Will My Favorite Chases Terminate if Evaluating Conjunctive Queries Does? One Does Not Simply Decide This (Extended Abstract)

Lucas Larroque[1], Quentin Manière[2]

[1]Inria, DI ENS, ENS, CNRS, PSL University, Paris, France
[2]LIRMM, Inria, University of Montpellier, CNRS, Montpellier, France

Abstract

This extended abstract summarizes our recent contribution [1], in which we ask: Within a class of existential rules that supports decidable query entailment, do the usual abstract classes become concrete? We answer in the negative for classes based upon the termination of all classical chase variants and for the bts class. This explains the necessity of diverse, class-specific techniques for proving chase termination, as query entailment offers no systematic tool for this purpose.

Keywords

Ontology-based query answering, Existential rules, Chase termination, Bounded treewidth sets

1. Introduction

An important task in knowledge representation and reasoning is ontology-based data access (OBDA). In OBDA, an ontology is used to enrich a database with domain knowledge, enabling the inference of new information that is not explicitly stored in the database. Alongside description logics (DLs), existential rules (also known as Datalog$^{\pm}$, or tuple-generating dependencies) are a prominent formalism to represent ontologies in OBDA [2]. A fundamental reasoning task in OBDA is the Boolean conjunctive query (BCQ) entailment problem, which consists in deciding whether a BCQ is entailed by a database and an ontology of interest. Unfortunately, with ontologies expressed as sets of existential rules, BCQ entailment is undecidable in general [3, 4]. To ensure decidable BCQ entailment, three main properties have been proposed.

First, the chase procedure [5] is a materialization-based algorithm that, given a database and a set of existential rules, iteratively applies the rules to the database to produce a (possibly infinite) universal model of the input. If finite, this universal model can then be queried using regular BCQ evaluation techniques to decide entailment. Many variants of the chase procedure exist, differing in the way rules are applied and redundancies are handled. In this work, we consider the oblivious [6], semi-oblivious (*a.k.a.* Skolem) [7], restricted (*a.k.a.* standard) [8], and core chase variants [9]. Guaranteed termination of the chase thus implies decidable BCQ entailment, but this property is also desirable in data exchange settings, where one is interested in translating data from a source schema to a target schema while preserving certain properties [10]. It also enables computing aggregates, which otherwise make little sense on infinite models [11], and ensures finite controllability, *i.e.* only considering finite models does not impact query answers.

Second, query rewriting techniques aim at rewriting the input BCQ into a first-order query that can be directly evaluated over the input database. Rule sets for which such a rewriting exists (and is computable) are called *fus*, for finite unification set [12]. Again, *fus* rule sets have benefits beyond decidability of BCQ entailment, as they allow answering the rewritten query over any database. In particular, this allows answering queries even without edit access, or answering queries efficiently

DL 2026: 39th International Workshop on Description Logics, July 17–19, 2026, Lisbon, Portugal
lucas.larroque@inria.fr (L. Larroque); quentin.maniere@inria.fr (Q. Manière)
https://www-sop.inria.fr/members/Quentin.Maniere/ (Q. Manière)
0009-0007-2351-2681 (L. Larroque); 0000-0001-9618-8359 (Q. Manière)

in streaming settings [13]. In the context of DLs, ontology-mediated queries formulated within the DL-Lite family are well-known to support such rewritings [14], and thus enjoy AC^0 data complexity.

Third, bounded treewidth sets (*bts*) [6] are classes of existential rules that ensure that, for any database, the universal model produced by the chase has bounded treewidth. This property renders BCQ entailment decidable by using Courcelle's theorem [15], which states that any property definable in monadic second-order logic can be decided in linear time over structures of bounded treewidth. DLs with the so-called tree-model (or forest-model) property exploits the same mechanism to warrant decidability, with techniques that are usually tailored to exhibit precise complexity upper bounds [16].

Deciding whether a given set of existential rules has any of these properties (*i.e.* whether it guarantees termination of a certain chase variant, is *fus*, or is *bts*) is, here again, undecidable in general. The classes of such sets of rules are thus called *abstract*, as opposed to *concrete* classes for which membership is decidable. Efforts have been devoted to proving that, when restricting to some concrete classes of existential rules, membership in the abstract classes presented above becomes decidable. This is especially the case for chase termination: oblivious and semi-oblivious chase termination have been proven decidable for guarded and for sticky rules [17, 18], thus covering several (positive fragments of) lightweight DLs, while core chase termination is decidable for guarded rules [19] but remains open for sticky rules. Restricted chase termination appears more difficult: its decidability has been proven for single-head guarded and sticky rules [20], but remains open in the multi-head case for both classes. The multi-head case for linear rules, a subclass of guarded rules, has been solved only recently [21].

2. Contribution

The approaches reported above vary greatly and often require sophisticated constructions, but they all share two properties: the considered concrete class always enjoys decidable BCQ entailment, and termination of the considered variant of the chase is always proved decidable. This raises the following question: do we get decidable chase termination (or *fus* membership? or *bts* membership?) for free when considering concrete classes of rules with decidable BCQ entailment? For *fus*, the answer is known to be negative, as shown by [22], who proved that boundedness of Datalog (which here coincides with *fus* membership) is undecidable. Surprisingly, for chase termination and *bts* membership, the question remains open. We answer it negatively in [1] by proving the following:

Theorem 1. *There exists a concrete class $\mathcal{C}$ of sets of existential rules such that:*

1. *Termination of the oblivious, semi-oblivious, restricted, and core chases, is undecidable in $\mathcal{C}$,*
2. *bts membership is undecidable in $\mathcal{C}$, and*
3. *BCQ entailment is decidable in $\mathcal{C}$.*

Our paper is then fully dedicated to proving Theorem 1 by explicitly constructing such a class $\mathcal{C}$, made of sets of rules that simulate Minsky machines. This simulation of Minsky machines is inspired from [23], a paper interested in termination of the oblivious, semi-oblivious and restricted chases. We modify their construction in non-trivial ways to also support the core chase and to introduce a flooding mechanism that is instrumental in achieving the properties regarding *bts* membership and BCQ entailment.

