Query Answering over Ontologies specified via Database Dependencies

Cristina Civili
Supervised by Riccardo Rosati
DIAG – Sapienza Università di Roma
civili@dis.uniroma1.it
Expected graduation date: December 2014

ABSTRACT

In this work we present a novel graph-based approach for studying the tractability of query answering over ontologies expressed by means of tuple-generating dependencies (TGDs). We do this by defining a new class of TGDs that subsumes all the other known classes that enjoy a particularly desirable property called first-order rewritability of query answering.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Relational databases, Rule-based databases, Query processing*

General Terms

Languages, Theory

Keywords

ontologies, OBDA, TGDs, existential rules, databases, tractability, query answering

1. INTRODUCTION

Ontology-based Data Access (OBDA) is a well-known expression in the knowledge representation community that refers to an innovative approach to access large and possibly distributed data sources through an ontology.
In the last ten years, ontologies catalyzed significant research efforts, becoming the standard tool for the formalization of knowledge bases, whose use in Semantic Web [4] and Information Integration is well-established. An ontology is a conceptualization of a domain of interest in terms of classes of objects and relations occurring between them. Formally, an ontology is a logic theory expressed in some chosen formalism.
In OBDA systems, ontologies are used as an additional layer of information placed upon traditional relational databases with the purpose of enriching them with semantic features, such as automated reasoning. Typically, the ontology contains only the intensional part of the knowledge base, while the DBMS is used to manage the extensional part, i.e., the actual data. Often, an additional layer of information between the ontology and the data sources is needed as a way of relating the two layers through mapping assertions [14].
Query answering in such systems does not consist in simply evaluating a query against a DBMS under the closed world assumption (CWA), but it is a reasoning task that needs the query to be answered against the whole logical theory under the open world assumption (OWA). This requires different approaches to query answering, based on expansion techniques already used in data integration and data exchange, such as materialization or query rewriting [7, 9], which is the one explored in this work.
To pave their way to real world scenarios, these systems must guarantee the same complexity of query answering of the systems currently used. Hence, the current research challenge is to find formalisms for representing ontologies that, on the one hand, are powerful enough to satisfy the most common conceptual modeling needs and, on the other hand, keep the tractability of query answering. In this light, two are the most remarkable results in the literature: the DL-Lite family of languages [8], i.e., lightweight Description Logics designed with the purpose of reducing the data complexity of query answering, and the Datalog$^{\pm}$ family of languages [5, 6], i.e., formalisms whose syntax is based on variants of the Datalog language[1]. For DL-Lite, as well as for some formalisms of the Datalog$^{\pm}$ family, the complexity of query answering is AC^0, since these languages share one important feature: the first-order (FO) rewritability of query-answering. This property guarantees that a conjunctive query over an ontology can be rewritten as an equivalent SQL query over the original database, hence the complexity of query answering in OBDA systems using these languages matches the complexity of query evaluation in classical DBMSs.
Datalog was rightly considered inadequate for modeling purposes [13], mostly because of the lack of the so-called "value invention" property, i.e., the ability of generating new unknown individuals. The recent introduction of the Datalog$^{\pm}$ family, however, proved that extending this formalism with existential variables in the head of rules is sufficient to fill this gap, and allows for investigating the expressive power of languages that are, in general, much more expressive than DLs, allowing both n-ary predicates and complex forms of join. As a result, the last few years have seen a growing interest in a multiplicity of languages, referred to by differ-

SIGMOD'14 PhD Symposium, June 22, 2014, Snowbird, Utah, USA.
Copyright © 2014 ACM 978-1-4503-2924-8/14/06…$15.00.
http://dx.doi.org/10.1145/2602622.2602632.

ent names such as existential rules, ∀∃-rules, tuple generating dependencies (TGDs), Datalog$^\pm$ rules or Datalog$^\exists$ rules, that identify fragments of FO logic.

