Rewrite and Conquer: Dealing with Integrity Constraints in Data Integration

Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini

Abstract The work "Data Integration under Integrity Constraints", published at the CAiSE 2002 Conference, proposes a rewriting technique for answering queries in data integration systems, in the case where the global schema contains the classical key and foreign key constraints, and the mapping between the data sources and the global schema is of the global-as-view type. In this addendum, we explain why this research was important and how it gave raise to several results in the following years.

1 Introduction

Our work [3], republished in this volume, considers a data integration setting where a set of data sources is integrated into a global schema by means of *global-as-view (GAV)* mappings, and where the global schema contains integrity constraints. In data integration, the mapping establishes the relationship between the data at the sources and the elements of the global schema. While in the *local-as-view (LAV)* approach to mappings, every data source is described in terms of a view over the global schema, in the GAV approach the data sources are mapped to the global schema by a mechanism that associates to each element in the global schema a view over the data sources, with the meaning that every tuple satisfying the view at the sources, also

Andrea Calì
University of London, Birkbeck College and Oxford-Man Institute of Quantitative Finance, University of Oxford, e-mail: andrea@dcs.bbk.ac.uk

Diego Calvanese
Free University of Bozen-Bolzano, e-mail: calvanese@inf.unibz.it

Giuseppe De Giacomo
"Sapienza" Università di Roma, e-mail: degiacomo@dis.uniroma1.it

Maurizio Lenzerini
"Sapienza" Università di Roma, e-mail: lenzerini@dis.uniroma1.it

satifies the element of the gloabl schema in the virtual global database. On the other hand, the global schema provides a common representation for the domain of interest, and including integrity constraints is important if we aim at modeling the domain of interest with reasonable expressive power. Indeed, integrity constraints are the obvious means to express rules corresponding to semantic conditions characterizing the domain.

Referring to the setting described above, the paper addresses one of the most important problems in the design of a data integration system, namely, the definition of the method for computing the answer to queries posed in terms of the global schema. The integrity constraints considered in the paper are key and foreign key constraints, which are very popular mechanisms for adding semantics to a plain relational database schema. The challenge posed by the considered setting derives exactly from the presence of such integrity constraints. The query answering algorithm should compute the answer to queries by taking into account not only the data at the sources and the mapping from the data sources to the global schema, but also the facts that are implied by the constraints in the global schema.

The first contribution in the paper was to show that, when the global schema contains key and foreign key constraints, the semantics of the data integration system is best described in terms of a set of databases, rather than a single one, and this implies that query processing is intimately connected to the notion of querying incomplete databases. As a simple example, suppose that a foreign key constraint in the global schema asserts that every value of attribute A of relation r should appear in the (unary) key K of relation s, and assume that the data retrieved from the sources through the mappings do not satisfy this constraints, i.e., there is a value a in $r[A]$ that does not appear in position (attribute) K in any tuple of s. How do we interpret the semantics of the data integration system in this case? One option would be to consider the whole system incorrect, and not even trying to answer queries, which is obviously unacceptable. Another option is to interpret the absence of the tuple in s as a form of incompleteness, and consider as possible global databases every global database that has a new tuple $\mathbf{t}$ in s such that $\mathbf{t}[K] = a$. To account for incompleteness, given a query q, we should make sure that we answer q by computing the tuples that satisfy the query in every possible database, i.e, the so-called *certain answers* to q. This is exactly the approach adopted in the paper.

The second contribution of the paper was to propose a specific method to answer conjunctive queries posed to the global schema in the case of GAV mappings, and with key and foreign key constraints in the global schema. The method is based on a rewriting technique. A conjunctive query q is first rewritten into a union of conjunctive queries q' taking into account the integrity constraints in the global schema, and then q' is rewritten, taking into account the mapping, into a query q'' to be evaluated at the sources. The correctness of the method was proved by showing that the algorithm computes exactly the set of certain answers to the original query q. It is important to note that the query q'' is a first-order query over the data sources, and therefore the query answering algorithm runs in polynomial time (actually, in AC^0) with respect to data complexity, i.e., with respect to the size of data at the sources.

2 Historical Perspective

At the time of CAiSE 2002, the research on data integration was very active [17]. However, most of the contributions were based on the LAV approach, and only some of them considered the presence of integrity constraints in the global schema [17]. As for the GAV approach, there was the common belief that query answering is somehow trivial, because, at least in principle, a simple unfolding strategy suffices: substitute every atom α of the query with the source query associated to α by the mapping. Obviously, if the views over the sources are first-order queries, the query to be sent to the sources is also first-order, and the complexity of the whole process is AC^0 in data complexity. While this is true for the case of GAV without constraints, our work in [3] demonstrates that the presence of integrity constraints in the global schema, even of a very basic form, may change the picture considerably: key and foreign key constraints introduce a form of incomplete information in the system, and such incompleteness must be reasoned upon during query answering, which cannot be reduced to simple mapping-based unfolding.

