# Taming the Infinite Chase: Query Answering under Expressive Relational Constraints

Andrea Calì[2,1], Georg Gottlob[1,2], and Michael Kifer[3]

[1]Computing Laboratory *and*
[2]Oxford-Man Inst. of Quantitative Finance
University of Oxford
United Kingdom

[3] Dept. of Computer Science
SUNY Stony Brook
USA

{andrea.cali,georg.gottlob}@comlab.ox.ac.uk
kifer@cs.sunysb.edu

## Abstract

Answering queries posed over knowledge bases is a central problem in knowledge representation and database theory. In databases, query containment is one of the important query optimization and schema integration techniques [1, 12, 16]; in knowledge representation, it has been used for object classification, schema integration, service discovery, and more, in particular in the area of description logics [6, 14]. Results on practical instances of the general problem were studied in [12], followed by [5, 7, 2, 4, 13]. In particular, [5] and [7] deal respectively with query containment and efficient query answering under expressive description logic constraints, that can express several construct used in conceptual data modeling; [2] and [4] address query containment under constraints derived respectively from entity-relationship and object-oriented formalisms. The complexity of reasoning tasks on complex constraints based on answer set programs has been investgated in [19]. The problem of query containment is strictly related to that of answering queries over knowledge bases; indeed, the two are mutually reducible; we focus on the former, and our results immediately extend to the latter.

In our work, rather than focusing on specific logical theories, we analyze the fundamental difficulty that underlies earlier approaches, such as [12, 2, 4]. They all considered special classes of so-called *tuple-generating dependencies (TGDs)* and *equality-generating dependencies (EGDs)*, all used the technique called *chase*, and all faced the problem that the chase generates infinite relations, and query answering and containment are undecidable under general TGDs and EGDs. The chase [15, 12] is a procedure that "repairs" violations of TGDs and EGDs, until a fixed-point is reached; it has been used in many works in data exchange [9, 11, 17]; the chase is also a form of *tableau*, and it has been successfully applied in terminological reasoning based on description logics [7, 18]. We carve out a significantly larger classes of TGDs, with also the addition of EGDs. Notice that TGDs and EGDs are able to express most description logic constructs used in data modeling [7].

In particular, we first define the notions of sets of *guarded TGDs (GTGDs)* and of *weakly guarded TGDs (WGTGDs)*. A TGD is guarded if its body contains an atom called *guard* that covers all variables occurring in the body. Weakly guarded TGDs are a generalization of guarded TGDs that require guards to cover only variables occurring at *affected* positions, i.e., positions in predicates that may contain fresh labelled nulls generated during the chase. The notion of guard is crucial, since query evaluation becomes undecidable once we allow the presence of a single non-guarded TGD. Our main contribution lies in the complexity bound for query evaluation under WGTGDs and GTGDs. We show that the complexity of query evaluation (and, equivalently, of query containment) under WGTGDs is EXPTIME-hard, in case of a fixed set of TGDS, and 2-EXPTIME-hard in case the TGDs are part of the input.

As for upper bounds, let us first remark that we cannot (as one may think at the first glance) directly or easily use known results on guarded logics [10] to derive complexity results for query evaluation, since queries are in general non-guarded. We therefore develop new algorithms, and prove that query answering is EXPTIME-complete in case of bounded predicate arities, and even in case the set of WGTGDs is fixed, and is 2-EXPTIME complete in general. The proof of the upper bound is based on an alternating algorithm that mimicks the chase by using a finite number of configurations: each of them corresponds to what we call the *cloud* of one atom $\underline{a}$, i.e., the set of atoms in the chase whose arguments either appear in $\underline{a}$ or in the "active domain" of the input database instance.

Then, we derive complexity results for reasoning with GTGDs. While in the general case the complexity is the same as for WGTGDs, interestingly, when reasoning with a *fixed* set of dependencies (which is the usual setting in data exchange and in description logics), we get much better results: evaluating Boolean queries is NP-complete (same complexity of answering without constraints [8]), and in PTIME in case the query is atomic. Our results subsume the results of [12] on IDs alone as a special case.

Furthermore, we describe a semantic condition, called *Polynomial Clouds Criterion (PCC)*, imposing that the number of clouds it generates during a chase is polynomial in the size of the input database instance, and the cloud of each generated atom can be obtained in polynomial time from the cloud of the atom from which it was generated in the chase. Whenever a set of WGTDs fulfills the PCC, then answering Boolean queries is in NP, and answering atomic queries, as well as queries of bounded treewidth, is in PTIME.

Finally, we introduce EGDs together with WGTGDs: we define a class of *innocuous* EGDs, that have the property that they can be ignored in the query answering phase, since they do not actually interact with TGDs.

With the above results, we subsume both the main decidability and complexity result in [12], and decidability and complexity results on F-logic lite [13] as special cases, and we are actually way more general. We also show that F-logic Lite [4], a meaningful fragment of F-Logic [13], can be handled by our approach.

Details about the aforementioned results, including proofs, can be found in the technical report [3].

## References


1. A. Aho, Y. Sagiv, and J. D. Ullman. Equivalence of relational expressions. *SIAM J. of Computing*, 8(2):218–246, 1979.
2. Andrea Calì. Querying incomplete data with logic programs: ER strikes back. In *ER 2007*, pages 245–260, 2007.
3. Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: reasoning under expressive relational constraints. Unpublished technical report, `http://www.andreacali.com/CGK.pdf`, 2008.
4. Andrea Calì and Michael Kifer. Containment of conjunctive object meta-queries. In *VLDB 2006*, pages 942–952. VLDB Endowment, 2006.
5. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *PODS 1998*, pages 149–158, 1998.
6. D. Calvanese, G. De Giacomo, and M Lenzerini. Description logics for information integration. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski*, volume 2408 of *Lecture Notes in Computer Science*, pages 41–60. Springer, 2002.
7. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
8. A.K. Chandra and P.M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *STOC 1977*, pages 77–90, 1977.
9. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
10. M. Elisabeth Goncalves and Erich Grädel. Decidability issues for action guarded logics. In *Description Logics*, pages 123–132, 2000.
11. Georg Gottlob and Alan Nash. Data exchange: computing cores in polynomial time. In *PODS*, pages 40–49, 2006.
12. D.S. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *Journal of Computer and System Sciences*, 28:167–189, 1984.
13. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of ACM*, 42:741–843, July 1995.
14. L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *WWW 2003*, 2003.
15. David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *TODS*, 4(4):455–469, 1979.
16. T. Millstein, A. Levy, and M. Friedman. Query containment for data integration systems. In *PODS 2000*, pages 67–75, 2000.
17. Alan Nash, Alin Deutsch, and Jeff Remmel. Data exchange, data integration, and chase. Technical Report CS2006-0859, UCSD, April 2006.
18. Riccardo Rosati. On conjunctive query answering in EL. In *20th International Workshop on Description Logics (DL-2007)*. CEUR Electronic Workshop Proceedings, 2007.
19. Mantas Simkus and Thomas Eiter. DNC: Decidable non-monotonic disjunctive logic programs with function symbols. In *LPAR 2007*, pages 514–530, 2007.