HAL
open science

Answering Conjunctive Regular Path Queries over Guarded Existential Rules

Jean-François Baget, Meghyn Bienvenu, Marie-Laure Mugnier, Michaël Thomazo

To cite this version:

Jean-François Baget, Meghyn Bienvenu, Marie-Laure Mugnier, Michaël Thomazo. Answering Conjunctive Regular Path Queries over Guarded Existential Rules. IJCAI: International Joint Conference on Artificial Intelligence, Aug 2017, Melbourne, Australia. lirmm-01632224

HAL Id: lirmm-01632224

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01632224v1

Submitted on 9 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Answering Conjunctive Regular Path Queries over Guarded Existential Rules

Jean-François Baget
Inria, France
baget@lirmm.fr

Meghyn Bienvenu
CNRS, France
meghyn@lirmm.fr

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

Michael Thomazo
Inria, France
michael.thomazo@inria.fr

Abstract

Ontology-mediated query answering is concerned with the problem of answering queries over knowledge bases consisting of a database instance and an ontology. While most work in the area focuses on conjunctive queries (CQs), navigational queries are gaining increasing attention. In this paper, we investigate the complexity of answering two-way conjunctive regular path queries (CRPQs) over knowledge bases whose ontology is given by a set of guarded existential rules. We first consider the subclass of linear existential rules and show that CRPQ answering is EXPTIME-complete in combined complexity and NL-complete in data complexity, matching the recently established bounds for answering non-conjunctive RPQs. For guarded rules, we provide a non-trivial reduction to the linear case, which allows us to show that the complexity of CRPQ answering is the same as for CQs, namely 2EXPTIME-complete in combined complexity and PTIME-complete in data complexity.

1 Introduction

In ontology-mediated query answering (OMQA), a database is enriched with an ontological layer that expresses domain knowledge, and queries are answered by taking into account the information from both the data and ontology. Two main families of ontology languages are considered in this setting, namely description logics (DLs) and existential rules (see e.g. the survey chapters [Ortiz and Simkus, 2012; Kontchakov *et al.*, 2013; Bienvenu and Ortiz, 2015] on OMQA with DLs and [Calì *et al.*, 2009a; Mugnier and Thomazo, 2014] for existential rules).

Although most work on OMQA to date has focused on the fundamental class of conjunctive queries (CQs), navigational queries are gaining increasing attention. The simplest such queries are regular path queries (RPQs, [Florescu *et al.*, 1998]), which ask for paths defined by a given regular language, hence allowing for a controlled form of recursion over binary predicates. Many extensions of RPQs have been considered, including *conjunctive regular path queries* (CRPQs) which generalize both RPQs and CQs. In particular, SPARQL 1.1, the new W3C standard for querying RDF data, extends its core queries, which roughly correspond to CQs, to CRPQs.

In recent years, such queries have been investigated for a variety of description logics, ranging from highly expressive DLs of the $\mathcal{Z}$ family [Calvanese *et al.*, 2007; 2009; 2014], to Horn DLs like Horn-$\mathcal{SROIQ}$ [Ortiz *et al.*, 2011] and lightweight DLs of the DL-Lite and $\mathcal{EL}$ families [Stefanoni *et al.*, 2014; Bienvenu *et al.*, 2015].

When the data are not only graphs, as in RDF and DL fact bases, but more generally hypergraphs, or, in logical terms, sets of facts with any predicate arity, navigational queries are still very relevant as binary relations remain central in data and ontological modeling. While the complexity landscape for answering navigational queries in the presence of DL ontologies is now quite clear, the combination of navigational queries and existential rules has only just recently begun to be explored. In [Bienvenu and Thomazo, 2016], the authors studied the complexity of RPQ answering over *linear* existential rules, a natural generalization of DL-Lite$_\mathcal{R}$. It was shown that the problem complexity remains the same as for DL-Lite$_\mathcal{R}$ in terms of data complexity (NL-complete) and combined complexity with bounded predicate arity (PTIME-complete). However, in the unbounded arity case, the combined complexity rises to EXPTIME-complete. Interestingly, in DL-Lite$_\mathcal{R}$, RPQ answering is easier than CQ answering for combined complexity (P- vs. NP-complete [Bienvenu *et al.*, 2015]), while the opposite holds for linear rules without arity restriction (CQ answering being PSPACE-complete [Calì *et al.*, 2009a]).

In this paper, we make a step towards a better understanding of the complexity of answering CRPQs over existential rule bases, focusing on a central class of existential rules, namely *guarded* rules, of which linear rules are a special case [Calì *et al.*, 2008; 2009b]. Note that the problem is already known to be decidable (from [Rudolph and Krötzsch, 2013]).

Summary of the contributions. We obtain tight complexity results for both combined and data complexities over guarded rules and their linear subclass. Namely, CRPQ answering in the linear case is EXPTIME-complete in combined complexity, PSPACE-complete in combined complexity with bounded arity and NL-complete in data complexity (Thm 1); hence, it is not more difficult than RPQ answering, except for combined complexity with bounded arity (in which

case the complexity is again the same as for DL-Lite$_\mathcal{R}$). In the guarded case, CRPQ answering is 2EXPTIME-complete in combined complexity, EXPTIME-complete in combined complexity with bounded arity, and PTIME-complete in data complexity (Thm 2); hence, it is not more difficult than plain CQ answering.

To achieve these results, we first investigate the case of linear rules and provide a CRPQ answering algorithm that uses RPQ answering as an oracle and runs within the mentioned complexity classes. The matching lower bounds come from earlier results on RPQ and CQ answering. We next provide a non-trivial reduction of the guarded case to the linear case. This translation involves a double exponential blow-up of the rule base (while the instance only grows exponentially in the predicate arity). However, a careful analysis of the algorithm provided for linear rules shows that it actually runs in 2EXPTIME with respect to the input guarded knowledge base (and in EXPTIME in the case of bounded-arity rules).

2 Preliminaries

We consider a *vocabulary* composed of a finite set of predicates and an infinite set of constants. A (standard) *atom* α has the form $r(\mathbf{t})$ where r is a predicate of arity n and $\mathbf{t}$ is a tuple of terms (i.e., variables or constants) with $|\mathbf{t}| = n$. For $1 \leq i \leq |\mathbf{t}|$, we denote by $\alpha[i]$ the term at position i in α. We denote by *terms*(α) the set of terms in α and extend the notation to a set of atoms.

