Enumerating Answers to Ontology-Mediated Queries: Partial Answers and Efficiency (Extended Abstract)

Carsten Lutz and Marcin Przybyłko

Department of Computer Science, University of Bremen, Germany

Ontology-mediated query evaluation has mostly been studied in the form of *single-testing*: given an ontology-mediated query (OMQ) $Q(\bar{x}) = (\mathcal{O}, \mathbf{S}, q)$, a database D over schema $\mathbf{S}$, and a candidate answer $\bar{a} \in \mathsf{adom}(D)^{|\bar{x}|}$, decide whether $\bar{a} \in Q(D)$ [2, 4, 7, 8]. From a practical perspective, however, it is often not realistic to assume that a candidate answer is available. This leads us to study answer *enumeration* where only Q and D are given as an input, and the task is to produce all answers without repetitions, in an unspecified order. More precisely, an enumeration algorithm works in two phases. In the *preprocessing phase*, the algorithm uses Q and D to construct a data structure to be used later on, but no output. In the *enumeration phase*, it uses the precomputed structure to output all tuples from $Q(D)$. Related to enumeration is *all-testing* which initially gets the same two inputs and has the same preprocessing phase, followed by a *testing phase* where the algorithm repeatedly receives candidate answers $\bar{a} \in \mathsf{adom}(D)^{|\bar{x}|}$ as additional inputs and returns 'yes' or 'no' depending on whether $\bar{a} \in Q(D)$.

These modes of query evaluation have been extensively studied in database theory, see for example [3, 6, 9–13, 15]. A case of particular importance is enumeration in CD$\circ$Lin, where the preprocessing takes time linear in the size of the database D and the delay between two answers is independent of D. Note that there is no restriction on how the running time of the preprocessing or how the delay depends on the OMQ Q. This corresponds to *data complexity* in single-testing where Q is fixed and thus of constant size. An excellent recent survey of the work on answer enumeration in database theory is [5].

We consider enumeration and all-testing for two kinds of answers: the traditional *certain answers*, where $\bar{a} \in Q(D)$ if and only if $\bar{a}$ is a tuple of constants from D such that $\bar{a} \in q(\mathcal{I})$ for every model $\mathcal{I}$ of D and $\mathcal{O}$, and a novel notion of *partial answers* that is able to take into account fresh constants introduced by existential quantifiers in $\mathcal{O}$ ('*nulls*'). We next define the latter.

Fix a wildcard symbol '$*$' that cannot occur as a constant in a database. A *wildcard tuple* for a database D takes the form $(c_1, \ldots, c_n) \in (\mathsf{adom}(D) \cup \{*\})^n$ where $n \geq 0$ and $\mathsf{adom}(D)$ denotes the set of constants used in D. For wildcard tuples $\bar{c} = (c_1, \ldots, c_n)$ and $\bar{c}' = (c'_1, \ldots, c'_n)$, we write $\bar{c} \preceq \bar{c}'$ if $c'_i \in \{c_i, *\}$ for $1 \leq i \leq n$. Moreover, $\bar{c} \prec \bar{c}'$ if $\bar{c} \preceq \bar{c}'$ and $\bar{c} \neq \bar{c}'$. Intuitively, $\bar{c} \prec \bar{c}'$ expresses that tuple

$\bar{c}$ carries more information than tuple $\bar{c}'$. For example, $(a, b) \prec (a, *) \prec (*, *)$. A *partial answer* to OMQ $Q(\bar{x}) = (\mathcal{O}, \mathbf{S}, q)$ on $\mathbf{S}$-database D is a wildcard tuple $\bar{c}$ for D of length $|\bar{x}|$ such that for every model $\mathcal{I}$ of D and $\mathcal{O}$, there is a $\bar{c}' \in q(\mathcal{I})$ such that $\bar{c}' \preceq \bar{c}$. Note that some positions in $\bar{c}'$ may contain constants that are not in $\mathsf{adom}(D)$, and that the corresponding positions in $\bar{c}$ must have the wildcard. We say that a partial answer $\bar{c}$ to Q on $\mathbf{S}$-database D is a *least partial answer* if there is no partial answer $\bar{c}'$ to Q on D with $\bar{c}' \prec \bar{c}$. We are then interested in enumerating the set $Q(D)^*$ of all least partial answers to Q on D.

