_1)$, and that TF1 $\subseteq$ GNFO.

For the left-to-right implication, let $\mathcal{G}' \supseteq \mathcal{G}^+$ be a model that satisfies Γ. Since $\mathcal{G}' \supseteq \mathcal{G}$, we know, by hypothesis, that for some k-tuple $\bar{a} \in X$, we have $\mathcal{G}', \{\bar{x} \mapsto \bar{a}\} \models q(\bar{x})$. This implies that there is an r-tuple of constants $\bar{b}$ such that $\mathcal{G}', \{\bar{x}\bar{y} \mapsto \bar{a}\bar{b}\} \models q'(\bar{x}\bar{y})$. By the choice of $\bar{y}$, any constant c of $\bar{b}$ is at distance at most m from a constant in $\mathcal{G}$, and then $c \in U$. Therefore, $(\bar{a}\bar{b}) \in X'$ and we are done.

For the right-to-left implication, we first observe that *Chase*$(\mathcal{G}, \Gamma)$ satisfies Γ by property (a) of the Chase, and *Chase*$(\mathcal{G}, \Gamma) \supseteq \mathcal{G}^+$ by construction. By hypothesis we know that for some k-tuple $\bar{a}$ and some r-tuple $\bar{b} = (b_1, \dots, b_r)$ such that $\bar{a}\bar{b} \in X'$ we have *Chase*$(\mathcal{G}, \Gamma), \{\bar{x}\bar{y} \mapsto \bar{a}\bar{b}\} \models q'(\bar{x}\bar{y})$. Let $\mathcal{G}' \supseteq \mathcal{G}$ be a model that satisfies Γ. By property (b) of the Chase, we have that that there is a homomorphism $h :$ *Chase*$(\mathcal{G}, \Gamma) \to \mathcal{G}'$ such that $h(a) = a$ for every $a \in$ *adom*$(\mathcal{G})$. Since any query in UC2RPQ is preserved under homomorphisms, we have $\mathcal{G}' \models q'(\bar{a}, h(b_1), \dots, h(b_r))$, and this implies that $\mathcal{G}' \models q(\bar{a})$. □

The remaining of this section is devoted to the proof of Lemma 1, which is shown via a somewhat involved reduction to the satisfiability problem for GNFO.

4.1 Proof of Lemma 1

For simplicity we consider $q(\bar{x})$ in C2RPQ$(\mathcal{S}_0)$, but the argument can be easily adapted for $q(\bar{x})$ in UC2RPQ$(\mathcal{S}_0)$. Let $\mathcal{G}$ be a finite model, let Γ be a finite set of GNFO sentences, let $q(\bar{x})$ in C2RPQ$(\mathcal{S}_0)$ be a query with k free variables $\bar{x} = (x_1, \dots, x_k)$ and let $X \subseteq_{\text{fin}} (adom(\mathcal{G}))^k$ be a set of k-ary tuples. We build a sentence φ in GNFO such that

> φ is satisfiable [resp. finitely satisfiable] if, and only if, there is a model [resp. finite model] $\mathcal{G}' \supseteq \mathcal{G}$ satisfying Γ such that $X \cap q(\mathcal{G}') = \emptyset$. (†)

Then we are done, since GNFO has the finite model property. In what follows, we first introduce some notions needed for the proof. Then we define the sentence φ. And finally we verify that it is in GNFO and that it satisfies (†).

Regions. Before giving the definition of φ, we introduce some constructions and notions over the skeleton sk of q.

A **region of** $sk = (M, \nu, \mu)$ is the skeleton of a connected subgraph of M. A **finite region** r **of** sk is a region in which $\mu(e) = < \infty$ for every atom e in r. Although $sk \in \mathcal{S}_0$ may contain cycles, the idea is to regard sk as a labeled directed tree, identifying all maximal finite regions by a single node, being connected to other nodes only by infinite languages. Henceforth, by finite region we mean a *maximal* finite region of sk and, for simplicity and without loss of generality, we assume that sk is connected. When we look at the connections between finite regions of $sk \in \mathcal{S}_0$, it looks a like a tree, as in Figure 2. The definition of φ exploits this structure to define, bottom-up, the execution of the automata corresponding to the regular languages in q. In the next paragraph we introduce some notation for naming different parts of this tree, to then define formally φ.

We say that two finite regions of sk are **connected** if there is an infinite edge with the source node in one region and the target in the other one. Notice that by definition of $\mathcal{S}_0$, a region does not connect to itself and two regions could be connected by at most one infinite language (as otherwise sk would contain an infinite cycle). What is more the simple graph having regions as nodes and an edge between every pair of connected regions is a tree (otherwise, the existence of a cycle in it, together with the connectivity of the regions, would imply an infinite cycle in sk). We designate a region *root* of sk as the **root**, and we use the usual tree nomenclature (parent, child, etc.). For a region r, we denote by $\mathrm{ch}(r)$ the set of all its children. Observe that if r is the parent of r', then there is exactly one variable y' in r', exactly one variable y in r, and exactly one infinite regular language L such that either $y \xrightarrow{L} y'$ or $y' \xrightarrow{L} y$ is in sk. We will denote variables y' and y by $\mathrm{src}_{r'}$ and $\mathrm{tgt}_{r'}$ respectively. We define $L_{r'}$ as L in case $y' \xrightarrow{L} y$, and as L^{-} in case $y \xrightarrow{L} y'$. The **parent-interface** of a region $r \neq root$ is src_r. The **children-interface** of a region r is the set $\{\mathrm{tgt}_{r'} \mid r' \in \mathrm{ch}(r)\}$. The **region contraction of** sk **with root** *root* is the labeled tree-like digraph $\mathcal{T}$ whose set of nodes $V_{\mathcal{T}}$ is the set of maximal finite regions of sk, the set of labeled edges $E_{\mathcal{T}} \subseteq V_{\mathcal{T}} \times V_{\mathcal{T}}$ is defined by $(r', r) \in E_{\mathcal{T}}$ iff r is the parent of r', and for $(r', r) \in E_{\mathcal{T}}$, and we define the labeling $\ell(r', r) = L_{r'}$. Notice also that every edge of $\mathcal{T}$ points towards the root *root*. See Figure 2 for an example.