A *ground* atom contains only constants. An *instance* is a finite set of facts, which are grounds atoms. A *conjunctive query* (CQ) has the form $q(\mathbf{x}) = \exists \mathbf{y}.B$ where $\mathbf{x}$ and $\mathbf{y}$ are disjoint tuples of variables, and B is a conjunction of atoms such that *terms*$(B) = \mathbf{x} \cup \mathbf{y}$. A tuple of constants $\mathbf{a} = \pi(\mathbf{x})$, where π denotes a substitution, is a *certain answer* to $q(\mathbf{x})$ over I if $I \models q(\mathbf{a})$, where $q(\mathbf{a}) = \exists \mathbf{y}.\pi(B)$ and $\models$ denotes classical logical entailment. A CQ is *Boolean* if $\mathbf{x} = \emptyset$.

We often identify (the existential closure of) a conjunction of atoms with the set of these atoms. Given two sets of atoms A and B, a *homomorphism* from B to A is a substitution π of the variables in B by the terms in A such that $\pi(B) \subseteq A$. It holds that $A \models B$ iff there is a homomorphism from B to A. Given a (possibly infinite) set of atoms A and a set of terms T, $A_{|T}$ denotes the subset of A containing the atoms whose arguments are a subset of T.

Existential Rules An *existential rule* (or simply *rule*) is of the form $\forall \mathbf{x} \forall \mathbf{y}.[B(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}.H(\mathbf{y}, \mathbf{z})]$ where B and H are non-empty conjunctions of atoms on variables, respectively called the *body* and the *head* of the rule, and $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ are pairwise disjoint. The variables of $\mathbf{z}$ are called *existential variables*. For brevity, quantifiers are often omitted, and we write $B \rightarrow H$. We will make the classical assumption that the head of a rule is restricted to a single atom. A set of atoms G is *guarded* if it contains an atom α (called a guard) such that *terms*(α) = *terms*(G). Since G may have several guards, we use the notation (G, α) to specify that α is the considered guard. A rule is *guarded* if its body is guarded. A rule is *linear* if its body contains a single atom. A *knowledge base* (KB) is of the form $\mathcal{K} = (I, \mathcal{R})$, where I is an instance and $\mathcal{R}$ a set of existential rules. Let $\mathcal{K} = (I, \mathcal{R})$ be a KB and $q(\mathbf{x})$ be a CQ. A *certain answer* to q over $\mathcal{K}$ is a tuple of constants $\mathbf{a}$ such that $I, \mathcal{R} \models q(\mathbf{a})$.

A rule $R = B \rightarrow \alpha$ is *applicable* to a set of atoms A if there is a homomorphism π from B to A. The result of the application of R on A according to π is the atom $\pi^s(\alpha)$, where π^s extends π by assigning a distinct fresh variable (called a *null*) to each existential variable. The application is said to *generate* the set of atoms $A \cup \pi^s(\alpha)$, which is uniquely defined up to the choice of nulls.

The *chase* is the fundamental tool on existential rules. Starting from the instance, it performs rule applications in a breadth-first manner. Several variants of the chase are known. We consider here the simplest variant, known as the oblivious chase, which at each breadth-first step performs all new rule applications, and proceeds until no rule is applicable according to a new homomorphism. All the chase variants compute a so-called *universal model* of the KB, i.e., a possibly infinite set of facts seen as a classical logical interpretation, which is a model of the KB that maps by homomorphism to any other model of the KB. The result of the chase is also called chase. It follows that for any $\mathcal{K} = (I, \mathcal{R})$ and CQ q, $(I, \mathcal{R}) \models q$ iff the chase of $\mathcal{K}$ is a model of q iff there is a homomorphism from q to the chase of $\mathcal{K}$.

(Conjunctive) Regular Path Queries A *regular language* can be represented either by a regular expression or by a non-deterministic finite automaton (NFA). Let Σ be a finite set of symbols. A regular expression $\mathcal{E}$ over Σ is defined by the grammar: $\mathcal{E} \rightarrow \varepsilon \mid a \mid \mathcal{E} \cdot \mathcal{E} \mid \mathcal{E} + \mathcal{E} \mid \mathcal{E}^*$, where $a \in \Sigma$ and ε denotes the empty word. An NFA over Σ is a tuple $\mathbb{A} = (S, \Sigma, \delta, s_0, F)$, where S is a finite set of states, $\delta \subseteq S \times \Sigma \times S$ is the transition relation, $s_0 \in S$ is the initial state and $F \subseteq S$ is the set of final states.

We denote by $\mathcal{P}_2$ the binary predicates in our vocabulary and set $\mathcal{P}_2^{\pm} = \mathcal{P}_2 \cup \{r^- \mid r \in \mathcal{P}_2\}$. A *path atom* takes the form $\Lambda(t, t')$, where Λ is a regular language over $\mathcal{P}_2^{\pm}$ and t, t' are terms. Note that standard atoms with binary predicates are a special case of path atoms. We assume w.l.o.g. that automata associated with distinct path atoms have disjoint sets of states.

A *conjunctive two-way regular path query* (CRPQ[1]) has the form $q(\mathbf{x}) = \exists \mathbf{y}.B$ and is defined as a CQ except that B is a conjunction of standard and path atoms.

