



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Polynomial Combined Rewritings for Linear Existential Rules and DL-Lite with n-ary Relations

Citation for published version:

Gottlob, G, Manna, M & Pieris, A 2015, Polynomial Combined Rewritings for Linear Existential Rules and DL-Lite with n-ary Relations. in *Proceedings of the 28th International Workshop on Description Logics, Athens, Greece, June 7-10, 2015*. CEUR Workshop Proceedings (CEUR-WS.org). <<http://ceur-ws.org/Vol-1350/paper-22.pdf>>

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Publisher's PDF, also known as Version of record

Published In:

Proceedings of the 28th International Workshop on Description Logics, Athens, Greece, June 7-10, 2015.

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Polynomial Combined Rewritings for Linear Existential Rules and DL-Lite with n -ary Relations

Georg Gottlob¹, Marco Manna², and Andreas Pieris³

¹ Department of Computer Science, University of Oxford, UK
georg.gottlob@cs.ox.ac.uk

² Department of Mathematics and Computer Science, University of Calabria, Italy
manna@mat.unical.it

³ Institute of Information Systems, Vienna University of Technology, Austria
pieris@dbai.tuwien.ac.at

1 Introduction

This paper considers the setting of ontology-based query answering (OBQA). In this setting, Description Logics (DLs) and existential rules (a.k.a. tuple-generating dependencies or Datalog[±] rules) are popular ontology languages, while conjunctive queries (CQs) is the main querying tool. Among KR researchers there is a clear consensus that the required level of scalability in OBQA can only be achieved via query rewriting, which attempts to reduce OBQA into the problem of evaluating a query over a relational database. Two query languages are usually considered: *first-order queries* (FO) and *non-recursive Datalog queries* (NDL).

An interesting approach to query rewriting is the *polynomial combined approach* [7], which can be described as follows: an ontology Σ can be incorporated together with a CQ q into a database query q_Σ in polynomial time, and also with a database D into a database D_Σ in polynomial time, such that q_Σ over D_Σ yields the same result as q evaluated over the knowledge base consisting of D and Σ . The polynomial combined approach has been applied to \mathcal{ELH}_\perp^{dr} [7], an extension of the well-known DL \mathcal{EL} , to DL-Lite^N_{horn} [5, 6], one of the most expressive logics of the DL-Lite family, and only recently to the main guarded- and sticky-based classes of existential rules [3].

Research Challenges. The problem of applying the polynomial combined approach to existing DLs and classes of existential rules is relatively understood. Nevertheless, there are still basic open questions that deserve our attention. Regarding DLs, little is known about formalisms with n -ary relations such as DLR-Lite \mathcal{R} , that is, the extension of DL-Lite \mathcal{R} with n -ary roles. Regarding existential rules, it is open whether the polynomial combined approach can be applied to the class of *linear existential rules* (or simply *linear rules*), that is, assertions of the form $\forall \mathbf{X} \forall \mathbf{Y} (s(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z} p(\mathbf{X}, \mathbf{Z}))$, where $s(\mathbf{X}, \mathbf{Y})$ and $p(\mathbf{X}, \mathbf{Z})$ are atomic formulas [1].

It is not difficult to show that, if linear rules are polynomially combined rewritable, then also DLR-Lite \mathcal{R} is polynomially combined rewritable — this follows from the fact that query answering under DLR-Lite \mathcal{R} can be easily reduced to query answering under linear rules [1]. Thus, the key question that we need to answer, which has been explicitly stated as an open problem in [3], is the following:

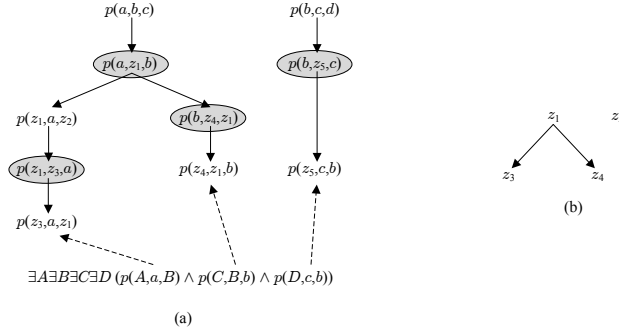


Fig. 1. Illustration of a proof generator.