Sometimes we refer to a region r also as a node of $\mathcal{T}$. For simplicity, we replace some atoms $x_i \xrightarrow{L_i} y_i$ with $y_i \xrightarrow{L_i^{-}} x_i$ in $q(\bar{x})$, so that the new directions agree with those in $\mathcal{T}$.

Expressing regions in first order. It is easy to see that for every atom $x \xrightarrow{L} y$ of q with L finite there is an equivalent formula $\chi_L(x, y)$ in positive existential first order logic with equality over the same signature computable in linear time.

For every region $r \neq root$ of sk we define the formula

$$\psi_r(\bar{b}_r, \bar{w}_r, \mathrm{src}_r) \stackrel{\mathrm{def}}{=} \exists \bar{z}_r \bigwedge_{\substack{t \xrightarrow{L} t' \text{ is an} \\ \text{atom in } r}} \chi_L(t, t'), \qquad (1)$$

where $\bar{b}_r$ is the tuple of bound variables of r which are in the children-interface of r, $\bar{w}_r$ is the tuple of free variables of r, src_r is the parent-interface of r, and $\bar{z}_r$ is the tuple of bound variables in r that are not in $\bar{b}_r$. See Figure 2 for an example. When $r = root$ we define ψ_r in a similar way, except that the variable src_r is omitted, and we simply write $\psi_r(\bar{b}_{root}, \bar{w}_{root})$. If there are no atoms in r we define ψ_r simply as $\top$. The free variables of ψ_r are among $\bar{b}_r, \bar{w}_r, \mathrm{src}_r$; notice that this tuple could have repetitions: in case the parent-interface of r is also a child-interface (namely, $\mathrm{src}_r \in \bar{b}_r$, or src_r is free, see Figure 2). Notice that ψ_r is basically the subformula of $q(\bar{x})$ for which r is the corresponding skeleton, except that the bound variables in $\bar{b}_r$ are free in ψ_r, *i.e.*, r is the skeleton of $\exists \bar{b}_r \psi_r$.

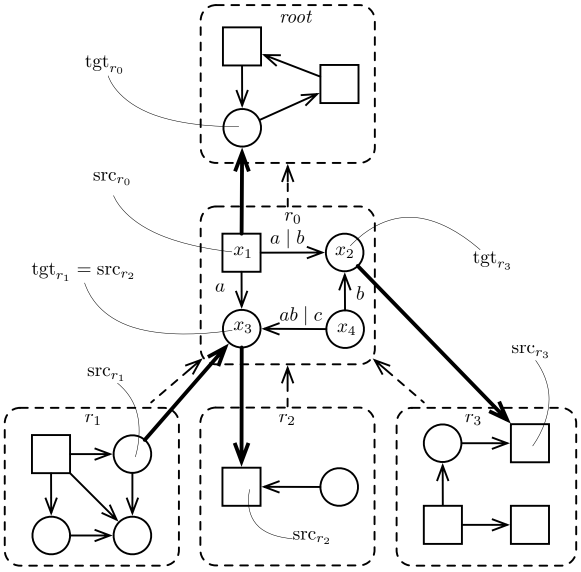


Figure 2: In solid lines, a skeleton $sk \in \mathcal{S}_1$ (infinite languages in boldface, finite languages in lightface, free variables as boxes, and bound variables as circles). In dotted lines, the contraction of sk, represented as a tree (regions and arrows between them towards the root). $\psi_{r_0} = \exists x_4((x_4 \xrightarrow{c} x_3) \vee \exists z(x_4 \xrightarrow{a} z \wedge z \xrightarrow{b} x_3)) \wedge x_1 \xrightarrow{a} x_3 \wedge x_4 \xrightarrow{b} x_2 \wedge (x_1 \xrightarrow{a} x_2 \vee x_1 \xrightarrow{b} x_2)$; according to notation in (1), its free variables are $(b_1, b_2) = (x_3, x_4)$, $w = x_1$ and $\mathrm{src}_r = x_1$.

Let $adom(\mathcal{G}) = \{v_1, \ldots, v_n\}$. We denote by $\bar{v}$ the tuple $(v_1, \ldots, v_n)$. For every variable y of $q(\bar{x})$ and k-tuple of constants $\bar{a} = (a_1, \ldots, a_k) \in (adom(\mathcal{G}))^k$, we define $\bar{a}[y]$ as the constant symbol v_j whenever $y = x_i$ is a free variable of $q(\bar{x})$ and $a_i = v_j$, and as the variable name y otherwise. For every $\bar{a} \in X$, we define $\psi_r^{\bar{a}}$ as the result of replacing every variable $y \in \bar{w}_r$ with $\bar{a}[y]$ in ψ_r.The free variables of $\psi_r^{\bar{a}}$ are among those in $\bar{b}_r, \mathrm{src}_r$. We explicitly use notation $\psi_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r)$ to stress this fact, although src_r might not be in this formula.