A few months after CAiSE 2002, a seminal paper on data exchange, namely [11], was presented at ICDT 2003. *Data exchange* is a form of information integration where the emphasis is on transferring data from a source to a target database according to a set of mappings. Thus, differently from data integration, where the global database can be virtual, in data exchange the main task is to use the mappings to materialize a database starting from the source data. The work in [11] illustrates the importance of the *chase* for data exchange. The chase is a fixpoint algorithm enforcing implication of data dependencies over an incomplete database. Since mappings can be expressed as special dependencies, namely, *tuple-generating dependencies*, the chase turns out to be the right tool to exchange data from the source to the target. Now, if the target schema includes integrity constraints, depending on the form of such constraints, the chase may not terminate. For this reason, [11] introduced a specific form of target constraints, namely, *weakly-acyclic tuple-generating dependencies*, for which the chase terminates, so that a correct target database can be computed by chasing source data with respect to both the mappings and the integrity constraints in the target schema.

Unfortunately, this algorithm does not work in the case of key and foreign key constraints: indeed, in the presence of such constraints, and in particular when the foreign keys are cyclic, the chase does not terminate. We believe that one of the merits of [3] was to show an alternative way to treat integrity constraints in the global schema with respect to a chase-based algorithm, namely via *rewriting*. Our work also indicated that classes of practically relevant integrity constraints, that cannot be treated by the chase, can still be taken into account in data integration.

In the next two sections, we discuss two lines of research following the approach and the methodology presented in [3]. Section 3 describes the efforts to single out new classes of integrity constraints that can be dealt with by means of first-order rewriting algorithms. Section 4 reports on recent research work on incorporating the constraints considered in [3] in Datalog-like languages for expressing the global schema.

3 First-order Rewritability

One line of research deriving from our work [3] has been concerned with identifying classes of languages to express constraints over the global-schema that allow for computing certain answers with an approach based on rewriting. The key feature of this approach is that the constraints are taken into account by *rewriting*, independently of the underlying data, the given query q into a new query q_r that can be *expressed in first-order logic*. Hence q_r can be unfolded and evaluated by a standard relational database engine. This property of an integrity constraint language is what later became known as *first-order rewritability* of query answering [8], and it has been investigated intensively in the setting of ontology languages.

First-order rewritability imposes strict conditions on the expressive power of the underlying constraint/ontology language. It has led to the development of the *DL-Lite* family of lightweight ontology languages [8]. On the one hand, for the logics of this family conjunctive query answering is first-order rewritable. On the other hand, such logics are tightly connected to conceptual modeling formalisms, and can indeed capture the most important modeling features of UML class diagrams and Entity-Relationship diagramsas well as constructs that are part of the OWL standard [7, 1]. An exception are covering constraints, which require disjunction to be represented, and lead to query answering that is coNP-hard in data complexity, hence not first-order rewritable [9]. Interestingly, first-order rewritability does not impose any form of acyclicity on the set of constraints. Hence, the logics of the *DL-Lite* family depart from the constraint languages adopted in data-exchange to ensure finiteness of the chase (but see also Section 4).

First-order rewritability per se does not guarantee overall efficiency of query answering. In general, rewritings expressed as unions of conjunctive queries [8] may be very large (in the worst case exponential in the size of the original query), and hence not manageable by the DBMS engine. Experiments have shown that such a blowup typically occurs in real-world scenarios. This triggered the development of alternative rewriting techniques [18, 21, 14], whose focus has been on the reduction of the size of generated queries. These techniques produce rewritings that in many cases are polynomial, however the worst-case complexity is still exponential. The technique proposed in [12] produces worst-case polynomial rewritings at the cost of significantly complicating their structure, so that their execution is likely to suffer from poor performance [13]. An alternative, so called *combined*, approach has also been developed [15], in which the original data is first expanded with respect to the constraints/ontology (cf. Section 4), and then a rewritten query is executed over this expanded data. This allows for keeping the rewriting both small and efficiently executable, offering good performance at query time. However, it might not be applicable in those settings where no direct control over the data sources is granted, e.g., in data integration scenarios. Current research is investigating approaches in which a holistic view of the query answering/integration system is taken, that considers together with the constraints/ontology expressed at the global level, also the dependencies coming from the data sources and/or induced by the mappings, to optimize the overall query answering process [19, 20, 10].