We extend the usual definition of an interpretation $\mathcal{I} = (\Delta, .^\mathcal{I})$ being a *model* of a conjunction of atoms by saying that, for a regular language Λ, $(d, d') \in \Lambda^\mathcal{I}$ if there are $r_1 \ldots r_n \in \Lambda$ and $(d = d_0, \ldots, d_n = d') \in \Delta^{n+1}$ such that $\forall 1 \leq i \leq n$, if $r_i \in \mathcal{P}_2$ then $(d_{i-1}, d_i) \in r_i^\mathcal{I}$, otherwise $r_i = s_i^-$ and $(d_i, d_{i-1}) \in s_i^\mathcal{I}$. We call $p = d_0 r_1 d_1 \cdots r_n d_n$ a *path* in $\mathcal{I}$. A *match* for a Boolean CRPQ q in an interpretation $\mathcal{I} = (\Delta, .^\mathcal{I})$ is a mapping π from the terms in q to elements of Δ such that *(i)* $\pi(a) = a^\mathcal{I}$ for each constant a; *(ii)* $\pi(\mathbf{t}) \in r^\mathcal{I}$ for each standard atom $r(\mathbf{t})$ in q; and *(ii)* $(\pi(t), \pi(t')) \in \Lambda^\mathcal{I}$ for each path atom $\Lambda(t, t')$ in q. Note that the restriction of π to the standard atoms of q is a homomorphism. Then $\mathcal{I}$ is a

[1] As we only consider the two-way variant, we will use the abbreviation (C)RPQ instead of the more traditional (C)2RPQ.

model of q if and only if there is a match for q in $\mathcal{I}$. Certain answers are defined in the same way as for CQs. Since the chase of $\mathcal{K} = (I, \mathcal{R})$ is a universal model, and CRPQs are closed by homomorphism, $I, \mathcal{R} \models q$ if and only if there is a match π of q in the chase of $\mathcal{K}$. A *path of terms* in the chase is the natural translation of a path in an interpretation.

The *CRPQ Answering problem* asks, given a KB $\mathcal{K} = (I, \mathcal{R})$, a CRPQ $q(\mathbf{x})$ and a tuple of constants $\mathbf{a}$ whether $\mathbf{a}$ is an answer to q w.r.t. $\mathcal{K}$. The *CRPQ Entailment problem* asks, given a KB $\mathcal{K} = (I, \mathcal{R})$ and a Boolean CRPQ q, if $I, \mathcal{R} \models q$. Both problems are classically polynomially reducible one to another. To simplify the presentation, we will study the CRPQ Entailment problem.

3 Chase Graph of Linear Rules

With the chase is naturally associated a derivation graph, in which nodes are labelled by atoms and edges represent rule applications, i.e., there is an edge (ν, ν') if the application of a rule to the atom that labels ν has produced the atom that labels ν'. In the case of linear rules, this graph is a forest. Several derivation graphs can be associated with the same chase, since the same atom may be produced in different ways. However, in our proofs we need to be independent from a particular graph and to have an edge (ν, ν') if and only if a rule application to the atom that labels ν can produce the atom that labels ν'. Hence, we will consider a particular graph that encodes all derivations associated with a given chase. This graph is still a forest.

Definition 1 *Let I be an instance and $\mathcal{R}$ be a set of linear rules. The* chase graph *associated with $(I, \mathcal{R})$, denoted by $\mathcal{CG}(I, \mathcal{R})$ (and simply $\mathcal{CG}$ if the context is clear), is a possibly infinite node-labeled directed forest built as follows (we use ℓ for the labeling function):*

- *the roots are in bijection with I via ℓ;*
- *for every node ν and rule $R \in \mathcal{R}$ applicable to $\ell(\nu)$, there is exactly one edge from ν to a node ν' labeled by the atom resulting from this rule application.*

Note that an atom can label two distinct nodes, effectively encoding two different ways of generating that same atom. Given $(I, \mathcal{R})$, the chase graph is uniquely defined, up to the choice of nulls. We formally define the *chase* as the set of all atoms that label $\mathcal{CG}(I, \mathcal{R})$. We denote it by $\mathsf{chase}(I, \mathcal{R})$.

We can now express a fundamental property of linear rules: each atom from the instance can be 'chased' independently.

Property 1 *Let $\mathcal{R}$ be a set of linear rules and I be an instance. Then $\mathsf{chase}(I, \mathcal{R}) = \cup_{\alpha \in I} \mathsf{chase}(\{\alpha\}, \mathcal{R})$.*

For ν and ν' in the same tree of $\mathcal{CG}$, $glb(\nu, \nu')$ denotes their greatest lower bound (i.e., their deepest common ancestor).

An essential property of the chase graph is that, for any null z, the nodes labeled by the atoms that contain z form a connected component. This is also true for the initial terms inside each tree. In the next proposition, to distinguish between undirected paths in $\mathcal{CG}$ and the previously introduced paths (on terms in the chase), we call the former *atom-paths*.

Proposition 1 (Connectivity property) *Let ν_1 and ν_2 be two nodes of $\mathcal{CG}(I, \mathcal{R})$ and $t \in \mathsf{terms}(\ell(\nu_1)) \cap \mathsf{terms}(\ell(\nu_2))$. Then:*

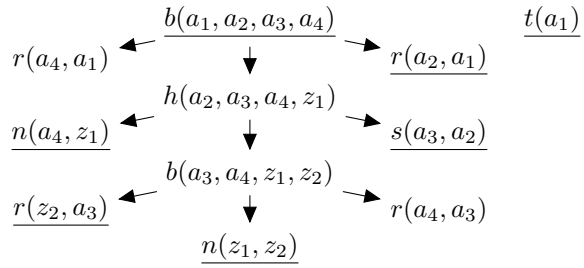

Figure 1: Part of $\mathcal{CG}(I_e, \mathcal{R}_e)$

- *either t is contained in I and there are atom-paths P_1 from ν_1 to a root ν_1' and P_2 from ν_2 to a root ν_2' such that for every node $\nu \in P_1 \cup P_2$, $\ell(\nu)$ contains t;*
- *or there is an atom-path P from ν_1 to ν_2, and it is such that for every node $\nu \in P$, $\ell(\nu)$ contains t.*

The next proposition specifies the relationships between a path in the chase and an atom-path in the chase graph. It strongly relies on the connectivity property.