The formula φ. Suppose that $q(\bar{x}) = \exists \bar{y} \bigwedge_{i \in I} t_i \xrightarrow{L_i} t'_i$. For every regular language L_i, we denote the NFA that accepts it as $\mathcal{A}_i$ and we assume that it has one initial state q_0 and one final state q_f. If for region r, $L_r = L_i$ then we define $\mathcal{A}_r$ as $\mathcal{A}_i$.

We define φ over the signature extending that of $\mathcal{G}$ (both constants and predicates) with unary relations $s^{\bar{a}}_{\mathcal{A}_i}$ for each state s of $\mathcal{A}_i$ ($i \in I$), and for every $\bar{a} \in X$. Concretely, the sentence φ is defined by

$$\varphi \overset{\text{def}}{=} \varphi_1 \wedge \varphi_2 \wedge \bigwedge_{\bar{a} \in X} (\varphi_3^{\bar{a}} \wedge \varphi_4^{\bar{a}} \wedge \varphi_5^{\bar{a}}).$$

The idea is that if φ holds in a structure $\mathcal{M}$ then φ_1 expresses that $\mathcal{G} \subseteq \mathcal{M}$, and φ_2 that $\mathcal{M}$ satisfies the constraints. φ_3 expresses that any region $r \neq \mathit{root}$ that can be realized in $\mathcal{M}$ and that is a leaf in $\mathcal{T}$ is forced to start the automaton $\mathcal{A}_r$ of its parent-interface node; any region that can be realized in $\mathcal{M}$ is also forced to start, provided the automaton of all its children have reached the final state. $\varphi_4^{\bar{a}}$ expresses that the states of the automaton of any infinite language in $q(\bar{x})$ are forced to be spread over $\mathcal{M}$ according to its rules. Finally, $\varphi_5^{\bar{a}}$ expresses that if all the nodes in the children-interface of *root* are in the final state of the corresponding languages, then *root* is not realized in $\mathcal{M}$.

Observe that sentences φ_1 and φ_2 are trivial to encode in GNFO. We show how to define the remaining subsentences.

$\varphi_3^{\bar{a}}$: Initiating automata. For any $\bar{a} \in X$ and region r, let

$$\theta_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r) \overset{\text{def}}{=} \psi_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r) \wedge \bigwedge_{r' \in \mathrm{ch}(r)} qf^{\bar{a}}_{\mathcal{A}_{r'}}(\bar{a}[\mathrm{tgt}_{r'}]),$$

(when $r = \mathit{root}$, we simply write $\theta^{\bar{a}}_{\mathit{root}}(\bar{b}_{\mathit{root}})$) and for any $\bar{a} \in X$ and $r \neq \mathit{root}$, we define

$$\varphi_3^{\bar{a}} \overset{\text{def}}{=} \bigwedge_{r \neq \mathit{root}} \forall \bar{b}_r \forall \mathrm{src}_r \big(\theta_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r) \Rightarrow q0^{\bar{a}}_{\mathcal{A}_r}(\bar{a}[\mathrm{src}_r])\big).$$

The idea of $\varphi_3^{\bar{a}}$ is, for every region $r \neq \mathit{root}$, to force to trigger the run of the automaton corresponding to the parent-interface of r whenever all the nodes of the children-interface of r have reached the final state, and the restriction given by r is satisfied. This is done for *any* possible assignment of variables in $\bar{b}_r$, and of src_r, the parent interface of r, whenever it is bound in r. Observe that if r' is a child of r, we have that $\bar{a}(\mathrm{tgt}_{r'})$ is a constant in $\mathit{adom}(\mathcal{G})$ in case $\mathrm{tgt}_{r'}$ is free and is a variable in $\bar{b}_r$ otherwise. A similar situation occurs with src_r: in case src_r is free then $\mathrm{src}_r \in \mathit{adom}(\mathcal{G})$ and the quantifier $\forall \mathrm{src}_r$ is spurious; else, $\bar{a}(\mathrm{src}_r) = \mathrm{src}_r$ and is universally quantified.

$\varphi_4^{\bar{a}}$: Runs of automata. For every $\bar{a} \in X$, we define

$$\varphi_4^{\bar{a}} \overset{\text{def}}{=} \forall x, y \bigwedge_{\substack{|L_i| = \infty \text{ in } q(\bar{x}),\\ t \xrightarrow{\alpha} s \text{ in } \mathcal{A}_i}} \big(t^{\bar{a}}_{\mathcal{A}_i}(x) \wedge \chi_\alpha(x, y) \Longrightarrow s^{\bar{a}}_{\mathcal{A}_i}(y)\big).$$

The formula $\varphi_4^{\bar{a}}$ forces to spread the states of the automata of infinite languages in $q(\bar{x})$ as unary relations over nodes of the model (finite languages are already handled by $\varphi_3^{\bar{a}}$).

$\varphi_5^{\bar{a}}$: Root does not terminate. Finally, for every $\bar{a} \in X$, we define $\varphi_5^{\bar{a}} \overset{\text{def}}{=} \neg \exists \bar{b}_{\mathit{root}}\ \theta^{\bar{a}}_{\mathit{root}}(\bar{b}_{\mathit{root}})$. It negates the possibility that all the nodes in the children-interface of *root* may reach the final state and the region *root* can be realized.