4 Datalog-based Approach: Tractable Query Answering

Our work [3], which deals with “traditional” key and foreign key constraints, was the starting point of several studies on more general constraint languages. Datalog, in particular, has been used as a paradigmatic query language for over three decades, and can be naturally adopted in data integration. Datalog has some limits in modeling ontologies, which can be overcome with the intruduction of existential quantification in the rule heads; this way, rules become *tuple-generating dependencies (TGDs)*. Unfortunately, checking the entailment of a ground fact by a database (set of ground facts) and a set of TGDs is undecidableThe Datalog$^\pm$ family of languages [4] naturally extends [3] by proposing several TGD-based languages based on restrictions on the form of TGD bodies, so as to ensure decidability of query answering, and in some case tractability in data complexity. The two main decidability paradigms in Datalog$^\pm$ are *guardedness* and *stickiness*, which we briefly discuss below. Notice that, following the approach of our paper [3], none of the Datalog$^\pm$ languages guarantees the finiteness of the chase, which – we believe – is a necessary premise to ensure sufficient expressive power.

Guardedness is a well-known property of first-order theories that guarantees decidability. *Guarded TGDs* [4], or *guarded Datalog*$^\pm$, have been inspired by this notion, and offer PTIME data complexity of query answering. *Linear TGDs*, or *linear Datalog*$^\pm$, are a less expressive extension of the keys and foreign keys of [3], which enjoys better computational properties that guarded Datalog$^\pm$; in particular, linear Datalog$^\pm$ is first-order rewritable, with a technique analogous to that of [3]. Extension of guarded Datalog$^\pm$ include the addition of stratified negation, and a relaxation of guardedness that defines *weakly-guarded Datalog*$^\pm$ [6].

Stickiness is a completely different paradigm from guardedness, and it has been designed with the aim of devising a first-order rewritable Datalog$^\pm$ language. *Sticky sets of TGDs*, or *sticky Datalog*$^\pm$ [6], are defined by an easily testable syntactic condition, and are obviously first-order rewritable. Extension of sticky Datalog$^\pm$ are also studied in [6].

To achieve better expressive power, some works extend Datalog$^\pm$ with so-called negative constraints [6] and *Equality-Generating Dependencies* [5], the latter obviously extending the key constraints in our original work [3].

Datalog$^\pm$ languages have found several applications; without the restriction of chase termination, their expressive power allows for capturing several ontology languages. Interestingly, the work [16] unites the notions of chase termination and guardedness in a single language.

Other works propose semantic characterizations of sets of TGDs, with emphasis on rewriting. The work [2] defines the notion of *finite unification set*, that is, a set of TGDs that is first-order rewritable by means of a backward-chaining unification algorithm. Rewritability, introduced by us in ontology-based data access in [3], remains a crucial notion for reasons of efficiency of query answering.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research*, 36:1–69, 2009.
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. A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini. Data integration under integrity constraints. In *Proc. of CAiSE 2002*, volume 2348 of *LNCS*, pages 262–279. Springer, 2002.
4. A. Calì, G. Gottlob, T. Lukasiewicz, B. Marnette, and A. Pieris. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proc. of LICS 2010*, pages 228–242, 2010.
5. A. Calì, G. Gottlob, G. Orsi, and A. Pieris. On the interaction of existential rules and equality constraints in ontology querying. In *Correct Reasoning*, pages 117–133, 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. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati. Ontologies and databases: The *DL-Lite* approach. In S. Tessaris and E. Franconi, editors, *Semantic Technologies for Informations Systems – 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.
8. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. *Artificial Intelligence*, 2012. To appear.
10. F. Di Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo. Optimizing query rewriting in ontology-based data access. In *Proc. of EDBT 2013*, 2013.
11. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. In *Proc. of ICDT 2003*, pages 207–224, 2003.
12. G. Gottlob and T. Schwentick. Rewriting ontological queries into small nonrecursive Datalog programs. In *Proc. of KR 2012*, pages 254–263, 2012.
13. S. Kikot, R. Kontchakov, V. Podolskii, and M. Zakharyaschev. Long rewritings, short rewritings. In *Proc. of DL 2012*, volume 846 of *CEUR*, `ceur-ws.org`, 2012.
14. S. Kikot, R. Kontchakov, and M. Zakharyaschev. Conjunctive query answering with OWL 2 QL. In *Proc. of KR 2012*, pages 275–285, 2012.
15. R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to ontology-based data access. In *Proc. of IJCAI 2011*, pages 2656–2661, 2011.
16. M. Krötzsch and S. Rudolph. Extending decidable existential rules by joining acyclicity and guardedness. In *Proc. of IJCAI 2011*, pages 963–968, 2011.
17. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.
18. H. Pérez-Urbina, B. Motik, and I. Horrocks. Tractable query answering and rewriting under description logic constraints. *J. of Applied Logic*, 8(2):186–209, 2010.
19. M. Rodriguez-Muro and D. Calvanese. High performance query answering over *DL-Lite* ontologies. In *Proc. of KR 2012*, pages 308–318, 2012.
20. R. Rosati. Query rewriting under extensional constraints in *DL-Lite*. In *Proc. of DL 2012*, volume 846 of *CEUR*, `ceur-ws.org`, 2012.
21. R. Rosati and A. Almatelli. Improving query answering over *DL-Lite* ontologies. In *Proc. of KR 2010*, pages 290–300, 2010.