2. THE PROBLEM

Despite the rising interest in this topic, and the huge impact that could have on the actual spread of semantic tools, the current picture that tractable classes of TGDs offer is a plethora of different fragments of FO logic whose expressive power is, in most of the cases, incomparable.

Provided that deciding the FO-rewritability of a set of TGDs is, in general, an undecidable problem [3] and that all these known classes identifies sufficient conditions for FO-rewritability, we believe that much can be done towards the goal of defining a more expressive class that subsumes all the other and still keeps the FO-rewritability of query answering.

Our goal is thus to answer the two following questions:

Question 1 - Is there a way to unify the known FO-rewritable classes of TGDs by defining a more general and feasibly checkable sufficient condition?

Question 2 - In doing so, is it possible to push the boundaries even further by allowing to express patterns that are outside of the scope of the other classes, yet do not increase the complexity of query answering?

The answer turned out to be positive for both these questions, thanks to the introduction of a novel graph-based approach for studying the FO-rewritability of query answering over ontologies expressed by means of TGDs.

3. PRELIMINARIES

A *TGD* R is an expression $\beta_1, ..., \beta_n \to \alpha_1, ..., \alpha_m$, where $\alpha_1, ..., \alpha_m, \beta_1, \ldots, \beta_n$ are atoms ($n, m \geq 1$), i.e., expressions of the form $r(t_1, \ldots, t_k)$, where $r = \mathit{Rel}(\beta)$ is a relation symbol, $k = \mathit{Arity}(r)$ is a non-negative integer associated to r and every t_i is either a constant symbol or a variable symbol. We call the expressions $\alpha_1, ..., \alpha_m$ and $\beta_1, \ldots, \beta_n$, respectively, the *head* of R ($\mathit{head}(R)$) and the *body* of R ($\mathit{body}(R)$).

We call *distinguished variables of* R the variables occurring both in the head and in the body of R, *existential body variables* of R the variables that occur only in the body of R, and *existential head variables* of R the variables that occur only in the head of R.

A *conjunctive query (CQ)* is an existentially quantified conjunction of positive atoms (possibly with free variables) of the form $q(\mathbf{x})$:- $\alpha_1, \ldots, \alpha_n$, where $\alpha_1, \ldots, \alpha_n$ is a sequence of atoms, called the *body* of q, the variables $\mathbf{x}$ are the *distinguished variables* of q and every variable of $\mathbf{x}$ occurs at least once in the body of q; the non-distinguished variables occurring in the body of q are called *existential variables* of q (and, analogously to TGDs, the existential variables occurring in more than one atom of $\mathit{body}(q)$ are called *NLE-variables* of q). A *union of conjunctive queries (UCQ)* is a set of CQs of the same arity.

We define the semantics of TGDs through FO logic under the Unique Name Assumption, i.e., different constant symbols are interpreted as different domain elements in every interpretation.

Given a TGD $R : \beta_1, ..., \beta_n \to \alpha_1, ..., \alpha_m$ and a database B, we say that B *satisfies* R if the FO interpretation $\mathcal{I}^B$ (i.e., the FO interpretation isomorphic to B) satisfies the FO sentence $\forall \vec{x}.\beta_1 \wedge \ldots \wedge \beta_n \to \exists \vec{y}.\alpha_1 \wedge \ldots \wedge \alpha_n$, where $\vec{x}$ denotes all the variables occurring in the body of R and $\vec{y}$ denotes the existential head variables of R. Given a set P of TGDs and a database D over the signature of P, we say that a database B over the signature of P *satisfies* (P, D) if $B \supseteq D$ and B satisfies every TGD in P. Moreover, we denote by $\mathit{sem}(P, D)$ the set of all databases B over the signature of P such that B satisfies (P, D).

Let q be a FO query and let B be a database. We denote by $\mathit{ans}(q, B)$ the set of tuples of constants $\mathbf{c}$ such that $\mathcal{I}^B$ satisfies $q(\mathbf{c})$, where $q(\mathbf{c})$ is the FO sentence obtained from q by replacing its free variables with the constants $\mathbf{c}$.