Verification. It is easy to verify that φ is in GNFO. We now prove property (†). Let us first show the left-to-right implication. Let $\mathcal{H}$ be a model that satisfies φ and let $\mathcal{H}'$ be the reduct of $\mathcal{H}$ to the predicate symbols in $\mathcal{G}$. Since $\mathcal{H} \models \varphi_1 \wedge \varphi_2$, $\mathcal{H}' \supseteq \mathcal{G}$ and $\mathcal{H}'$ satisfies Γ. It suffices to show that $\bar{a} \notin q(\mathcal{H}')$ for all $\bar{a} \in X$, since $\mathcal{H}$ is finite iff $\mathcal{H}'$ is so.

Assume by contradiction that $\bar{a} \in X$ is a k-tuple of constants such that $\mathcal{H}' \models q(\bar{a})$. Then there is a valuation ν in $\mathcal{H}'$ that interprets all variables $\bar{b}_r, \mathrm{src}_r$, for all regions r in sk such that $\mathcal{H}, \nu \models \psi_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r)$ for any r. Furthermore, if r is the parent of r' then there is a w-labeled path in $\mathcal{H}'$ from $\nu(\mathrm{src}_{r'})$ to $\nu(\mathrm{tgt}_{r'})$, with $w \in L_{r'}$.

It can be shown by induction in $\mathcal{T}$ that for every region r:

1. $\mathcal{H}, \nu \models \theta_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r)$,
2. $q0^{\bar{a}}_{\mathcal{A}_r}$ is true in $\mathcal{H}$ at the node $\nu(\bar{a}[\mathrm{src}_r])$, and
3. $qf^{\bar{a}}_{\mathcal{A}_{L_r}}$ is true in $\mathcal{H}$ at the node $\nu(\bar{a}[\mathrm{tgt}_r])$.

In particular, item 1 is true for $r = \mathit{root}$. Since *root* has no parent interface, this means that $\mathcal{H}, \nu \models \theta^{\bar{a}}_{\mathit{root}}(\bar{b}_{\mathit{root}})$ and this contradicts the fact that $\mathcal{H} \models \varphi_5^{\bar{a}}$.

For the right-to-left implication of (†), assume that there is a model $\mathcal{H} \supseteq \mathcal{G}$ satisfying Γ such that $\mathcal{H} \not\models q(\bar{a})$ for every $\bar{a} \in X$. We define the first order model $\mathcal{H}'$ as the one induced by $\mathcal{H}$ plus some interpretation for unary relations $s^{\bar{a}}_{\mathcal{A}_r}$, for each $\bar{a} \in X$, region r of sk and state of s of $\mathcal{A}_r$ in such a way that $\mathcal{H}' \models \varphi$.

We define $s^{\bar{a}}_{\mathcal{A}_r}$ recursively as follows:

i. If r is a leaf in sk then $q0^{\bar{a}}_{\mathcal{A}_r}$ is true in $\mathcal{H}'$ at all elements d of $\mathcal{H}'$ such that for some assignment ν over variables $\bar{b}_r, \mathrm{src}_r$, we have 1) $\mathcal{H}, \nu \models \theta_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r)$ and 2) $d = \nu(\mathrm{src}_r)$.

ii. $s^{\bar{a}}_{\mathcal{A}_r}$ is true at all elements d of $\mathcal{H}'$ such that there is a transition $t \xrightarrow{\alpha} s$ in $\mathcal{A}_r$ and an element c of $\mathcal{H}'$ satisfying $t^{\bar{a}}_{\mathcal{A}_r}$ and $c \xrightarrow{\alpha} d$ in $\mathcal{H}'$.

iii. If r is not a leaf in sk then $q0^{\bar{a}}_{\mathcal{A}_r}$ is true in $\mathcal{H}'$ at all elements d of $\mathcal{H}'$ such that for some assignment ν over variables $\bar{b}_r, \mathrm{src}_r$, we have 1), 2) as in item i, and 3) for all r' child of r we have that $\nu(\mathrm{tgt}_{r'})$ satisfies $qf^{\bar{a}}_{\mathcal{A}_{r'}}$.

By construction, it is clear that $\mathcal{H}' \models \varphi_1 \wedge \varphi_2$ and it is also clear that for any $\bar{a} \in X$ we have $\mathcal{H}' \models \varphi_4^{\bar{a}}$. Let $\bar{a} \in X$, and ν be a valuation that interprets all variables $\bar{b}_r, \mathrm{src}_r$, for all regions r in sk.

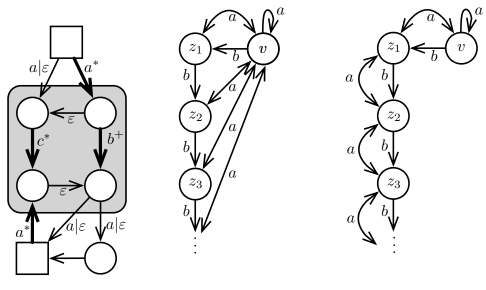
One can show that $\mathcal{H}', \nu \models \theta_r^{\bar{a}}(\bar{b}_r, \mathrm{src}_r) \implies q0^{\bar{a}}_{\mathcal{A}_{L_r}}(\bar{a}[\mathrm{src}_r])$ for all $r \neq \mathit{root}$, and $\mathcal{H}', \nu \not\models \theta^{\bar{a}}_{\mathit{root}}(\bar{b}_{\mathit{root}})$. This shows that $\mathcal{H}' \models \varphi_3^{\bar{a}} \wedge \varphi_5^{\bar{a}}$.

5 Non-Finitely Controllable Cases

We first show that non-finitely controllable cases can be found for any skeleton outside $\mathcal{S}_1$ under frontier-guarded rules (FG), even if we only consider rules with no constants. This means that item 1 of the main theorem holds even for any fragment $\mathcal{T}$ of GNFO containing FG without constants.

