Combined Complexity of Answering Tree-like Queries in OWL 2 QL

Meghyn Bienvenu[1], Stanislav Kikot[2], and Vladimir Podolskii[3]

[1] LRI – CNRS & Université Paris Sud, Orsay, France
[2] Institute for Information Transmission Problems & MIPT, Moscow, Russia
[3] Steklov Mathematical Institute & Higher School of Economics, Moscow, Russia

Introduction The OWL 2 QL ontology language [11], based upon the description logic *DL-Lite*$_R$, is considered particularly well suited for applications involving large amounts of data. While the data complexity of querying OWL 2 QL knowledge bases is well understood, far less is known about combined complexity of conjunctive query (CQ) answering for restricted classes of conjunctive queries. By contrast, the combined complexity of CQ answering in the relational setting has been thoroughly investigated.

In relational databases, it is well known that CQ answering is NP-complete in the general case. A seminal result by Yannakakis established the tractability of answering tree-shaped (aka acyclic) CQs [14], and this result was later extended to wider classes of queries, most notably to bounded treewidth CQs [5]. Gottlob et al. [6] pinpointed the precise complexity of answering tree-shaped and bounded treewidth CQs, showing both problems to be complete for the class LOGCFL of all languages logspace-reducible to context-free languages [13]. In the presence of arbitrary OWL 2 QL ontologies, the NP upper bound for arbitrary CQs continues to hold [4], but answering tree-shaped queries becomes NP-hard [8]. Interestingly, the latter problem was recently proven tractable in [3] for DL-Lite$_{\mathsf{core}}$ (a slightly less expressive logic than OWL 2 QL), raising the hope that other restrictions might also yield tractability.

This extended abstract summarizes our investigation [2] into the combined complexity of conjunctive query answering in OWL 2 QL for tree-shaped queries, their restriction to linear and bounded leaf queries and their generalization to bounded treewidth queries. Our complexity analysis reveals that all query-ontology combinations that have not already been shown NP-hard are in fact tractable. Specifically, in the case of bounded depth ontologies, we prove membership in LOGCFL for bounded treewidth queries (generalizing the result in [6]) and membership in NL for bounded leaf queries. We also show LOGCFL-completeness for linear and bounded leaf queries in the presence of arbitrary OWL 2 QL ontologies. This last result is the most interesting technically, as the upper and lower bounds rely on two different characterizations of the class LOGCFL.

Preliminaries We assume the reader familiar is OWL 2 QL (or *DL-Lite*$_R$) knowledge bases (KBs), composed of a TBox $\mathcal{T}$ and ABox $\mathcal{A}$ built from countably infinite, mutually disjoint sets $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ of *concept names*, *role names*, and *individual names*. *Roles* R and *basic concepts* B are defined in a standard way, cf. [4]. We use $\mathsf{N}_R^{\pm}$ to refer to the set of all roles. We recall that every consistent *OWL 2 QL* KB $(\mathcal{T}, \mathcal{A})$ possesses a *canonical model* $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ with the property that $\mathcal{T}, \mathcal{A} \models \mathbf{q}(\mathbf{a})$ iff $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models \mathbf{q}(\mathbf{a})$ for every CQ $\mathbf{q}$ and tuple $\mathbf{a} \subseteq \mathsf{inds}(\mathcal{A})$. Thus, CQ answering in *OWL 2 QL* corresponds to *deciding the existence of a homomorphism of the query into the canonical model*. Informally,

$\mathcal{C}_{\mathcal{T},\mathcal{A}}$ is obtained from $\mathcal{A}$ by repeatedly applying the axioms in $\mathcal{T}$, introducing fresh elements as needed to serve as witnesses for the existential quantifiers. According to the standard construction (cf. [10]), the domain $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ consists of $\mathsf{inds}(\mathcal{A})$ and all words of the form $aR_1R_2\dots R_{n-1}R_n$ ($n \geq 1$) with $a \in \mathsf{N_C}$ and $R_i \in \mathsf{N}_R^{\pm}$. Intuitively, the element $aR_1R_2\dots R_{n-1}R_n$ is obtained by applying an axiom with right-hand side $\exists R_n$ to the element $aR_1R_2\dots R_{n-1} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. A TBox $\mathcal{T}$ is *of depth* ω if there is an ABox $\mathcal{A}$ such that the domain of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ is infinite; $\mathcal{T}$ is *of depth* d, $0 \leq d < \omega$, if d is the greatest number such that some $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ contains an element of the form $aR_1\dots R_d$.