Let P be a set of TGDs, let q be a UCQ and let D be a database. We define the *certain answers* to q over P and D, denoted by $\mathit{cert}(q, P, D)$, as the set of tuples of constants $\mathbf{c}$ such that $\mathbf{c} \in \bigcap_{B \in \mathit{sem}(P,D)} \mathit{ans}(q, B)$.

Finally, we introduce the notion of FO-rewritable set of TGDs.

DEFINITION 1. *Let P be a set of TGDs. We say that P is* FO-rewritable *if, for every UCQ q, there exists a FO query q' such that, for every database D, $\mathit{cert}(q, P, D) = \mathit{ans}(q', D)$.*

4. METHODOLOGY

The basic idea behind our approach is to use a graph to encode the structure of a set of TGDs as well as some properties of the set that are relevant for defining a sufficient condition for its FO-rewritability.

Both in its simplest version, called the *position graph*, and in its extended version, called the *P-node graph*, the structure is such that its edges reflect the structure of the TGDs: roughly, they connect the node representing the atom occurring in the head of a TGD R (or atoms unifiable with such atoms) with the nodes representing atoms occurring in the body of R.

In the position graph, the structure is such that nodes represent atoms in a very approximated way, i.e., by referring to generic positions inside such atoms, or to specific positions occupied by existential variables.

The P-node graph takes this approach further, by representing the atoms more precisely: each node of the graph represents (still in an approximated way) an atom and its "context", i.e., the set of all atoms resulting from the "application" of a TGD, including the atom itself.

In both these structures, every edge from an atom σ to an atom σ' represents the possible transformation of σ into σ' through a query rewriting step. Therefore, both graphs provide an approximate representation of all the possible transformations of *single atoms*, through resolution steps involving the TGDs of P, starting from the atoms occurring in the heads of the TGDs of P.

Then, we also define a labeling of the graph edges, in order to encode relevant aspects of the behavior of the corresponding query rewriting steps. In particular, we are interested in identifying four conditions related to a rewriting step that uses a TGD R: "splitting" an existential variable in two different atoms of the body of the TGD R (s-edge); "missing" a distinguished variable in an atom of the body of the TGD R (m-edge); "decreasing" the number of bounded arguments in an atom of the body of the TGD R (d-edge); and the generation of an atom that is "isolated" in the body of the TGD R (i-edge). Only the first two conditions are considered in the position graph, while all of them are needed in the P-node

graph to correctly characterize the general case.
As shown in the following sections, both graphs allow for the definition of a sophisticated acyclicity condition that distinguishes between dangerous and harmless cycles, and that is a sufficient condition for the FO-rewritability of a set of TGDs.

5. EARLIER WORK

In [10], we started tackling this problem under some preliminary restrictions on the form of the TGDs that make the problem significantly easier, while still keeping the goal of providing a more expressive class of TGDs. More precisely, we focused our attention on *simple TGDs*, i.e., TGDs where: (*i*) the presence of repeated variables in the atoms is not allowed, (*ii*) the presence of constants is not allowed, (*iii*) the head contains a single atom.
In this restricted setting, we provided a formal definition of the class of *Simply Weakly Recursive* (SWR) TGDs and we proved that this class is FO-rewritable, by defining an algorithm that is able to compute the FO rewriting of CQs over SWR TGDs and proving its termination over the class.
In the following, we present the main content of the paper, i.e. the definition of the position graph and the class of SWR TGDs.