Proposition 5. *If $\mathcal{S} \not\subseteq \mathcal{S}_1$, then* OMQA *of* CRPQ($\mathcal{S}$) *under* FG *is non-finitely controllable. This holds even for a singleton set of rules with no constants.*

Proof. We show that if $sk \notin \mathcal{S}_1$ there exists a model $\mathcal{G}$, a query $q \in$ CRPQ($\{sk\}$), a tuple $\bar{t}$ of constants of $\mathcal{G}$, and a finite set Γ of FG rules such that



(a) (b) (c)

Figure 3: (a) A skeleton (notation conventions as in Figure 2) and the labeling of edges. In grey, the distinguished bound cycle. The distinguished edge is the rightmost in the cycle. (b) *Chase*$(\mathcal{G}, \Gamma)$ for the proof of Prop. 5. (c) *Chase*$(\mathcal{G}, \Gamma)$ for the proof of Prop. 8.

1. $q(\bar{t})$ is false in some infinite model extending $\mathcal{G}$ that satisfies Γ (in particular, in *Chase*$(\mathcal{G}, \Gamma)$), and
2. $q(\bar{t})$ is true in every finite extension of $\mathcal{G}$ that satisfies Γ.

As in the proof of Proposition 4, we resort to the Chase.

The skeleton sk gives the shape of q, where we only have to fill in the languages we want to use respecting the finite/infinite cardinality according to the skeleton. Suppose sk has free variables $\bar{x} = x_1, \ldots, x_k$. There must be some infinite edge in a bound cycle. On this edge, we put b^+, and in the rest of the edges of this cycle, we put c^* for every infinite edge and ε for every finite edge. To all the other edges in sk, we put either a^* or $a \mid \varepsilon$ depending on whether they are infinite or finite, respectively. See Figure 3(a).

We then define the following singleton set of FG rules:

$$\forall x, y((x \xrightarrow{a} y \wedge x \xrightarrow{a} x) \Rightarrow \exists z(y \xrightarrow{b} z \wedge x \xrightarrow{a} z \wedge z \xrightarrow{a} x))$$

Let $\mathcal{G} = \{v \xrightarrow{a} v\}$ and let ν be the valuation $\nu(x_i) = v$ for all i. It can be shown that *Chase*$(\mathcal{G}, \Gamma)$ is the infinite model depicted in Figure 3(b). Observe that all edges of $\mathcal{G}$ are labeled with a or b, so if *Chase*$(\mathcal{G}, \Gamma), \nu \models q$, there should to be a nontrivial cycle of b's in *Chase*$(\mathcal{G}, \Gamma)$, which is not true. Therefore *Chase*$(\mathcal{G}, \Gamma), \nu \not\models q$, and item 1 is shown.

To see item 2, let $\mathcal{G}'$ be a finite model that extends $\mathcal{G}$ and satisfies Γ. Let $h :$ *Chase*$(\mathcal{G}, \Gamma) \to \mathcal{G}'$ be a homomorphism. There must be some $i < j$ so that $h(z_i) = h(z_j)$. Consider the following assignment ν' of the variables of q to the vertices of $\mathcal{G}'$: all free variables are mapped to $h(v)$, and all other variables are mapped to $h(z_i)$. We show that every atom in q is true in $\mathcal{G}'$ under ν'.

Suppose $x \xrightarrow{L} y$ is an atom in q. If x and y are free in q, then L is either $a \mid \varepsilon$ or a^*, and $\nu'(x) = \nu'(y) = h(v)$. Then the atom is true, since $\varepsilon \in L$. If x is free and y is bound in q, then L is either $a \mid \varepsilon$ or a^*, and $\nu'(x) = h(v)$ and $\nu'(y) = h(z_i)$. Then the atom is true, since $a \in L$ and in *Chase*$(\mathcal{G}, \Gamma)$ there is an a-labeled edge from v to z_i. If x is bound and y is free, the argument is similar. If x and y are bound in q, then $\nu'(x) = \nu'(y) = h(z_i)$. By construction, either $\varepsilon \in L$, in which case, the atom is true, or $L = b^+$, in which case the atom is also true, since there is a non-trivial cycle $h(z_i) \xrightarrow{b} h(z_{i+1}) \xrightarrow{b} \cdots \xrightarrow{b} h(z_j) = h(z_i)$ in $\mathcal{G}'$. □

In the presence of constants in the rules, a similar proof yields the following.

Proposition 6. *If* $\mathcal{S} \not\subseteq \mathcal{S}_1$, *then* OMQA *of* CRPQ$(\mathcal{S})$ *under* F1 *is non-finitely controllable.*

Proof. We proceed as in the proof of Proposition 5, with the only difference that now the set Γ consists of the rule

$$\forall y(v \xrightarrow{a} y \Rightarrow \exists z(y \xrightarrow{b} z \wedge v \xrightarrow{a} z \wedge z \xrightarrow{a} v))$$

using the constant v. It follows that the Chase generates the same model as in Proposition 5. □

Observe that in the proof above we *need* to use constants in the rules. Indeed, as we know from item 2 of the main theorem, should constants be disallowed we would be in the TF1 class and thus it would be finitely controllable.