Given a (Boolean) CQ q , a database D , and a set Σ of linear rules, can we rewrite in polynomial time: (i) q and Σ , independently of D , into a \mathcal{Q} -query q_Σ , where $\mathcal{Q} \in \{\text{FO}, \text{NDL}\}$, and (ii) D and Σ , independently of q , into a database D_Σ , such that $(D \cup \Sigma) \models q$ iff $D_\Sigma \models q_\Sigma$?

The answer to the above question is affirmative under the assumption that the arity of the underlying schema is bounded; implicit in [2]. However, little is known for arbitrary linear rules, without any assumption on the underlying schema. We give a positive answer even for linear rules that use predicates of unbounded arity. For more details, we refer the reader to [4].

2 Proof Generator

We assume the reader is familiar with the *chase procedure*. Recall that the chase for a database D and a set Σ of rules, denoted $\text{chase}(D, \Sigma)$, is a *universal model* of D and Σ , and thus $(D \cup \Sigma) \models q$ iff $\text{chase}(D, \Sigma) \models q$, for each CQ q . The instance $\text{chase}(D, \Sigma)$ can be naturally seen as a directed labeled graph, called *chase graph*, denoted $CG(D, \Sigma)$. It is also easy to verify that for linear rules, $CG(D, \Sigma)$ is a directed forest; for details on the chase, see, e.g., [1]. Our main technical tool is called proof generator, and it formalizes the intuitive idea that (Boolean) CQ answering under linear rules can be realized as a reachability problem on the chase graph. Let us illustrate the key ideas underlying the proof generator via a simple example.

Example 1. Let $D = \{p(a, b, c), p(b, c, d)\}$, and let Σ be the set of linear rules (for brevity, the universal quantifiers are omitted):

$$\begin{aligned} p(X, Y, Z) &\rightarrow \exists W p(X, W, Y) & p(X, Y, Z) &\rightarrow \exists W p(Z, W, Y) \\ p(X, Y, Z) &\rightarrow \exists W p(Y, X, W) & p(X, Y, Z) &\rightarrow p(Y, Z, X). \end{aligned}$$

Given the BCQ $q = \exists A \exists B \exists C \exists D (p(A, a, B) \wedge p(C, B, b) \wedge p(D, c, b))$, as shown in Figure 1(a), there exists a homomorphism h (dashed arrows in the figure) that maps q to an initial segment of the chase graph of D and Σ , and thus $(D \cup \Sigma) \models q$. It is interesting to observe that the nulls occurring in $h(q)$, i.e., z_1, z_3, z_4 and z_5 , form a rooted forest

F , depicted in Figure 1(b), with the following properties; for brevity, let $\nu(z)$ be the atom in $CG(D, \Sigma)$ where the null z is invented (see shaded atoms in Figure 1(a) for $\nu(z_1), \nu(z_3), \nu(z_4)$ and $\nu(z_5)$): (i) for every root node z , $\nu(z)$ is reachable from D ; (ii) for every edge (z, w) , $\nu(w)$ is reachable from $\nu(z)$; and (iii) for every atom $\underline{a} \in h(q)$, there exists a unique path π in F that contains all the nulls in \underline{a} , and, assuming that the leaf node of π is z , \underline{a} is reachable from $\nu(z)$. Indeed, it is easy to verify that $\nu(z_1)$ and $\nu(z_5)$ are reachable from D , $\nu(z_3)$ and $\nu(z_4)$ are reachable from $\nu(z_1)$, and finally, $h(p(A, a, B)) = p(z_3, a, z_1)$ is reachable from $\nu(z_3)$, $h(p(C, B, b)) = p(z_4, z_1, b)$ is reachable from $\nu(z_4)$, and $h(p(D, c, b)) = p(z_5, c, b)$ is reachable from $\nu(z_5)$. ■