Definition 2. ***(Position)*** *A* position σ *is either an expression of the form $r[i]$ or an expression of the form $r[\,]$, where r is a relation symbol (and is denoted by $Rel(\sigma)$), and i is an integer such that $1 \leq i \leq k$, where k is the arity of r.*

Definition 3. ***(R-compatibility)*** *Let R be a simple TGD of the form $\beta_1, \ldots, \beta_n \to \alpha$. Then: (i) given an position $r[\,]$, we say that α* is R-compatible with $r[\,]$ *if $Rel(\alpha) = r$; (ii) given a position $r[i]$, we say that α* is R-compatible with $r[i]$ *if $Rel(\alpha) = r$ and $\alpha[i]$ is a distinguished variable of R.*

Given an atom β and a variable x occurring in β in position i, we denote by $Pos(x, \beta)$ the position $r[i]$.
Based on the above notions, we are now ready to define the position graph.

Definition 4. ***(Position Graph)*** *Given a set P of simple TGDs, the* position graph of *P, denoted by $AG(P)$, is a triple $\langle V, E, L\rangle$ where V (the set of nodes of $AG(P)$) is a set of positions, E is a set of edges (pairs of nodes), and L is an edge labeling function $L : E \to 2^{\{m,s\}}$. $AG(P)$ is inductively defined as follows:*

- *for every simple TGD $R \in P$ of the form $\beta_1, \ldots, \beta_n \to \alpha$, $r[\,] \in V$ where $r = Rel(\alpha)$;*
- *if $\sigma \in V$, then for every simple TGD $R \in P$, if $\alpha = head(R)$ is R-compatible with σ:*
    1. *for every atom $\beta \in body(R)$:*
        - *(a) $\langle \sigma, s[\,]\rangle \in E$, where $s = Rel(\beta)$;*
        - *(b) for each existential body variable z of R occurring in β, $\langle \sigma, \sigma'\rangle \in E$, where $\sigma' = Pos(z, \beta)$;*
        - *(c) if σ is of the form $r[i]$, then $\langle \sigma, \sigma''\rangle \in E$, where $\sigma'' = Pos(y, \beta)$ and y is the variable occurring in α at position i;*
        - *(d) if there exists a distinguished variable of R which does not occur in β, then, for every edge e added to E at points (a), (b), (c), $m \in L(e)$;*
    2. *if there exists an existential body variable x of R occurring in at least two atoms of $body(R)$, then for every edge e added to E at point 1, $s \in L(e)$;*
    3. *if σ is of the form $r[i]$, y is the variable occurring in α at position i, and y occurs in at least two atoms of $body(R)$, then, for every edge e added to E at point 1, $s \in L(e)$.*

We call *m-edge* an edge e such that $m \in L(e)$, and call *s-edge* an edge e such that $s \in L(e)$.
We are now ready to define simply weakly recursive sets of TGDs.

Definition 5. ***(Simply Weakly Recursive TGDs)*** *A set P of TGDs is* Simply Weakly Recursive (SWR) *if: (i) P is a set of simple TGDs; (ii) in $AG(P)$ there exists no cycle that contains both an m-edge and an s-edge.*

Theorem 1. *Every set of SWR TGDs is FO-rewritable.*

Example 1. *Let P be the following set of TGDs:*

$$R_1 : s(y_1, y_2, y_3), t(y_4) \to r(y_1, y_3)$$
$$R_2 : v(y_1, y_2), q(y_2) \to s(y_1, y_3, y_2)$$
$$R_3 : r(y_1, y_2) \to v(y_1, y_2)$$

Since there are no s-edges in the position graph $AG(P)$, shown in Figure 1, it immediately follows that P is a set of SWR TGDs, thus it is FO-rewritable.

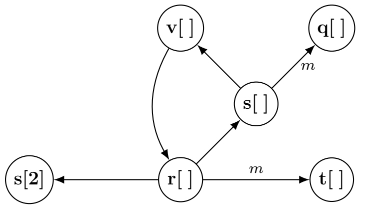

Figure 1: The position graph of P

■

It can be easily verified that the problem of establishing whether a set of TGDs is SWR is in PTIME. Moreover, we have shown in [10] that, under the restriction to simple TGDs, i.e., conditions (*i*), (*ii*), and (*iii*), SWR TGDs subsumes Linear TGDs, Multilinear TGDs, Sticky TGDs and Sticky-Join TGDs.