Proposition 7. *If* $\mathcal{S} \not\subseteq \mathcal{S}_1$, *then* OMQA *of* CRPQ$(\mathcal{S})$ *under* ID *is not finitely controllable.*

Proof. We reason as in the proof of Proposition 5. We define the labeling of sk as in that proof. We define $\mathcal{G} = \{R(v, w, w), v \xrightarrow{b} w, v \xrightarrow{a} v\}$, and the valuation ν for each free variable x_i of q is defined by $\nu(x_i) = v$. We let

$$\Gamma = \{R(x, y, z) \Rightarrow R(x, z, t),\ \ R(x, y, z) \Rightarrow y \xrightarrow{b} z,$$
$$R(x, y, z) \Rightarrow x \xrightarrow{a} z,\ \ R(x, y, z) \Rightarrow z \xrightarrow{a} x\}$$

(quantifications over variables x, y, z, t are implicit). It follows that *Chase*$(\mathcal{G}, \Gamma)$ is the model defined in Figure 3(b) extended with the facts $\{R(v, w, w), R(v, w, t_0)\} \cup \{R(v, t_i, t_{i+1}) \mid i \geq 0\}$. The rest of the proof is as the one in Proposition 5. □

Proposition 8. *If* $\mathcal{S} \not\subseteq \mathcal{S}_2$, *then* OMQA *of* CRPQ$(\mathcal{S})$ *under* TF1 *is not finitely controllable.*

Proof. Again, we reason as in the proof of Proposition 5. We define the labeling of sk as before, but this time the distinguished cycle C has only bound variables, all of which are at infinite distance from every free variable.

We define $\mathcal{G}$ and ν as in the proof of Proposition 5. The singleton set $\Gamma \subseteq$ TF1 of rules is now:

$$\forall x, y(x \xrightarrow{a} y \Rightarrow \exists z(y \xrightarrow{a} z \wedge z \xrightarrow{a} y \wedge y \xrightarrow{b} z)).$$

It follows that *Chase*$(\mathcal{G}, \Gamma)$ is the infinite model depicted in Figure 3(c). As before, *Chase*$(\mathcal{G}, \Gamma), \nu \not\models q$.

Let $\mathcal{G}'$ be a finite model that extends $\mathcal{G}$ and satisfies Γ. Let $h :$ *Chase*$(\mathcal{G}, \Gamma) \to \mathcal{G}'$ be a homomorphism and let $i < j$ so that $h(z_i) = h(z_j)$. Consider the following assignment ν': if x is a free or bound variable at a finite distance from a free variable, we let $\nu'(x) = h(v)$. For all the other variables x, we let $\nu'(x) = h(z_i)$. Suppose $x \xrightarrow{L} y$ is an atom in q. If x and y are variables at finite distance from a free variable, then $x \xrightarrow{L} y$ is true in $\mathcal{G}', \nu'$ since by construction we have $\varepsilon \in L$. If x is at finite distance from a free variable and y is not (or vice-versa), then L is infinite. Notice that neither x nor y belong to C. By construction $L = a^*$ and then the atom $x \xrightarrow{L} y$ is true in $\mathcal{G}', \nu'$. Suppose that x and y

are variables at infinite distance from any free variable. Then $\nu'(x) = \nu'(y) = h(z_i)$. If both x and y are out of C, then $\varepsilon \in L$ and we are done. If one of them is in C and the other one is not, the argument is similar. If x and y are variables in C, then either $\varepsilon \in L$, or $L = b^+$. In both cases, the atom $x \xrightarrow{L} y$ is true in $\mathcal{G}', \nu'$. □

6 Related Work

Barceló and Fontaine (2017) study the problem of consistent query answering of CRPQ under conjunctive regular path constraints (CRPC). CRPC are defined as the containment of CRPQ queries. The problem of *consistent query answering under superset repairs* they study is equivalent to the FOMQA problem. Their results imply that FOMQA of CRPQ under CRPC is undecidable, and this even holds if the query and constraints are fixed, non-recursive RPQs. However, the regular path constraints are not in GNFO, as they are not guarded nor frontier-guarded, and this is a crucial requisite for their undecidability results. However, if constraints are further restricted to have a single edge on the right-hand side the FOMQA problem becomes decidable, NL in data complexity.

To our knowledge there have not been other works on FOMQA for CRPQ. Still, we review here some of the works most relevant to our setting.

Finite OMQA Prior work on *finite* ontology mediated query answering has been done for unions of Conjunctive Queries (UCQ) or the equi-expressive class of positive existential queries. FOMQA of UCQ's and positive-existential queries under guarded TGD and frontier-guarded TGD is finitely controllable and decidable in 2EXPTIME and PTIME in data complexity (Baget et al. 2011b). This 2EXPTIME bound holds even for GNFO queries and constraints. These results are a consequence of the finite model property and decidability of the satisfiability problems for GNFO (Bárány, ten Cate, and Segoufin 2015) and GFO (Bárány, Gottlob, and Otto 2014). See (Segoufin 2017) for a survey on GNFO. Whether FOMQA of CRPQ under GTGD or any of its extensions above is decidable is an open question, one of the issues being that CRPQ lacks finite-controllability with respect to GTGD.

Gogacz, Ibáñez-García, and Murlak (2018) study the FOMQA for UCQ in the presence of expressive description logics with transitive roles. We, on the other hand, study query languages with transitivity but on ontologies with no transitive roles. In the presence of transitive roles, FOMQA is undecidable already for $\mathcal{SHOIF}$ (Rudolph 2016), and they show decidability for three fragments thereof: $\mathcal{SOI}$, $\mathcal{SOF}$, and $\mathcal{SIF}$. Another known decidable ontology for positive existential queries FOMQA is the Horn fragment of $\mathcal{ALCIF}$ (Ibáñez-García, Lutz, and Schneider 2014) (without transitivity). Without transitivity, Rudolph (2016) shows that FOMQA for 2RPQ under $\mathcal{ALCOIF}$ is undecidable.