Proposition 2 *Let ν_1 and ν_2 be two nodes in $\mathcal{CG}$ under a common root. Let $p = t_1 \dots t_p$ be a path in the chase such that t_1 occurs in $\ell(\nu_1)$ and t_p occurs in $\ell(\nu_2)$. Then for any node ν on the atom-path from ν_1 to ν_2 in $\mathcal{CG}$, $\ell(\nu)$ contains a term t_i from p.*

Example 1 *Let $\mathcal{R}_e$ contain the following rules:*

$$\begin{array}{ll} b(x,y,z,t) \to h(y,z,t,u) & h(x,y,z,t) \to b(y,z,t,u) \\ b(x,y,z,t) \to r(y,x) & h(x,y,z,t) \to s(y,x) \\ b(x,y,z,t) \to n(z,t) & h(x,y,z,t) \to n(z,t) \\ b(x,y,z,t) \to r(t,x) & \end{array}$$

Part of $\mathcal{CG}(I_e = \{b(a_1, a_2, a_3, a_4), t(a_1)\}, \mathcal{R}_e)$ is represented in Fig. 1. For clarity, we represent directly the atoms that label nodes (here, the labeling is injective). The terms a_i are constants while the terms z_i are nulls. Let us consider the following path of terms in the chase: $p = a_4\ n\ z_1\ n\ z_2\ r\ a_3\ s\ a_2\ r\ a_1$. We can see that the structure associated with p has no direct connection with the structure of the chase graph. However, Prop. 2 allows us to make a connection: if we take any pair of nodes ν_1 and ν_2 labeled by atoms that respectively contain a_4 and a_1, we can see that along the atom-path between ν_1 and ν_2 all the nodes contain at least a term from p. Now, consider the following Boolean CRPQ q (existential quantifiers omitted):

$$b(x_1, x_2, x_3, x_4) \wedge (n^*(rsr)^*)(x_4, x_5) \wedge t(x_5)$$

There is a match of q to the partial chase: $x_1 \mapsto a_1$, $x_2 \mapsto a_2$, $x_3 \mapsto a_3$, $x_4 \mapsto a_4$, $x_5 \mapsto a_1$. Note that $(n^*(rsr)^*)(x_4, x_5)$ is mapped to the path p. The atoms of the chase graph involved in this match are underlined.

4 CRPQ Entailment over Linear Rules

In this section, we describe a method for CRPQ entailment in the presence of linear rules, which allows us to pinpoint

the worst-case complexity of this problem. Throughout this section, we let I be an instance, $\mathcal{R}$ a set of linear rules, and q be the Boolean CRPQ we wish to answer.

As it will be important to keep track of the paths in the chase that link the terms of a pair of atoms, we define the set of transitions between two atoms.

Definition 2 *Let α_1 and α_2 be two atoms of* chase$(I, \mathcal{R})$ *of arities k_1 and k_2 respectively. The* transitions from α_1 to α_2*, denoted by $\mathcal{T}_{q,I,\mathcal{R}}(\alpha_1, \alpha_2)$, is the set of quadruples (i_1, s_1, i_2, s_2) such that $1 \leq i_1 \leq k_1$, $1 \leq i_2 \leq k_2$, and there exists an automaton $\mathbb{A}$ in q and a path p in* chase$(I, \mathcal{R})$ *going from $\alpha_1[i_1]$ to $\alpha_2[i_2]$ such that $\lambda(p)$ takes $\mathbb{A}$ from s_1 to s_2. More generally, by* transition*, we will mean a quadruple (i_1, s_1, i_2, s_2) where s_1, s_2 are states appearing in q, and i_1, i_2 do not exceed the maximum predicate arity in $I \cup \mathcal{R}$.*

We now introduce the key technical notion underlying our approach, namely, *proof schemes*. Formally, a proof scheme is a directed forest whose nodes are atoms, together with transitions for different pairs of atoms occurring in the proof scheme. Intuitively, proof schemes correspond to a subset of chase atoms, with the transitions placing requirements on the paths linking these atoms.

Definition 3 *A* proof scheme *is a pair $\mathbb{P} = (\mathcal{F}, \mathcal{T})$ such that:*

- *$\mathcal{F} = (V, E)$ is a directed forest;*
- *V is a set of atoms;*
- *for each null n, the set of nodes in which n occurs form a connected subgraph of $\mathcal{F}$;*
- *$\mathcal{T}$ is a set of pairs $(\tau, (\alpha_1, \alpha_2))$, where τ is a transition, α_1 and α_2 belong to V;*
- *for each $(\tau, (\alpha_1, \alpha_2)) \in \mathcal{T}$, either $\alpha_1 = \alpha_2$, $(\alpha_1, \alpha_2) \in E$, $(\alpha_2, \alpha_1) \in E$, or both α_1 and α_2 are roots.*

The terms of a proof scheme, denoted by terms$(\mathbb{P})$*, is the union of all the terms of atoms in V.*

We next define *valid* proof schemes. The validity conditions enforce that the proof scheme behaves as intended: its atoms can be mapped into the chase graph in a way that respects the forest structure, and the transitions in the proof scheme are indeed realized in the chase. Note that valid proof schemes are not far, in essence, from the proof generators used in [Gottlob *et al.*, 2015] for deciding CQ entailment in linear rule KBs; however, the presence of transitions and their interactions with linear rules make the validity check much more complex. The idea is to consider the following brute-force algorithm for CRPQ answering: enumerate all proof schemes, check whether the considered proof scheme is valid w.r.t $I, \mathcal{R}$ and q and whether there is a 'match' of q in the proof scheme. To obtain our complexity results, we will upper bound the size of the proof schemes that need to be considered and the complexity of the validity check.

To formalize the notion of a valid proof scheme, we need the classical notion of the *type of an atom*, as well as the notion of *isomorphic atoms with respect to a bijection*, which allows us to identify atoms that 'act the same'.