6. ONGOING RESEARCH

Our next goal is to weaken some of our preliminary restrictions in order to provide more general results. We are tackling this goal in two steps: the first one is to drop conditions (*i*) and (*ii*) and to keep (*iii*), and the second one is to deal with arbitrary TGDs, possibly containing multiple atoms in the head.
It turned out that, just by allowing for the presence of constants and repeated variables in the TGDs, the complexity of rewriting queries increases in such a way that the position graph is not sufficient to encode its behavior, as shown in the following motivating example.

Example 2. *Consider the set P of TGDs:*

$$R_1 = t(y_1, y_2), r(y_3, y_4) \to s(y_1, y_3, y_2)$$
$$R_2 = s(y_1, y_1, y_2) \to r(y_2, y_3)$$

P is not a set of simple TGDs, since $s(y_1, y_1, y_2)$ in $body(R_2)$ contains two occurrences of the same variable y_1. Let us try to use the notion of position graph nonetheless. The position graph of P is shown in Figure 2.

■

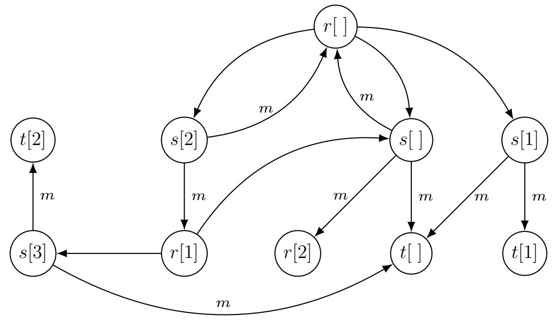

Figure 2: The position graph of P

According to our previous results, since there are no cycles in the position graph, P should be FO-rewritable. Unfortunately, by considering a simple boolean query such as $q() \leftarrow r(\text{“}a\text{”}, x)$ over P, it can be proved that this is not the case, due to the introduction of an unbounded number of existential join variables (a so-called *unbounded chain*) in the rewriting.

In Example 2, the position graph fails at correctly classifying the set of TGDs because it is unable to deal with multiple occurrences of a variable in the same atom. More precisely, the position graph allows for following the evolution of an existential variable introduced during the rewriting by tracing the positions of the atoms in which the variable appears, until it “splits”. However, since positions are encoded singularly (i.e. one in each node), there is no way to represent the fact that two positions contain the same existential variable, and thus to detect the splitting of such a variable.

In other words, dealing with multiple occurrences of the same variable inside an atom requires a more complex structure that is able to keep track of equalities between variables. Consider now another motivating example.

Example 3. *Let P be the set of TGDs:*

$$\begin{aligned}
R_1 &= r(y_1, y_2) \rightarrow t(y_3, y_1, y_1)\\
R_2 &= s(y_1, y_2, y_3) \rightarrow r(y_1, y_2)\\
R_3 &= u(y_1), t(y_1, y_1, y_2) \rightarrow s(y_1, y_1, y_2)
\end{aligned}$$

■

The set P of TGDs of Example 3 is not SWR, since, as before, we have multiple occurrences of the same variable inside one atom. Moreover, it is easy to verify that the set in neither Linear, since *body*(R_3) contains two atoms, nor multilinear, since $u(y_1)$ in R_3 does not contain the variable y_2, nor Sticky, since y_1 appears twice in the atom $t(y_1, y_1, y_2)$ of R_3, nor Sticky-Join, since y_1 appears in two different atoms of *body*(R_3).