Contributions In what follows, we briefly formulate our combined complexity results and provide some intuitions about the proof techniques. See [2] for details.

Theorem 1. *CQ answering is in* LOGCFL *for bounded treewidth queries and bounded depth ontologies.*

Proof sketch. We exploit the fact that CQ answering over a KB $(\mathcal{T},\mathcal{A})$ corresponds to evaluating the query over the canonical model $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ viewed as a database. If $\mathcal{T}$ has depth k (with k a fixed constant), then $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ can be computed by a deterministic logspace Turing machine (TM) with access to an NL oracle. Indeed, the depth bound k implies the finiteness of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ and that all domain elements can be described using logarithmically many bits. To complete the argument, we use the fact that answering bounded treewidth queries over databases is in LOGCFL [7] and that the class LOGCFL is closed under $\mathrm{L}^{\mathrm{LOGCFL}}$ (and hence L^{NL}) reductions [13]. □

Theorem 2. *CQ answering is* NL*-complete for bounded leaf queries and bounded depth ontologies.*

Proof sketch. The lower bound is an immediate consequence of the NL-hardness of answering atomic queries in OWL 2 QL. To prove the upper bound, we apply a straightforward non-deterministic procedure for deciding $(\mathcal{T},\mathcal{A}) \models q$:

1. Fix a directed tree T compatible with q. Let v_0 be the root variable.
2. Guess $u_0 \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. Return **no** if v_0 *cannot be mapped* to u_0.
3. Initialize Frontier to $\{(v_0,u_0)\}$.
4. While Frontier $\neq \emptyset$
   (a) Remove (v_1,u_1) from Frontier.
   (b) For every child v_2 of v_1
      i. Guess an element u_2 from $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$.
      ii. Return **no** if (v_1,v_2) *cannot be mapped* to (u_1,u_2).
      iii. Add (v_2,u_2) to Frontier.
5. Return **yes**.

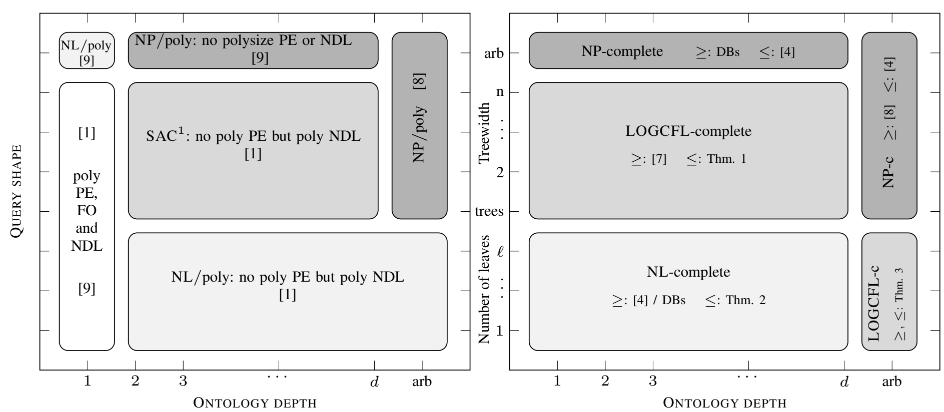
For lack of space, we have not specified how to check whether a variable (resp. pair of variables) can be mapped to an element (resp. pair of elements), but this can be done in NL using a small number of entailment checks. Also note that the bound on the number of leaves yields the bound on size of Frontier, and the bound on the TBox depth guarantees that we only need logarithmically many bits per pair in Frontier. □

Theorem 3. *CQ answering is* LOGCFL*-complete for bounded leaf queries and arbitrary ontologies. The lower bound holds already for linear queries.*

Proof sketch. Concerning the upper bound, it is easy to adapt the previous algorithm to handle arbitrary TBoxes: we simply replace $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ by $\{aw \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}} \mid |w| \leq 2|\mathcal{T}| + |q|\}$. The modified algorithm gives the correct answers, but it does not have the required complexity, because it might need more than logarithmically many bits to store guessed elements aw. To show LOGCFL membership, we further modify the procedure so that it can be implemented by a non-deterministic polytime logspace-bounded Turing machine augmented with a stack (such TMs are known to capture LOGCFL computation [12]). The stack is used to store the word part w of a domain element aw. The modification is not at all obvious since we need to store several words at a time while the specified machine has only a single stack; the trick is to employ a careful 'synchronization' of traversals of different branches of the query.