Roughly speaking, the triple consisting of: (1) the homomorphism h , that maps q to the chase; (2) the function ν , that gives the atoms in the chase where the nulls occurring in $h(q)$ were invented; and (3) the forest F , that encodes how the nulls of $h(q)$ depend on each other, as well as the order of their generation, is what we call a proof generator for q w.r.t. D and Σ . The existence of such a triple, allows us to generate the part of the chase due to which the query is entailed, i.e., the proof of the query (hence the name “proof generator”). Therefore, for query answering purposes under linear rules, we simply need to check if such a proof generator exists.

Lemma 1. $(D \cup \Sigma) \models q$ iff there exists a proof generator for q w.r.t. D and Σ .

3 The Combined Rewriting

We give a positive answer to our research question regarding linear rules and the polynomial combined approach. More precisely, we show that:

Theorem 1. *The class of linear rules is polynomially combined \mathcal{Q} -rewritable, where $\mathcal{Q} \in \{\text{FO}, \text{NDL}\}$.*

To establish the above theorem, we heavily rely on the notion of the proof generator. Fix a (Boolean) CQ q , a database D , and a set Σ of linear rules. By Lemma 1, it suffices to construct in polynomial time a query $q_\Sigma \in \mathcal{Q}$ (independently of D), and a database D_Σ (independently of q), such that $D_\Sigma \models q_\Sigma$ iff a proof generator for q w.r.t. D and Σ exists. Roughly, the query q_Σ guesses a triple (h, ν, F) (as described in Example 1), and then verifies that the guessed triple is a proof generator for q w.r.t. D and Σ . The interesting part of q_Σ is the component that applies the crucial reachability checks required by the definition of the proof generator. Although the path among two atoms in the chase graph may be of exponential size, its existence can be checked via \mathcal{Q} -queries of polynomial size. An immediate consequence of Theorem 1 is that:

Corollary 1. *The description logic $\text{DLR-Lite}_{\mathcal{R}}$ is polynomially combined \mathcal{Q} -rewritable, where $\mathcal{Q} \in \{\text{FO}, \text{NDL}\}$.*

The results of this work are, for the moment, of theoretical nature and we do not claim that they will directly lead to better practical algorithms. We believe that a smart implementation of the proposed techniques can lead to more efficient rewriting procedures; this will be the subject of future research. We are also planning to optimize the proposed rewriting algorithm, with the aim of constructing queries of optimal size.

Acknowledgements. G. Gottlob was supported by the EPSRC Programme Grant EP/M025268/ “VADA: Value Added Data Systems – Principles and Architecture”. M. Manna was supported by the MIUR project “SI-LAB BA2KNOW – Business Analytics to Know”, and by Regione Calabria, programme POR Calabria FESR 2007-2013, projects “ITravel PLUS” and “KnowRex: Un sistema per il riconoscimento e l’estrazione di conoscenza”. A. Pieris was supported by the Austrian Science Fund (FWF): P25207-N23 and Y698.

References

1. Cali, A., Gottlob, G., Lukasiewicz, T.: A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14, 57–83 (2012)
2. Gottlob, G., Kikot, S., Kontchakov, R., Podolskii, V.V., Schwentick, T., Zakharyashev, M.: The price of query rewriting in ontology-based data access. *Artif. Intell.* 213, 42–59 (2014)
3. Gottlob, G., Manna, M., Pieris, A.: Polynomial combined rewritings for existential rules. In: *KR* (2014)
4. Gottlob, G., Manna, M., Pieris, A.: Polynomial rewritings for linear existential rules. In: *IJCAI* (2015)
5. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyashev, M.: The combined approach to query answering in DL-Lite. In: *KR* (2010)
6. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyashev, M.: The combined approach to ontology-based data access. In: *IJCAI*. pp. 2656–2661 (2011)
7. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic \mathcal{EL} using a relational database system. In: *IJCAI*. pp. 2070–2075 (2009)