The set P, however, is FO-rewritable. The intuition behind this fact is that the cyclic application of R_1, R_2, R_3 cannot ever occur in practice, due to restrictions on the applicability of such rules imposed by the occurrences of existential head variables and by the occurrences of repeated variables. Ideally, we aim our class to capture this kind of situations in which the recursion is only apparent, or “weak”, which also justifies the name weakly recursive.

Thus, in order to answer both *Question 1* and *Question 2* for the general case, we need to change the structure of the position graph. In particular, following the intuition of Examples 2 and 3, we need a more involved structure that is able to represent more accurately the cases in which the TGDs contain multiple occurrences of variables in the same atom. The first idea is to substitute positions with atoms as nodes of the graph; in such a way we can keep trace of the equalities between variables just by reusing the same variable symbol inside the atom. The second idea is to use a special variable symbol to mark the introduction of an existential variable in a step of the rewriting, and to use it in the same way in which positions of the form $r[i]$ were used in the position graph.

Moreover, we need to carefully address the two following problems: since we do not want to represent every atom of the rewriting (otherwise we would come up with a possibly infinite structure), we need a way to obtain an approximate representation. For this purpose, we rename the variables appearing in the atoms of the TGDs in such a way that we always use a finite number of symbols. This is formalized by the notion of P-atom.

Definition 6. ***(P-atom)*** *Given a set P of TGDs, a P-atom σ is an atom of the form $r(t_1, \ldots, t_n)$, where: r is a relation symbol of arity n occurring in P (and is denoted by $Rel(\sigma)$); every t_i is either a constant occurring in P or a variable symbol in the set $X_P = \{z, x_1, \ldots, x_k\}$, where k is the maximum arity of a relation occurring in P.*

Finally, since we want to define a graph that is a more precise approximation of the rewriting, we need to come up with a sharper definition of applicability of a rule. For this purpose, we pair P-atoms with their “context”, i.e., the set of atoms that appear together with such atoms as a result of the application of a TGD, and that determine whether its existential variables are bounded or not.

Definition 7. ***(P-node)*** *A P-node is a pair $\langle \sigma, \Sigma \rangle$, where Σ is a set of P-atoms and $\sigma \in \Sigma$.*

The graph, called P-atom graph, is then defined as a refinement of the position graph, in which: (i) nodes are P-nodes; (ii) the compatibility condition is much more involved and requires to check the context of a P-atom in order to establish whether such P-atom can unify with the head of a rule; edges can be of four types, namely: s-edges, m-edges, d-edges and i-edges. The two additional types of edges, i-edges and d-edges are related, respectively, to isolated components inside of a query, and to the notion of distinguished (i.e., bounded) arguments.

For space reasons, we do not give the detail of the definition of P-atom graph here. But we claim that, using a methodology analogous to the definition of the class of SWR TGDs, we can use such a structure to define a new class of TGDs, that we call Weakly Recursive (WR) [12].

Definition 8. ***(Weakly-recursive set of TGDs)*** *A set P of TGDs is* weakly recursive (WR) *if in the P-atom graph of P there exists no cycle that contains a d-edge, an m-edge, an s-edge, and does not contain any i-edge.*

Although such a class is still part of the ongoing research, we conjecture the following: (i) every set of WR TGDs is FO-rewritable; (ii) the decision problem of checking the membership of a set of TGDs to the class WR is in PSPACE; (iii) the class of WR TGDs strictly subsumes every other known class of FO-rewritable TGDs, including domain-restricted TGDs and acyclic graph of rule dependencies [2], which are incomparable with SWR TGDs.