Definition 4 *The* type of an atom $r(t_1 \dots t_k)$ *is the pair $(r, \mathcal{P})$ where $\mathcal{P}$ is the partition of $\{1, \dots, k\}$ induced by term equality (i.e., i and j are in the same class of $\mathcal{P}$ iff $t_i = t_j$).*

Definition 5 *Two atoms α_1 and α_2 are* isomorphic with respect to a bijection *b from a set of terms T_1 to a set of terms T_2 if they have the same type $(r, \mathcal{P})$ and, for any class C of $\mathcal{P}$, if $\alpha_1[C] \in T_1$ then $\alpha_2[C] = b(\alpha_1[C])$, otherwise $\alpha_2[C] \notin T_2$, and $\alpha_1[C]$ and $\alpha_2[C]$ are both nulls.*

Definition 6 *A proof scheme $\mathbb{P} = (\mathcal{F}, \mathcal{T})$ with $\mathcal{F} = (V, E)$ is* valid *w.r.t. $(I, \mathcal{R}, q)$ if there is an injective mapping w from* terms$(\mathbb{P})$ *to* terms(chase$(I, \mathcal{R})$) *such that:*

- *for each root atom $\alpha \in V$, $w(\alpha) = \alpha$ and $\alpha \in I$;*
- *for each atom $\alpha \in V$, $w(\alpha)$ belongs to* chase$(I, \mathcal{R})$*;*
- *for each atom $\alpha \in V$, α and $w(\alpha)$ are isomorphic with respect to the identity function on* terms(I)*;*
- *for each edge (α_1, α_2), it holds that:*
  - *α_2 and $w(\alpha_2)$ are isomorphic with respect to the bijection b from* terms(α_1) *to* terms$(w(\alpha_1))$ *induced by their common type;*
  - *$\mathcal{CG}(I, \mathcal{R})$ contains nodes ν_1, ν_2 such that $\nu_1 \prec \nu_2$, $\ell(\nu_1) = w(\alpha_1)$, $\ell(\nu_2) = w(\alpha_2)$.*
- *for each $(\tau, (\alpha_1, \alpha_2)) \in \mathcal{T}$: $\tau \in \mathcal{T}_{q,I,\mathcal{R}}(w(\alpha_1), w(\alpha_2))$.*

In a proof scheme, transitions label pairs of atoms (α_1, α_2) such that $\alpha_1 = \alpha_2$, or both atoms are roots, or one is the parent of the other in a tree. Saturation builds all the transitions that can be obtained by composing these transitions:

Definition 7 *Let $\mathbb{P}$=$(\mathcal{F}, \mathcal{T})$ be a proof scheme. The* saturation *of $\mathcal{T}$ is the smallest set $\mathcal{T}'$ such that $\mathcal{T} \subseteq \mathcal{T}'$ and $\{((i_1, s_1, i_2, s_2), (\alpha_1, \alpha_2)), ((i_2, s_2, i_3, s_3), (\alpha_2, \alpha_3))\} \subseteq \mathcal{T}'$ implies $((i_1, s_1, i_3, s_3), (\alpha_1, \alpha_3)) \in \mathcal{T}'$.*

After saturating the transitions of $\mathbb{P}$, one may not keep a proof scheme as new transititions may label any pair of atoms, however if $\mathbb{P}$ is valid, the validity conditions remain satisfied.

We next define the notion of a *match* of a query in a proof scheme to formalize what it means for a query to be mapped into the part of the chase represented by a valid proof scheme.

Definition 8 *Let $\mathbb{P} = (\mathcal{F}, \mathcal{T})$ be a proof scheme with $\mathcal{F} = (V, E)$. We say that there is a* match *of q in $\mathbb{P}$ if there is a mapping π from* vars(q) *to* terms$(\mathbb{P})$ *such that:*

- *if α is a standard atom in q, then $\pi(\alpha) \in V$;*
- *if $\alpha = \Lambda(x, y)$ is a path atom in q with associated NFA $\mathbb{A}$, then there is a pair $((i_x, s, i_y, f), (\alpha_x, \alpha_y))$ in the saturation of $\mathcal{T}$ such that $\pi(x) = \alpha_x[i_x]$, $\pi(y) = \alpha_y[i_y]$ and s (resp. f) is an initial (resp. final) state of $\mathbb{A}$.*

Note that the problem of deciding if there is a match from given q in given $\mathbb{P}$ is NP-complete. The next proposition shows that CRPQ answering reduces to the existence of a match in a valid proof scheme.

Proposition 3 *$I, \mathcal{R} \models q$ if and only if there exists a match of q in some valid proof scheme.*

Proof Sketch: Suppose $I, \mathcal{R} \models q$, and let π' be a match of q in chase$(I, \mathcal{R})$. Let V_1 be the set of images of the standard atoms of q, as well as the atoms in which the image of the terms of q has been created. Let V_1' contain for each $\alpha \in V_1$ a node $\nu_\alpha \in \mathcal{CG}$ such that $\ell(\nu_\alpha) = \alpha$. Define $V_2' = V_1' \cup \{glb(\nu, \nu')\}$,

where the union is performed over all $(\nu, \nu') \in V_1' \times V_1'$ such that ν and ν' have the same root in $\mathcal{CG}$. Let $\mathcal{F}$ be the forest structure induced by $\mathcal{CG}$ over V_2'. We define V as $\{\ell(\nu) \mid \nu \in V_2'\}$, and E as the pairs $(\ell(\nu), \ell(\nu'))$ for any edge (ν, ν') in $\mathcal{F}$. We then define $\mathcal{T}$ as the transitions from α_1 to α_2 such that one is the child of the other, or both are roots, or $\alpha_1 = \alpha_2$. One can check that π' is a match of q in $\mathbb{P} = ((V, E), \mathcal{T})$. The main argument is as follows: by Prop. 2, the saturation of $\mathbb{P}$ contains all the transitions that hold between an atom and its descendants. As the *glb* of any two nodes with the same root also belongs to $\mathcal{F}$, the saturation of $\mathbb{P}$ also contains all the transitions that hold between any pair of atoms of $\mathbb{P}$. As the mapping of standard atoms is direct, π' is a match of q in $\mathbb{P}$. The converse direction follows from the definitions. □

Importantly, the size of the valid proof scheme built in the previous proof is polynomial in the size of q, so it is only necessary to consider polynomial-size proof schemes. It remains to show how to test whether a given proof scheme is valid. We will make use of the following lemma, which shows how transitions between an atom in the chase graph and an ancestor atom can be computed using the transitions relating its parent atom to the same ancestor, together with the transitions that can be obtained by considering the atom in isolation.