Example 1. Consider the ontology $\mathcal{O}$ that contains the TGDs

$$\mathsf{Researcher}(x) \to \exists y\, \mathsf{HasOffice}(x, y) \qquad \mathsf{HasOffice}(x, y) \to \mathsf{Office}(y)$$
$$\mathsf{Office}(x) \to \exists y\, \mathsf{InBuilding}(x, y)$$

the schema $\mathbf{S}$ that consists of all relation symbols in $\mathcal{O}$, and the CQ

$$q(x_1, x_2, x_3) = \mathsf{Researcher}(x_1) \wedge \mathsf{HasOffice}(x_1, x_2) \wedge \mathsf{InBuilding}(x_2, x_3)$$

giving rise to OMQ $Q(x_1, x_2, x_3) = (\mathcal{O}, \mathbf{S}, q)$. Further consider the database

$$D = \{\, \mathsf{Researcher}(\mathsf{mary}), \mathsf{Researcher}(\mathsf{mike}), \mathsf{HasOffice}(\mathsf{mary}, \mathsf{room1}) \,\}$$

Then $Q(D) = \emptyset$, but $Q(D)^* = \{(\mathsf{mary}, \mathsf{room1}, *), (\mathsf{mike}, *, *)\}$.

This abstract reports about the forthcoming article [14] where we consider guarded TGDs $\mathbb{G}$ and the description logic $\mathcal{ELI}$ as the ontology language and conjunctive queries (CQs) as the query language. Recall that, up to syntactic normalization, $\mathcal{ELI}$ is a fragment of $\mathbb{G}$. Our main result is as follows where *complete answers* mean the traditional certain answers.

Theorem 1. *Let $Q = (\mathcal{O}, \mathbf{S}, q)$ be an OMQ from the OMQ language $(\mathbb{G}, \mathbb{CQ})$. If Q is acyclic and free-connex, then the following problems are in CD∘Lin:*

1. *enumeration of complete answers and of least partial answers to Q;*
2. *all-testing of complete answers to Q.*

Let us clarify the notions used in Theorem 1. A CQ $q(\bar{x})$ is *acyclic* if it has a join tree. An acyclic CQ $q(\bar{x})$ is *free-connex* if it remains acyclic after adding an atom $R(\bar{x})$ with R a fresh relation symbol of arity $|\bar{x}|$.

The results for complete answers in Theorem 1 are obtained by reduction to the case without ontologies whereas the result for least partial answers requires the design of a novel enumeration algorithm.

Theorem 1 is accompanied by lower bounds that identify significant challenges in extending enumeration in CD∘Lin beyond OMQs that satisfy the structural properties mentioned in the theorem. As in the case without ontologies, these lower bounds (i) are conditional on certain assumptions whose failure would imply a remarkable advance in algorithm theory and (ii) do not result in fully fledged dichotomies as they rely on additional assumptions regarding the query.

The *triangle conjecture* states that it is not possible, given an undirected graph G with m edges as an adjacency list, to decide in time $O(m)$ whether G contains a triangle [1]. *Sparse Boolean matrix multiplication* means to compute, given two Boolean matrices A and B as a list of their non-zero entries, the non-zero entries of the matrix product AB over the Boolean semiring, see e.g. [16]. There is no known algorithm that solves sparse Boolean matrix multiplication in time $O(m)$, m the sum of the numbers of non-zero entries of A, B, and AB. If such an algorithm exists, then finding it requires dramatic advances in algorithm theory. See e.g. [5] for more information.

Theorem 2. *Let $Q = (\mathcal{O}, \mathbf{S}, q)$ be an OMQ from the OMQ language $(\mathcal{ELI}, \mathbb{CQ})$ that is non-empty and self-join free.*

1. *If q is not acyclic, then enumeration of Q is not in CD∘Lin unless the triangle conjecture fails, both for complete answers and for least partial answers.*
2. *If q is connected and acyclic, but not free-connex, then enumeration of Q is not in CD∘Lin unless sparse Boolean matrix multiplication is possible in time $O(m)$, both for complete answers and for least partial answers.*

We also show that least partial answers cannot be added to Point 2 of Theorem 1 as there is an OMQ $Q \in (\mathbb{ELI}, \mathbb{CQ})$ that is free-connex acyclic such that all-testing least partial answers to Q is not in CD∘Lin unless the triangle conjecture fails.

Finally, enumeration and all-testing in CD∘Lin is closely related to single-testing in linear time (in data complexity), and we also clarify the limits of that.

Theorem 3.

