Exploiting Semantic Treewidth for Graph Queries Evaluation

Cristina Feier[1], Tomasz Gogacz[1] and Filip Murlak[1]

[1]University of Warsaw, Poland

Abstract

Conjunctive two-way regular path queries (C2RPQs) are a common abstraction of the core of query languages for graph databases, much like conjunctive queries (CQs) in the relational case. As in the case of CQs, their evaluation is NP-complete. Also, similarly to the CQ case, existing work on efficient evaluation showed that bounded semantic treewidth (TW) is a sufficient condition for fixed-parameter tractable (fpt) evaluation. This has been achieved by providing witnesses for bounded semantic TW, i.e., equivalent queries of bounded syntactic TW. In subsequent steps, a witness of TW up to twice the semantic TW of the query has been constructed, followed more recently by a witness of TW equal to the semantic TW of the query. Both witnesses can be exploited to obtain fpt evaluation algorithms, the latter giving better data complexity due to the optimal treewidth. However, while the size of the suboptimal-TW witness is exponential in the size of the query, the size of the optimal-TW witness is doubly exponential, which leads to an evaluation algorithm doubly exponential in the size of the query. We show that the additional blow-up in combined complexity can be avoided by exploiting the connection between the two witnesses. We devise an evaluation algorithm that fully exploits the equivalent witness of minimal TW without actually computing it, and runs in time singly exponential in the size of the query.

Keywords

conjunctive two-way regular path queries, fixed-parameter tractable evaluation, semantic treewidth

1. Introduction

Graph databases are nowadays mainstream [1] in a wide range of application domains, including social networks, fraud detection, biological networks, bioinformatics, cheminformatics, medical data, and knowledge management [2]. While for relational databases conjunctive queries (CQs) are a prèmiere abstract query language, graph databases are typically queried using *conjunctive (two-way) regular path queries* (C2RPQs) which generalize conjunctive queries by replacing atoms with *(two-way) regular path queries* (2RPQs) [3]. The complexity of evaluating C2RPQs is the same as for CQs: NP-complete.

In the case of CQs, there is a long line of research concerning efficient evaluation. This spans from the well-known polynomial time Yannakakis' algorithm for evaluating acyclic CQs [4] to tractability results for queries of bounded treewidth [5], bounded (fractional) hypertreewidth [6, 7], and complete characterizations of fixed-paramater tractability in the bounded [8] and unbounded arity setting [9, 10]. It turns out that for such characterizations, *semantic* measures play an important role: these are structural measures, like treewidth (TW) or submodular width, modulo query equivalence.

15th Alberto Mendelzon International Workshop on Foundations of Data Management, May 22–26, 2023, Santiago, Chile

CEUR Workshop Proceedings (CEUR-WS.org)

In the context of the semantic TW of C2RPQs, Romero, Barcelo, and Vardi introduced two notions of equivalence of C2RPQs [11]: one based on homomorphisms, and another based on logical equivalence. For the homomorphism-based notion, C2RPQs of bounded semantic TW are tractable, but the techniques used to show tractability in this case, based on existential pebble games [12], are no longer applicable in the logical equivalence setting [11]. However, in the latter setting, it is shown that C2RPQs of bounded semantic TW are fixed parameter tractable (fpt) by computing an actual witness, i.e. an equivalent query of bounded TW. The size of this query is exponential in the size of the original query Φ, however its TW is not optimal: it can be up to $2k + 1$, when the semantic TW of Φ is k. This leads to an algorithm for evaluating C2RPQs of semantic TW k which runs in time $O(f(|\Phi|)|D|^{2k+2})$, with f being an exponential function.

It remained open how to decide semantic TW of C2RPQs under the logical equivalence notion, and, in particular, how to construct a witness of optimal TW. This has been settled recently by Figueira and Morvan [13], by constructing a witness for semantic TW of size doubly exponential in the size of the original query. This leads to an algorithm for evaluating C2RPQs of semantic TW k which runs in time $O(f(\Phi)|D|^{k+1})$, with f being a doubly exponential function. While this is a significant improvement in data complexity compared to existing algorithms, the algorithm is impractical due to its high combined complexity.

Here we describe a more efficient algorithm for evaluating C2RPQs of semantic treewidth k which runs in time $O(g(|\Phi|)|D|^{k+1})$, where g is an exponential function. We observe that the TW-k witness query from [13] can be seen as a specialization of the TW-$(2k + 1)$ query from [14]. This enables us to encode the evaluation of the TW-k witness query into a Datalog program of width $k + 1$ without explicitly constructing the query. As the size of the Datalog program is exponential in the size of the original query, we obtain the desired result.