Lemma 1 *Suppose ν_2 is a descendant of ν_1 in $\mathcal{CG}(I, \mathcal{R})$, ν_3 is a child of ν_2, and $\alpha_i = \ell(\nu_i)$ for $1 \leq i \leq 3$. Then:*

1. $(i_1, s_1, i_3, s_3) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_1, \alpha_3)$ *iff there exist* $(i_1, s_1, i_2, s_2) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_1, \alpha_2)$ *and* $(i_2', s_2, i_3, s_3) \in \mathcal{T}_{q,\{\alpha_3\},\mathcal{R}}(\alpha_3, \alpha_3)$ *such that* $\alpha_2[i_2] = \alpha_3[i_2']$
2. $(i_1, s_1, i_3, s_3) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_3, \alpha_1)$ *iff there exist* $(i_1, s_1, i_2', s_2) \in \mathcal{T}_{q,\{\alpha_3\},\mathcal{R}}(\alpha_3, \alpha_3)$ *and* $(i_2, s_2, i_3, s_3) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_2, \alpha_1)$ *such that* $\alpha_2[i_2] = \alpha_3[i_2']$
3. $(i_1, s_1, i_4, s_4) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_3, \alpha_3)$ *iff there exist* $(i_1, s_1, i_2', s_2) \in \mathcal{T}_{q,\{\alpha_3\},\mathcal{R}}(\alpha_3, \alpha_3)$, $(i_2, s_2, i_3, s_3) \in \mathcal{T}_{q,I,\mathcal{R}}(\alpha_2, \alpha_2)$, *and MB: fixed typo:* $(i_3', s_3, i_4, s_4) \in \mathcal{T}_{q,\{\alpha_3\},\mathcal{R}}(\alpha_3, \alpha_3)$ *such that* $\alpha_3[i_2'] = \alpha_2[i_2]$ *and* $\alpha_3[i_3'] = \alpha_2[i_3]$

We now exploit this lemma and existing complexity results on RPQ answering to design a procedure for checking validity of a proof scheme, with the following complexity:

Proposition 4 *Checking the validity of a proof scheme can be done in single exponential time, in polynomial space if predicate arity is bounded, and in NL if q and $\mathcal{R}$ are fixed.*

Proof Sketch: First observe that if α, α' are root atoms of $\mathbb{P}$, we can compute $\mathcal{T}_{q,I,\mathcal{R}}(\alpha, \alpha')$ by calling an RPQ entailment oracle. We then work from the roots to the leaves, guessing for each edge (α_1, α_2) in $\mathbb{P}$, a sequence of rule applications that take α_1 to α_2, which ensures the satisfaction of the fourth validity condition. Importantly, we can bound the length of this sequence by noticing that after exponentially many rule applications, we will have generated two atoms β_1, β_2 that 'behave the same' (roughly: they are isomorphic w.r.t. *terms*(α_1) and have the same transitions w.r.t. α_1 and w.r.t. themselves). To check the fifth validity condition, we leverage Lemma 1 to inductively construct the set of transitions between α_1 and α_2, and between α_2 and itself. Note that we can directly compute $\mathcal{T}_{q,\{\alpha_2\},\mathcal{R}}(\alpha_2, \alpha_2)$ using RPQ entailment. The procedure we have sketched runs in non-deterministic polynomial space (constant space if q and $\mathcal{R}$ are fixed) with access to an RPQ entailment oracle. It then suffices to recall that RPQ entailment is in EXPTIME, in P for bounded-arity rules, and in NL if q and $\mathcal{R}$ are fixed. □

Since all polysize proof schemes can be enumerated in polynomial space, by Prop. 4 we obtain that the brute-force algorithm runs in EXPTIME in the general case, in PSPACE if the arity is bounded, and in NL if q and $\mathcal{R}$ are fixed (in which case the number of proof schemes and their size are constants, and checking the existence of a match is in constant time). Matching lower bounds are inherited from RPQ answering with linear rules [Bienvenu and Thomazo, 2016], or CRPQ answering in DL-Lite$_{\mathcal{R}}$ [Bienvenu *et al.*, 2015].

Theorem 1 *CRPQ entailment under linear rules is* EXPTIME*-complete in combined complexity,* PSPACE*-complete in combined complexity with bounded-predicate arity, and* NL*-complete in data complexity.*

5 Extension to Guarded Rules

We reduce CRPQ answering under guarded rules to CRPQ answering under linear rules. For simplicity, we assume that all the predicates in I also occur in $\mathcal{R}$ (however, the reduction is easily extended to drop this restriction). We first explain the ideas that underly the reduction. Prop. 1, which states that, under linear rules, facts can be chased independently, fails for guarded rules. However, a kind of independence can be recovered if instead of single fact we consider the (maximal) subset of the chase guarded by this fact, as expressed by Prop. 5, which also applies to non-necessarily ground atoms.

Proposition 5 *For any set of atoms F, it holds that* chase$(F, \mathcal{R})$ *is equivalent to* $\bigcup_{\alpha \in F}$ chase$(F^*_\alpha, \mathcal{R})$, *where* $F^*_\alpha =$ chase$(F, \mathcal{R})_{|\text{terms}(\alpha)}$.

By recursively using Prop. 5, one can devise the following alternative chase procedure. Let $F = I$; *(i)* Compute F^* the restriction of *chase*$(F, \mathcal{R})$ to *terms*(F); this can be done by calling an oracle for each candidate atom on *terms*(F) to decide if it is entailed by *chase*$(F, \mathcal{R})$; *(ii)* For each $\alpha \in F^*$ and G the maximal subset of F^* guarded by α (i.e., $G = F^*_\alpha$), for each rule $\rho \in \mathcal{R}$, let π be the homomorphism from $\text{body}(\rho)$ to G that maps the guard of ρ to α: *(ii-1)* apply ρ to G by π and let $G_{\rho,\pi}$ be the restriction of the generated set to the subset guarded by $\pi^s(\text{head}(\rho))$; *(ii-2)* Perform steps *(i)* and *(ii)* with $F = G_{\rho,\pi}$; *(iii)* The chase is the union of all the F^* built at Step *(i)*.