Gogacz et al. (2019) have studied a related problem, the finite query entailment for positive existential queries under $\mathcal{SQ}$ ontologies, possibly extended with nominals ($\mathcal{O}$) and inverse roles ($\mathcal{I}$).

OMQA for CRPQ On the other hand, there have been works on regular path queries, but focused on OMQA (*i.e.*, on arbitrary, unrestricted models). Baget et al. (2017) have studied the OMQA for C2RPQ under guarded existential rules (*cf.* Proposition 2) However, since C2RPQ are non-finitely controllable under guarded existential rules, this does not bring any results on the finitary version of the problem, FOMQA. Our work can be seen as the answer to the question: Which classes of C2RPQ are finitely controllable for guarded existential rules, and thus amenable to the techniques developed by Baget et al.?

Gutiérrez-Basulto, Ibáñez-García, and Jung (2018) study (unrestricted) OMQA for positive existential two-way Regular Path Queries (in particular containing CRPQ and C2RPQ) with respect to the $\mathcal{SQ}$ description logics, supporting transitive roles ($\mathcal{S}$) and number restrictions ($\mathcal{Q}$). They show a tree-like model property yielding a decidable, 2ExpTime, upper bound. As they observe, $\mathcal{SQ}$ lacks finite-controllability even for the single-atom Boolean CQ and this work does not shed light on the FOMQA problem.

Finite controllability Finite controllability was introduced in (Johnson and Klug 1984), where it was shown that query answering of CQs under ID is decidable (PSPACE-complete in the general case and NP-complete for bounded arities). In that work, the problem of finite controllability for ID was left open. The notion of finite controllability was later studied in (Rosati 2006; Rosati 2011) in the context of ontologies given by ID and functional dependencies, and UCQ, solving the aforementioned open question by using a finite model generation procedure, called *finite chase*. Finitely controllability was later studied in (Bárány, Gottlob, and Otto 2014), generalising Rosati's finite chase to show finite controllability of ontologies in guarded first-order and UCQs. For Datalog$^{\pm}$, finite controllability holds for several fragments (Gogacz and Marcinkowski 2013; Amendola, Leone, and Manna 2018; Baget et al. 2011a; Gottlob, Manna, and Pieris 2018; Civili and Rosati 2012).

7 Discussion

An inspection of the proofs for the non-finitely controllable cases shows that even the most basic CRPQ, having only regular expressions of the form a^*, ε and $a \mid \varepsilon$ (where a is a role name), and thus that our main theorem also holds when we restrict the query language to have these very simple expressions.

The standard definition of UC2RPQ that we use does not make use of constants. However, in practice constants are important for querying. Our results can be easily extended to UC2RPQ with constants and even conjunctions of relations of arbitrary arity (in particular generalizing CQs with constants on arbitrary arity relations).

The current work can be seen as a step forward in the investigation of the seemingly difficult question of FOMQA for CRPQ under GNFO-definable rules.

Acknowledgements

Work supported by Franco-Argentinian International Research Project (IRP) SINFIN and grant PICT-2016-0215. The first author acknowledges the support of ANR DéLTA (grant ANR-16-CE40-0007) and ANR QUID (grant ANR-18-CE40-0031).

References

Amendola, G.; Leone, N.; and Manna, M. 2018. Finite controllability of conjunctive query answering with existential rules: Two steps forward. In *IJCAI*, 5189–5193.

Andréka, H.; Németi, I.; and van Benthem, J. 1998. Modal languages and bounded fragments of predicate logic. *J. Philos. Log.* 27(3):217–274.

Arenas, M.; Bertossi, L. E.; and Chomicki, J. 1999. Consistent query answers in inconsistent databases. In *ACM Symposium on Principles of Database Systems (PODS)*, 68–79. ACM Press.

Baget, J.-F.; Leclère, M.; Mugnier, M.-L.; and Salvat, E. 2011a. On rules with existential variables: Walking the decidability line. *Artificial Intelligence* 175(9-10):1620–1654.

Baget, J.; Mugnier, M.; Rudolph, S.; and Thomazo, M. 2011b. Walking the complexity lines for generalized guarded existential rules. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 712–717. IJCAI/AAAI.

Baget, J.; Bienvenu, M.; Mugnier, M.; and Thomazo, M. 2017. Answering conjunctive regular path queries over guarded existential rules. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 793–799. ijcai.org.

Bárány, V., and Bojańczyk, M. 2012. Finite satisfiability for guarded fixpoint logic. *Information Processing Letters (IPL)* 112(10):371–375.

Bárány, V.; Gottlob, G.; and Otto, M. 2014. Querying the guarded fragment. *Logical Methods in Computer Science (LMCS)* 10(2).

Bárány, V.; ten Cate, B.; and Segoufin, L. 2015. Guarded negation. *Journal of the ACM* 62(3):22:1–22:26.

Barceló, P., and Fontaine, G. 2017. On the data complexity of consistent query answering over graph databases. *Journal of Computer and System Sciences (JCSS)* 88:164–194.

Benedikt, M.; Bourhis, P.; and Vanden Boom, M. 2016. A step up in expressiveness of decidable fixpoint logics. In *Annual Symposium on Logic in Computer Science (LICS)*, 817–826. ACM Press.