2. Preliminaries

We assume familiarity with CQs, databases, treewidth, pathwidth, Datalog, and non-deterministic finite automata (NFA). We note that every CQ q of TW k can be evaluated over database D in time $O(p(|q|)|D|^{k+1})$, where p is a polynomial [8]. Also, the *width* of a Datalog program Π is the maximum number of variables occurring in a rule in Π. Every such program Π of width k can be evaluated in time $O(p(|\Pi|)|D|^{k+1})$, where p is some polynomial.

Let us fix a countable alphabet Σ. A *graph database* $\mathcal{G}$ over Σ is a finite directed graph with edges labeled with symbols from Σ. A path p in a graph database is a sequence $(u_1 \xrightarrow{a_1} u_2 \ldots u_l \xrightarrow{a_l} u_{l+1})$. The label of p as above, denoted $\lambda(p)$, is the word $a_1 \ldots a_l$ from Σ^*.

A *regular path query (RPQ)* is a query of the form $L(u, v)$ where L is a regular language over Σ, expressed as an NFA. Assuming s and s' are states of the NFA L, $L[s, s']$ is the NFA obtained from L by having s as initial state and s' as a unique final state. For a graph database $\mathcal{G}$ and nodes u, v in $\mathcal{G}$, $\mathcal{G} \models L(u, v)$ if there exists a path p in $\mathcal{G}$ from u to v such that $\lambda(p) \in L$. When L is over the extended alphabet $\Sigma^{\pm} = \Sigma \cup \{a^- \mid a \in \Sigma\}$, we say that $L(u, v)$ is a *two-way RPQ (2RPQ)*; it is evaluated over the *completion* $\mathcal{G}^{\pm}$ of $\mathcal{G}$ with edges of the form $u \xrightarrow{a^-} v$, whenever $v \xrightarrow{a} u \in \mathcal{G}$.

A Boolean *conjunctive (two-way) regular path query*, abbreviated as C(2)RPQ, is a query of

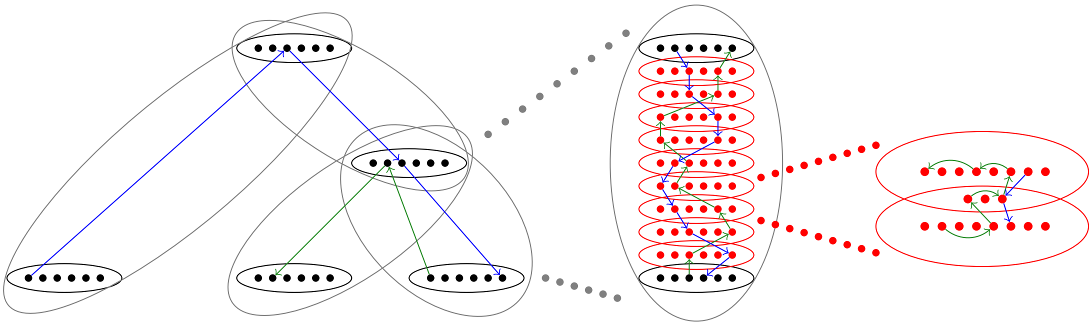

Figure 1: The construction of the two Witness Queries: ϕ_w and ϕ_k

the form $\phi = \exists \mathbf{z} \bigwedge_{1 \leq i \leq n} L(u_i, v_i)$ where each $L(u_i, v_i)$ is a (2)RPQ and $\mathbf{z}$ is the set of variables occurring in the query. We write $\mathcal{G} \models \phi$ iff there is a mapping $h : \mathbf{z} \to \mathsf{dom}(\mathcal{G})$ such that $\mathcal{G} \models L(h(u_i), h(v_i))$, for every $1 \leq i \leq n$. Unions of C(2)RPQs, abbreviated as UC(2)RPQs, as well as satisfaction of UC(2)RPQs and containment ($\subseteq$) of (U)C(2)RPQs are defined as expected.

3. Witnesses for Bounded Semantic Treewidth

In this section, we provide an overview of the witness queries from [11] and the ones from [13]. We begin with a definition.

Definition 1. *A UC2RPQ Φ' is a* TW-k approximation *of a C2RPQ Φ if: (i) Φ' has TW k, (ii) $\Phi' \subseteq \Phi$, and (iii) there is no UC2RPQ Φ'' of TW k such that $\Phi' \subset \Phi'' \subseteq \Phi$.*

In the limit case, when Φ has semantic TW k, a TW-k approximation is equivalent to Φ and thus a witness for semantic TW. The following lemma describes a sufficient condition for a TW-k query to be an approximation.