Now, let $(I, \mathcal{R})$ be the original KB, where $\mathcal{R}$ is a set of guarded rules, and $(I', \mathcal{R}')$ be the translated KB, where $\mathcal{R}'$ is a set of linear rules. Roughly, the reduction allows to simulate the alternative chase outlined above, with the translation being as follows: *(i)* we build I' from $(I, \mathcal{R})$ by replacing each $\alpha \in I$ by an atom that represents *chase*$(I, \mathcal{R})_{|\text{terms}(\alpha)}$ and, *(ii)* we build $\mathcal{R}'$ by replacing each rule $\rho \in \mathcal{R}$ by a set of linear rules, each representing an application of ρ to a distinct guarded subset of *chase*$(I, \mathcal{R})$. This construction will require new predicates associated with guarded sets of atoms. We will show that *chase*$(I, \mathcal{R})$ and *chase*$(I', \mathcal{R}')$ are homomorphically equivalent when restricted to the original predicates,

which implies that $\mathit{chase}(I', \mathcal{R}')$ can be used to answer CRPQs as these are closed under homomorphism.

To obtain a finite number of guarded sets (hence a finite set of new predicates, which will ensure the finiteness of the set of rules $\mathcal{R}'$), we have to rename them in a canonical way. To that end, we consider a new set of canonical variables $\mathcal{X} = \{T_1, \ldots, T_{2w}\}$, where w is the maximal predicate arity ($2w$ will be the maximal number of terms in a rule from $\mathcal{R}'$).

Definition 9 *Let α be an atom which may contain both classical terms and canonical variables. The* canonical renaming *$\Phi_{\alpha,\mathcal{Y}}$ of α w.r.t $\mathcal{Y} \subseteq \mathcal{X}$ is a substitution from $\mathsf{terms}(\alpha)$ to $\mathcal{X}$ such that $\Phi_{\alpha,\mathcal{Y}}(t_i) = t_i$ if $t_i \in \mathcal{X}$, otherwise $\Phi_{\alpha,\mathcal{Y}}(t_i) = T_j$ where j is the smallest integer such that $T_j \notin \mathcal{Y}$, T_j does not occur in α, and $T_j \neq \Phi_{\alpha,\mathcal{Y}}(t_k)$ for all $k < i$. If $\mathcal{Y} = \emptyset$, we simply write Φ_α.*

We denote by $\mathcal{G}$ the set of guarded atoms over canonical variables. Note that $\mathcal{G}$ is finite. For any guarded set (G, α), it holds that $(\Phi_\alpha(G), \Phi_\alpha(\alpha)) \in \mathcal{G}$. To each guarded set (G, α) in $\mathit{chase}(I, \mathcal{R})$ can be assigned a *complex predicate*, of the form $p_{\Phi_\alpha(G)}$ with the same arity as α. As $\mathcal{G}$ is finite, so is the set of complex predicates. The *canonical atom* associated with a guarded set (G, α) is $\mathit{can}(G, \alpha) = p_{\Phi_\alpha(G)}(\mathsf{terms}(\alpha))$.

Definition 10 *The* guarded translation *of I w.r.t. $\mathcal{R}$ is $I' = \{\mathit{can}(G(\alpha, I, \mathcal{R})) \mid \alpha \in I\}$, where $G(\alpha, I, \mathcal{R})$ denotes the guarded set $(\mathit{chase}(I, \mathcal{R})_{|\mathsf{terms}(\alpha)}, \alpha)$.*

Example 2 *Consider Ex. 1, where $I_e = \{\alpha_1, \alpha_2\}$, with $\alpha_1 = b(a_1, a_2, a_3, a_4)$ and $\alpha_2 = t(a_1)$. In the following, we underline the guard in a guarded set. $G(\alpha_2, I_e, \mathcal{R}_e) = \{\underline{t(a_1)}\}$, hence $\mathit{can}(G(\alpha_2, I_e, \mathcal{R}_e)) = p_{\{t(T_1)\}}(a_1)$. $G(\alpha_1, I_e, \mathcal{R}_e) = \{\underline{b(a_1, a_2, a_3, a_4)}, r(a_4, a_1), r(a_2, a_1), s(a_3, a_2), n(a_3, a_4), r(a_4, a_3)\}$, hence $\mathit{can}(G(\alpha_1, I_e, \mathcal{R}_e)) = p_{G'}(a_1, a_2, a_3, a_4)$, where G' is obtained by a canonical renaming of $G(\alpha_1, I_e, \mathcal{R}_e)$ (here, each a_i is simply replaced by T_i).*

The set $\mathcal{R}'$ is composed of two kinds of linear rules: rules that rebuild atoms on the original vocabulary (*reconstruction rules*) and rules that simulate the chase (*complex rules*).

Definition 11 *Let $(G, \alpha) \in \mathcal{G}$. The set of* reconstruction rules *associated with (G, α) contains for each $\beta \in G$ the (range-restricted) rule of the form $\mathit{can}(G, \alpha) \to \beta$.*

Definition 12 *Let $(G, \alpha) \in \mathcal{G}$ and $\rho \in \mathcal{R}$ be applicable to G by π. Let $\alpha' = \pi^s(\mathrm{head}(\rho))$ and $\Phi_{\alpha',\mathsf{terms}(\alpha)}$ be the canonical renaming of α' w.r.t. $\mathsf{terms}(\alpha)$. The* complex rule *associated with G, ρ and π is: $\mathit{can}(G, \alpha) \to \mathit{can}(G', \Phi_{\alpha',\mathsf{terms}(\alpha)}(\alpha'))$ where $G' = \Phi_{\alpha',\mathsf{terms}(\alpha)}(\mathit{chase}(G \cup \{\alpha'\}, \mathcal{R})_{|\mathsf{terms}(\alpha')})$.*

Example 3 *Consider $\mathcal{R}_e$ from Ex. 1 and $G = \{\underline{b(T_1, T_2, T_3, T_4)}, p(T_1)\}$ a guarded set in $\mathcal{G}$. The rule $b(x, y, z, t) \to h(y, z, t, u)$ is applicable to G through $\pi = \{x \mapsto T_1, y \mapsto T_2, z \mapsto T_3, t \mapsto T_4\}$. Hence, we build the complex rule $p_{\{b(T_1,T_2,T_3,T_4),p(T_1)\}}(T_1, T_2, T_3, T_4) \to p_{G'}(T_2, T_3, T_4, T_5)$, where $G' = \{\underline{h(T_2, T_3, T_4, T_5)}, n(T_3, T_4), n(T_4, T_5), s(T_3, T_2), r(T_4, T_3), \underline{s(T_5, T_4)}\}$.*

We finally obtain $\mathcal{R}' = \mathcal{R}_r \cup \mathcal{R}_c$, where $\mathcal{R}_r$ (resp $\mathcal{R}_c$) is the set of all reconstruction (resp. complex) rules. To prove the correctness of the reduction, we rely on the notion of the *expansion* of an atom.