Bienvenu, M., and Ortiz, M. 2015. Ontology-mediated query answering with data-tractable description logics. In Faber, W., and Paschke, A., eds., *Reasoning Web. Web Logic Rules - 11th International Summer School 2015*, volume 9203 of *Lecture Notes in Computer Science*, 218–307. Springer.

Bonifati, A.; Martens, W.; and Timm, T. 2019. Navigating the maze of Wikidata query logs. In *World Wide Web Conference (WWW)*, 127–138.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2009. Datalog extensions for tractable query answering over ontologies. In Virgilio, R. D.; Giunchiglia, F.; and Tanca, L., eds., *Semantic Web Information Management - A Model-Based Perspective*. Springer. 249–279.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general datalog-based framework for tractable query answering over ontologies. *J. Web Semant.* 14:57–83.

Civili, C., and Rosati, R. 2012. A broad class of first-order rewritable tuple-generating dependencies. In *International Datalog 2.0 Workshop*, 68–80. Springer.

Dolby, J.; Fokoue, A.; Kalyanpur, A.; Ma, L.; Schonberg, E.; Srinivas, K.; and Sun, X. 2008. Scalable grounded conjunctive query evaluation over large and expressive knowledge bases. In Sheth, A. P.; Staab, S.; Dean, M.; Paolucci, M.; Maynard, D.; Finin, T. W.; and Thirunarayan, K., eds., *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, 403–418. Springer.

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

Gogacz, T., and Marcinkowski, J. 2013. Converging to the chase–a tool for finite controllability. In *Annual Symposium on Logic in Computer Science (LICS)*, 540–549. IEEE.

Gogacz, T.; Gutiérrez-Basulto, V.; Ibáñez-García, Y.; Jung, J. C.; and Murlak, F. 2019. On finite and unrestricted query entailment beyond SQ with number restrictions on transitive roles. In *International Workshop on Description Logics (DL)*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Gogacz, T.; Ibáñez-García, Y. A.; and Murlak, F. 2018. Finite query answering in expressive description logics with transitive roles. In *Principles of Knowledge Representation and Reasoning (KR)*, 369–378. AAAI Press.

Gottlob, G.; Manna, M.; and Pieris, A. 2018. Finite model reasoning in hybrid classes of existential rules. In *IJCAI*, 1831–1837.

Gutiérrez-Basulto, V.; Ibáñez-García, Y. A.; and Jung, J. C. 2018. Answering regular path queries over SQ ontologies. In *AAAI Conference on Artificial Intelligence*, 1845–1852. AAAI Press.

Heymans, S.; Ma, L.; Anicic, D.; Ma, Z.; Steinmetz, N.; Pan, Y.; Mei, J.; Fokoue, A.; Kalyanpur, A.; Kershenbaum, A.; Schonberg, E.; Srinivas, K.; Feier, C.; Hench, G.; Wetzstein, B.; and Keller, U. 2008. Ontology reasoning with large data repositories. In Hepp, M.; Leenheer, P. D.; de Moor, A.; and Sure, Y., eds., *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications*, volume 7 of *Semantic Web and Beyond: Computing for Human Experience*. Springer. 89–128.

Ibáñez-García, Y. A.; Lutz, C.; and Schneider, T. 2014. Finite model reasoning in horn description logics. In *Principles of Knowledge Representation and Reasoning (KR)*. AAAI Press.

Johnson, D. S., and Klug, A. 1984. Testing containment of conjunctive queries under functional and inclusion dependencies. *Journal of Computer and system Sciences* 28(1):167–189.

Kontchakov, R.; Rodriguez-Muro, M.; and Zakharyaschev, M. 2013. Ontology-based data access with databases: A short course. In Rudolph, S.; Gottlob, G.; Horrocks, I.; and van Harmelen, F., eds., *Reasoning Web. Semantic Technologies for Intelligent Data Access - 9th International Summer School 2013, Mannheim, Germany*, volume 8067 of *Lecture Notes in Computer Science*, 194–229. Springer.

Malyshev, S.; Krötzsch, M.; González, L.; Gonsior, J.; and Bielefeldt, A. 2018. Getting the most out of Wikidata: Semantic technology usage in Wikipedia's knowledge graph. In *International Semantic Web Conference (ISWC)*, 376–394.

Mugnier, M., and Thomazo, M. 2014. An introduction to ontology-based query answering with existential rules. In Koubarakis, M.; Stamou, G. B.; Stoilos, G.; Horrocks, I.; Kolaitis, P. G.; Lausen, G.; and Weikum, G., eds., *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, volume 8714 of *Lecture Notes in Computer Science*, 245–278. Springer.

Ortiz, M., and Simkus, M. 2012. Reasoning and query answering in description logics. In Eiter, T., and Krennwallner, T., eds., *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012*, volume 7487 of *Lecture Notes in Computer Science*, 1–53. Springer.

Poggi, A.; Lembo, D.; Calvanese, D.; Giacomo, G. D.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J. Data Semantics* 10:133–173.

Rosati, R. 2006. On the decidability and finite controllability of query processing in databases with incomplete information. In *ACM Symposium on Principles of Database Systems (PODS)*, 356–365.

Rosati, R. 2011. On the finite controllability of conjunctive query answering in databases under open-world assumption. *Journal of Computer and System Sciences* 77(3):572–594.

Rudolph, S. 2016. Undecidability results for database-inspired reasoning problems in very expressive description logics. In *Principles of Knowledge Representation and Reasoning (KR)*, 247–257. AAAI Press.

Segoufin, L. 2017. A survey on guarded negation. *SIGLOG News* 4(3):12–26.