1. *Single-testing is in linear time for weakly acyclic OMQs from $(\mathbb{G}, \mathbb{CQ})$.*
2. *Let Q be an OMQ from $(\mathcal{ELI}, \mathbb{CQ})$ that is non-empty and self-join free. If Q is not weakly acyclic, single-testing for Q is not in linear time unless the triangle conjecture fails.*

Here, a CQ $q(\bar{x})$ is *weakly acyclic* if it becomes acyclic after consistently replacing all answer variables with fresh constants (and thus the connectedness condition of join trees only applies to quantified variables).

Acknowledgements. This research was funded by DFG project QTEC. We thank the anonymous reviewers for useful comments.

References

1. Abboud, A., Williams, V.V.: Popular conjectures imply strong lower bounds for dynamic problems. In: Proceedings of FOCS 2014. pp. 434–443. IEEE Computer Society (2014). https://doi.org/10.1109/FOCS.2014.53
2. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995), http://webdam.inria.fr/Alice/

3. Bagan, G., Durand, A., Grandjean, E.: On acyclic conjunctive queries and constant delay enumeration. In: Duparc, J., Henzinger, T.A. (eds.) Proceedings of CSL 2007. Lecture Notes in Computer Science, vol. 4646, pp. 208–222. Springer (2007). https://doi.org/10.1007/978-3-540-74915-8_18
4. Barceló, P., Dalmau, V., Feier, C., Lutz, C., Pieris, A.: The limits of efficiency for open- and closed-world query evaluation under guarded TGDs. In: Suciu, D., Tao, Y., Wei, Z. (eds.) Proceedings of PODS 2020. pp. 259–270. ACM (2020). https://doi.org/10.1145/3375395.3387653
5. Berkholz, C., Gerhardt, F., Schweikardt, N.: Constant delay enumeration for conjunctive queries: a tutorial. ACM SIGLOG News **7**(1), 4–33 (2020). https://doi.org/10.1145/3385634.3385636
6. Berkholz, C., Schweikardt, N.: Constant delay enumeration with fpt-preprocessing for conjunctive queries of bounded submodular width. In: Rossmanith, P., Heggernes, P., Katoen, J. (eds.) Proceedings of MFCS 2019. LIPIcs, vol. 138, pp. 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.MFCS.2019.58
7. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. ACM Trans. Database Syst. **39**(4), 33:1–33:44 (2014). https://doi.org/10.1145/2661643
8. Bienvenu, M., Ortiz, M.: Ontology-mediated query answering with data-tractable description logics. In: Faber, W., Paschke, A. (eds.) Proceedings of Reasoning Web. Lecture Notes in Computer Science, vol. 9203, pp. 218–307. Springer (2015). https://doi.org/10.1007/978-3-319-21768-0_9
9. Carmeli, N., Kröll, M.: Enumeration complexity of conjunctive queries with functional dependencies. Theory Comput. Syst. **64**(5), 828–860 (2020). https://doi.org/10.1007/s00224-019-09937-9
10. Carmeli, N., Kröll, M.: On the enumeration complexity of unions of conjunctive queries. ACM Trans. Database Syst. **46**(2), 5:1–5:41 (2021). https://doi.org/10.1145/3450263
11. Carmeli, N., Zeevi, S., Berkholz, C., Kimelfeld, B., Schweikardt, N.: Answering (unions of) conjunctive queries using random access and random-order enumeration. In: Suciu, D., Tao, Y., Wei, Z. (eds.) Proceedings of PODS 2020. pp. 393–409. ACM (2020). https://doi.org/10.1145/3375395.3387662
12. Deep, S., Hu, X., Koutris, P.: Enumeration algorithms for conjunctive queries with projection. In: Yi, K., Wei, Z. (eds.) Proceedings of ICDT 2021. LIPIcs, vol. 186, pp. 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ICDT.2021.14
13. Deep, S., Koutris, P.: Ranked enumeration of conjunctive query results. In: Yi, K., Wei, Z. (eds.) Proceedings of ICDT 2021. LIPIcs, vol. 186, pp. 5:1–5:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ICDT.2021.5
14. Lutz, C., Przybyłko, M.: Enumerating answers to ontology-meditated queries. To appear on arXiv
15. Segoufin, L.: Constant delay enumeration for conjunctive queries. SIGMOD Rec. **44**(1), 10–17 (2015). https://doi.org/10.1145/2783888.2783894
16. Yuster, R., Zwick, U.: Fast sparse matrix multiplication. ACM Trans. Algorithms **1**(1), 2–13 (2005). https://doi.org/10.1145/1077464.1077466