The resulting class exploits relations of arity up to 3 and thus does not fall within any common DL. Exhibiting a similar class within the realm of DLs, if it exists, would require expressive fragments, typically from the $\mathcal{S}$ or $\mathcal{Z}$ families, as otherwise *bts* membership often comes for free and is thus very much decidable.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

Acknowledgments

This work has been partially supported by ANR grant EXPAND (ANR-25-CE23-1215).

References

[1] L. Larroque, Q. Manière, Will my favorite chases terminate if evaluating conjunctive queries does? One does not simply decide this, in: Proceedings of the 35th International Joint Conference on Artificial Intelligence, IJCAI'26, 2026.

[2] R. Fagin, Tuple-generating dependencies, in: Encyclopedia of Database Systems, Second Edition, Springer, 2018. doi:10.1007/978-1-4614-8265-9_1274.

[3] C. Beeri, M. Y. Vardi, The implication problem for data dependencies, in: Proceedings of the 8th Colloquium on Automata, Languages and Programming, ICALP'81, Springer-Verlag, 1981, p. 73–85.

[4] A. Calì, G. Gottlob, T. Lukasiewicz, A general datalog-based framework for tractable query answering over ontologies, Journal of Web Semantics 14 (2012) 57–83. doi:https://doi.org/10.1016/j.websem.2012.03.001.

[5] D. Maier, A. O. Mendelzon, Y. Sagiv, Testing implications of data dependencies, ACM Trans. Database Syst. 4 (1979) 455–469. doi:10.1145/320107.320115.

[6] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, J. Artif. Intell. Res. 48 (2013) 115–174. doi:10.1613/JAIR.3873.

[7] B. Marnette, Generalized schema-mappings: from termination to tractability, in: Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'09, Association for Computing Machinery, 2009, p. 13–22. doi:10.1145/1559795.1559799.

[8] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: semantics and query answering, Theoretical Computer Science 336 (2005) 89–124. doi:https://doi.org/10.1016/j.tcs.2004.10.033, database Theory.

[9] A. Deutsch, A. Nash, J. Remmel, The chase revisited, in: Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'08, Association for Computing Machinery, 2008, p. 149–158. doi:10.1145/1376916.1376938.

[10] R. Fagin, P. G. Kolaitis, L. Popa, Data exchange: getting to the core, ACM Trans. Database Syst. 30 (2005) 174–210. doi:10.1145/1061318.1061323.

[11] F. Afrati, P. G. Kolaitis, Answering aggregate queries in data exchange, in: Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'08, Association for Computing Machinery, 2008, p. 129–138. doi:10.1145/1376916.1376936.

[12] J.-F. Baget, M. Leclère, M.-L. Mugnier, E. Salvat, Extending decidable cases for rules with existential variables, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09, Morgan Kaufmann Publishers Inc., 2009, p. 677–682.

[13] A. Ronca, M. Kaminski, B. C. Grau, B. Motik, I. Horrocks, Stream reasoning in temporal datalog, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI'18, AAAI Press, 2018, pp. 1941–1948. doi:10.1609/AAAI.V32I1.11537.

[14] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, J. Autom. Reason. 39 (2007) 385–429. doi:10.1007/S10817-007-9078-X.

[15] B. Courcelle, The monadic second-order logic of graphs. I. recognizable sets of finite graphs, Inf. Comput. 85 (1990) 12–75. doi:10.1016/0890-5401(90)90043-H.

[16] C. Lutz, The complexity of conjunctive query answering in expressive description logics, in: A. Armando, P. Baumgartner, G. Dowek (Eds.), Proceedings of the 4th International Joint Conference on Automated Reasoning, IJCAR'08, Lecture Notes in Computer Science, Springer, 2008, pp. 179–193. doi:10.1007/978-3-540-71070-7_16.

[17] M. Calautti, G. Gottlob, A. Pieris, Chase termination for guarded existential rules, in: Proceedings

of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS'15, Association for Computing Machinery, 2015, p. 91–103. doi:10.1145/2745754.2745773.

[18] M. Calautti, A. Pieris, Oblivious chase termination: The sticky case, in: Proceedings of the 22nd International Conference on Database Theory, ICDT'19, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, pp. 17:1–17:18. doi:10.4230/LIPIcs.ICDT.2019.17.

[19] A. Hernich, Computing universal models under guarded tgds, in: Proceedings of the 15th International Conference on Database Theory, ICDT'12, Association for Computing Machinery, 2012, p. 222–235. doi:10.1145/2274576.2274600.

[20] T. Gogacz, J. Marcinkowski, A. Pieris, All-instances restricted chase termination, in: Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS'20, Association for Computing Machinery, 2020, p. 245–258. doi:10.1145/3375395.3387644.

[21] L. Gerlach, L. Larroque, J. Marcinkowski, P. Ostropolski-Nalewaja, About the multi-head linear restricted chase termination, in: Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR'25, 2025, pp. 346–355. doi:10.24963/kr.2025/34.

[22] H. Gaifman, H. G. Mairson, Y. Sagiv, M. Y. Vardi, Undecidable optimization problems for database logic programs, J. ACM 40 (1993) 683–713. doi:10.1145/174130.174142.

[23] T. Gogacz, J. Marcinkowski, All-instances termination of chase is undecidable, in: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, ICALP'14, Springer, 2014, pp. 293–304. doi:10.1007/978-3-662-43951-7_25.