Let us now consider again Example 2. If we use the P-node graph instead of the position graph, the dangerous cycle is detected and the set of TGDs is correctly classified as not

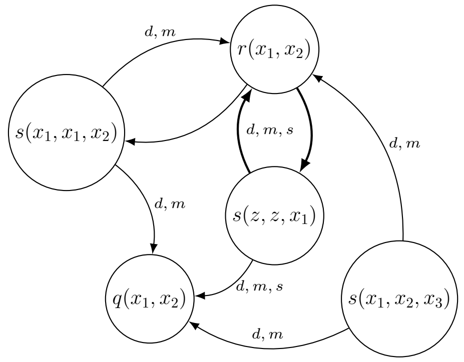

Figure 3: The *P*-node graph of P

being WR, as shown in Figure 3.
Finally, we started addressing one fundamental issue of our approach, concerning the expressive abilities of WR TGDs and, in particular, the ability of expressing modeling patterns that other known FO-rewritable ontology languages are unable to capture. Our preliminary results show that the class of WR TGDs allows for the identification of new FO-rewritable Description Logic languages.

7. FUTURE WORK

Up to now, we focused on identifying a class of FO-rewritable TGDs that is able to capture all the other classes that show the same computational complexity of query answering. We defined two classes that enjoy this property, namely SWR TGDs and WR TGDs, and we discovered that the complexity of deciding if a set of TGDs is WR rises from PTIME to PSPACE if we allow TGDs to have constants and multiple occurrences of the same variable inside an atom. These complexity results immediately suggest that this approach does not scale very well: depending on the size of the set of TGDs, we might be unable to determine in a reasonable amount of time if the set belongs to the class.
Now, if we assume to have an arbitrary set of TGDs P, we might end up in one of the following situations: (i) P is WR; (ii) we are unable to effectively establish if P is WR; (iii) P is not WR. The first case constitutes the positive scenario in which we know that we can use P as an additional layer on top of a relational database and still be able to do query answering in AC^0. The challenge is to find an answer about what to do if we end up in the second or the third situation. One future research direction is thus to explore this setting and to define approximation techniques that are able to tackle both the second and the third situations. Actually, a technique for addressing the third case, based on the concept of query patterns, was already presented in [11].

8. CONCLUSIONS

The main purpose of this work is to generalize the OBDA approach to classes of ontological languages that are more complex and more expressive than DLs. The introduction of these new languages is not just a theoretical exercise: richer ontology languages are actually needed, and these new formalisms could constitute an alternative to DLs for modeling ontologies in OBDA systems, and could be useful in all the cases in which the modeling capabilities of DLs have proved to be unsatisfactory. In this sense, our ultimate goal is to identify effective techniques for query answering through TGDs based on FO- rewritability and approximation techniques, with the more ambitious purpose of developing a working OBDA system that showcases our results.

9. ACKNOWLEDGMENTS

This research has been partially supported by the EU under FP7 project Optique - Scalable End-user Access to Big Data (grant n. FP7-318338).

10. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.

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

[3] C. Beeri and M. Y. Vardi. The implication problem for data dependencies. In *Automata, Languages and Programming*, pages 73–85. Springer, 1981.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

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

[6] A. Calì, G. Gottlob, and A. Pieris. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence*, 193:87–128, 2012.

[7] A. Calì, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 16–21, 2003.

[8] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. MASTRO-I: Efficient integration of relational data through DL ontologies. In *Proc. of the 20th Int. Workshop on Description Logic (DL 2007)*, volume 250 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, pages 195–202, 2007.

[9] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. What is query rewriting? In *Proc. of the 7th Int. Workshop on Knowledge Representation meets Databases (KRDB 2000)*, volume 29 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, pages 17–27, 2000.

[10] C. Civili and R. Rosati. A broad class of first-order rewritable tuple-generating dependencies. In *Proc. of the 2nd Datalog 2.0 Workshop*, 2012.

[11] C. Civili and R. Rosati. Query patterns for existential rules. In *Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR 2012)*, 2012.

[12] C. Civili and R. Rosati. Weakly recursive tgds. 2014. (Unpublished manuscript).

[13] P. F. Patel-Schneider and I. Horrocks. Position paper: a comparison of two modelling paradigms in the semantic web. In *Proc. of the 15th Int. World Wide Web Conf. (WWW 2006)*, pages 3–12, ‘2006.

[14] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.