The lower bound is by reduction from the problem of deciding whether an input of length l is accepted by the lth circuit of a *logspace-uniform* family of SAC^1 circuits (proven LOGCFL-hard in [13]). This problem was used in [7] to show LOGCFL-hardness of evaluating tree-shaped CQs over databases. We follow a broadly similar approach, but with one crucial difference: the power of OWL 2 QL TBoxes allows us to 'unravel' the circuit into a tree and to use linear queries instead of tree-shaped ones. □

Discussion If we compare the new and existing results for OWL 2 QL with those from relational databases, we observe that adding an OWL 2 QL TBox of bounded depth does not change the combined complexity for query answering, while for TBoxes of unbounded depth, the complexity class shifts one 'step' higher: from NL to LOGCFL for bounded leaf queries and from LOGCFL to NP for tree-shaped and bounded treewidth CQs. It is also interesting to compare the combined complexity landscape (below right) with the succinctness landscape for query rewriting (below left) from [1].



Observe that for our newly identified tractable classes, polynomial-size non-recursive datalog (NDL) rewritings are guaranteed to exist, whereas this is not the case for the positive existential (PE) rewritings more typically considered. In future work, we plan to marry these positive succinctness and complexity results by developing concrete NDL-rewriting algorithms for OWL 2 QL for which both the rewriting and evaluation phases run in polynomial time (as was done in [3] for DL-Lite$_{\mathsf{core}}$).

Acknowledgments. Partial support was provided by ANR grant 12-JS02-007-01, Russian Foundation for Basic Research and the programme "Leading Scientific Schools".

References

1. M. Bienvenu, S. Kikot, and V. V. Podolskii. Succinctness of query rewriting in OWL 2 QL: the case of tree-like queries. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, pages 45–57, 2014.
2. M. Bienvenu, S. Kikot, and V. V. Podolskii. Tree-like queries in OWL 2 QL: Succinctness and complexity results. In *Proc. of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015)*. IEEE, 2015.
3. M. Bienvenu, M. Ortiz, M. Simkus, and G. Xiao. Tractable queries for lightweight description logics. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI 2013)*. AAAI Press, 2013.
4. 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.
5. C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000.
6. G. Gottlob, N. Leone, and F. Scarcello. Computing LOGCFL certificates. In *ICALP-99*, pages 361–371, 1999.
7. G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. *J. ACM*, 48(3):431–498, 2001.
8. S. Kikot, R. Kontchakov, V. V. Podolskii, and M. Zakharyaschev. Exponential lower bounds and separation for query rewriting. In *Proc. of the 39th Int. Colloquium on Automata, Languages, and Programming (ICALP 2012), Part II*, volume 7392 of *LNCS*, pages 263–274. Springer, 2012.
9. S. Kikot, R. Kontchakov, V. V. Podolskii, and M. Zakharyaschev. On the succinctness of query rewriting over OWL 2 QL ontologies with shallow chases. In *Proc. of the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2014)*. ACM Press, 2014.
10. R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to query answering in DL-Lite. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010.
11. B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language profiles. W3C Recommendation, 11 December 2012. Available at http://www.w3.org/TR/owl2-profiles/.
12. I. H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the ACM*, 25(3):405–414, 1978.
13. H. Venkateswaran. Properties that characterize LOGCFL. *J. Computer and System Sciences*, 43(2):380–404, 1991.
14. M. Yannakakis. Algorithms for acyclic database schemes. In *Proc. of the 7th Int. Conf. on Very Large Data Bases (VLDB'81)*, pages 82–94. IEEE Computer Society, 1981.