Definition 13 *The* expansion *of a complex atom α is defined as $\mathrm{expansion}(\alpha) = \mathit{chase}(\alpha, \mathcal{R}_r) \setminus \alpha$. The expansion of a set of complex atoms is the union of the expansions of its atoms.*

Observe that the expansion of $\mathit{chase}(I', \mathcal{R}')$ is equal to its restriction to atoms with predicates in the original vocabulary.

We thus focus on the correspondence between $\mathcal{R}_c$ and $\mathcal{R}$.

Lemma 2 *Let $(G, \alpha = r(\mathbf{t}))$ be a guarded set of atoms. The following two statements hold:*

- *let $\rho \in \mathcal{R}$ be applicable to G by π, and let $F = G \cup \pi^s(\mathrm{head}(\rho))$. There exists a complex rule ρ' applicable to $p_{\Phi_\alpha(G)}(\boldsymbol{t})$ by π' such that: $\mathrm{expansion}(p_{\Phi_\alpha(G)}(\boldsymbol{t}) \cup \pi'^s(\mathrm{head}(\rho'))) \models F$.*
- *let $\rho' \in \mathcal{R}_c$ be applicable to $G' = p_{\Phi_\alpha(G)}(\boldsymbol{t})$ by π', and let $F' = G' \cup \pi'^s(\mathrm{head}(\rho'))$. There exists $\rho \in \mathcal{R}$ applicable to G, generating F, such that: $\mathit{chase}(G, \mathcal{R})_{|\mathsf{terms}(F)} \models \mathrm{expansion}(F')$.*

Lemma 2 allows us to show that $\mathrm{expansion}(\mathit{chase}(I', \mathcal{R}_c)) \equiv \mathit{chase}(I, \mathcal{R})$.

Proposition 6 *Given $(I, \mathcal{R})$, I' and $\mathcal{R}'$ as defined above can be computed in* 2ExpTime *in combined complexity,* ExpTime *when the arity is bounded, and in* PTime *in data complexity. The number of types whose predicate appears in $\mathcal{R}'$ is at most a double exponential in I and $\mathcal{R}$ (exponential when the arity is bounded, and constant* w.r.t. *the data).*

The next theorem follows from the provided reduction and our results for linear rules. Let us remark that the reduction to linear rules not being polynomial, we do not directly obtain the desired upper bounds. However, by carefully analyzing the algorithm for linear rules, we can show that it actually runs polynomially in the number of types.

Theorem 2 *CRPQ entailment under guarded rules is* 2ExpTime*-complete in combined complexity,* ExpTime*-complete in combined complexity with bounded-predicate arity, and* PTime*-complete in data complexity.*

6 Conclusion

We have made an important step in understanding the complexity of answering navigational queries over existential rules, by determining the precise complexity of answering CRPQs under linear and guarded rules, two central subclasses of existential rules. Our results are quite positive, as we showed that answering CRPQs under guarded rules is not harder than answering the more well-known CQs, under both combined complexity (with or without arity restrictions) and data complexity. For linear rules, we proved that moving from RPQs to CRPQs does not incur any additional computational cost, and for bounded-arity rules, we have the same complexity as for DL-Lite$_R$. We believe that the simulation of guarded rules by linear rules is of independent interest.

Acknowledgements

We thank Swan Rocher for fruitful discussions. This work was supported by the French ANR project PAGODA (ANR-12-JS02-0007).

References

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web*, pages 218–307, 2015.

[Bienvenu and Thomazo, 2016] Meghyn Bienvenu and Michaël Thomazo. On the complexity of evaluating regular path queries over linear existential rules. In *Web Reasoning and Rule Systems - 10th International Conference, RR 2016, Aberdeen, UK, September 9-11, 2016, Proceedings*, pages 1–17, 2016.

[Bienvenu *et al.*, 2015] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Simkus. Regular path queries in lightweight description logics: Complexity and algorithms. *J. Artif. Intell. Res. (JAIR)*, 53:315–374, 2015.

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

[Calì *et al.*, 2009a] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. Datalog extensions for tractable query answering over ontologies. In *Semantic Web Information Management - A Model-Based Perspective*, pages 249–279. 2009.

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

[Calvanese *et al.*, 2007] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. of AAAI*, pages 391–396, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI*, pages 714–720, 2009.

[Calvanese *et al.*, 2014] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.*, 237:12–55, 2014.

[Florescu *et al.*, 1998] Daniela Florescu, Alon Levy, and Dan Suciu. Query containment for conjunctive queries with regular expressions. In *Proc. of PODS*, 1998.

[Gottlob *et al.*, 2015] Georg Gottlob, Marco Manna, and Andreas Pieris. Polynomial rewritings for linear existential rules. In Q. Yang and M. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2992–2998. AAAI Press, 2015.

[Kontchakov *et al.*, 2013] Roman Kontchakov, Mariano Rodriguez-Muro, and Michael Zakharyaschev. Ontology-based data access with databases: A short course. In *Reasoning Web*, pages 194–229, 2013.

[Mugnier and Thomazo, 2014] Marie-Laure Mugnier and Michaël Thomazo. An introduction to ontology-based query answering with existential rules. In *Reasoning Web*, pages 245–278, 2014.

[Ortiz and Simkus, 2012] Magdalena Ortiz and Mantas Simkus. Reasoning and query answering in description logics. In *Reasoning Web*, pages 1–53, 2012.

[Ortiz *et al.*, 2011] Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proc. of IJCAI*, 2011.

[Rudolph and Krötzsch, 2013] Sebastian Rudolph and Markus Krötzsch. Flag & check: data access with monadically defined queries. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 151–162, 2013.

[Stefanoni *et al.*, 2014] Giorgio Stefanoni, Boris Motik, Markus Krötzsch, and Sebastian Rudolph. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *J. of Art. Intell. Res. (JAIR)*, 51:645–705, 2014.