Lemma 1. *Let Φ be a C2RPQ and Φ' be a TW-k UC2RPQ s.t. $\Phi' \subseteq \Phi$ and Φ' is equivalent to Φ on TW-k databases. Then Φ' is a TW-k approximation of Φ.*

In the spirit of Lemma 1, the approach from [11] constructs a witness Φ_w for bounded semantic TW of Φ, by considering all C2RPQs induced by mappings of Φ into TW-k databases D_k. For every such mapping, a set of relevant bags in D_k (up to $2|\Phi|$) is identified, and one considers all intersection points of images of atoms in Φ with nodes from these bags. Recall that the image of an atom is a path in D_k: such images of atoms are marked with green and blue in Figure 1). These intersection points induce splittings of atoms $L(u, v)$ in Φ into sequences of atoms of the form $L[s_0, s_1](u_0, u_1), \ldots, L[s_{n-1}, s_{n+1}](u_{n-1}, u_n)$, where $u_0 = u$, $u = v$, and s_0 and s_n are the initial state and some final state of L, respectively. In a subsequent step, all variables of the new and old atoms which map to the same database element are identified. Φ_w is defined as the union of all C2RPQs obtained by the above procedure; see Figure 1 (left).

As explained in the introduction, Φ_w has TW up to $2k + 1$: this is due to the fact that each of its C2RPQs has as the underlying graph a pseudo-tree decomposition of width k, i.e. a tree decomposition from which some non-branching nodes are removed and the remaining nodes are re-connected via new atoms. We will refer to such a sub-query induced by two bags in a tree decomposition and some interconnecting atoms as a *segment*.

The UC2RPQ Φ_k constructed in [13] as a witness of semantic TW can be seen as a specialization of the query Φ_w in the sense that atoms connecting endpoints of segments in C2RPQs in Φ_w are replaced by actual queries of pathwidth k (see Figure 1, middle), thus recovering TW k of the witness query. This is achieved by first considering the infinitary UC2RPQ which contains all C2RPQs induced by mappings of Φ into a TW-k database (this time, all elements of D_k are considered relevant and induce splittings and variable identifications in Φ). Clearly, this query is a specialization of Φ_w in the sense described above. Then, in a second step, using a pumping argument, it can be shown that it is enough to consider C2RPQs with segment realizations up to a certain size (exponential in Φ).

As the size of Φ_w is exponential in $|\Phi|$ and the size of Φ_k is doubly exponential in $|\Phi|$, and each C2RPQ can be evaluated by first materializing (in polynomial time) each 2RPQ in the database and then simply regarding the query as a CQ over the materialized database, one obtains the fpt algorithms with the complexities mentioned in the introduction.

4. The algorithm

Our algorithm for efficient evaluation of C2RPQs of semantic TW k is based on evaluating Φ_k without actually computing it. This is possible due to the fact that for queries of semantic TW k, $\Phi_k \equiv \Phi_w$ and each segment query from Φ_w is equivalent to the union of all of its realizations in some C2RPQ in Φ_k. Under this assumption, we can actually evaluate Φ_w instead of Φ_k. As explained in Section 3, the realization of a segment from Φ_w in Φ_k is a query of pathwidth k. Each bag from such a path query can be seen as having a certain type induced by atoms which cross it, i.e. atoms for which at least one of the arguments belongs to the bag. The important thing is that for each atom $L(u, v)$ in the original query Φ, there is at most one atom of the form $L[s_1, s_2](u', v')$ that crosses such a bag (derived from a splitting of the original atom). Thus, we can type each bag based on the signature of such atoms. There is an exponential number of types and a natural notion of bags/type compatibility: Figure 1 (right) depicts how two bags with compatible types are matched in a segment. In this view, each segment query from Φ_w can be seen as a reachability query between two $(k + 1)$-types and thus can be encoded using Datalog rules with at most $k + 1$ variables.

For joining the segments in a C2RPQ of Φ_w, the algorithm performs bottom-up evaluation. On an abstract level, it can be seen as a Yannakakis-style procedure [4], in which we have oracles to compute the upper end of a segment given its lower end. Again, this is easy to capture using $(k + 1)$-Datalog rules. We obtain a Datalog program Π of exponential size (exponential number of rules) and of width $k + 1$ which is equivalent to Φ. This proves our main result.

Theorem 1. *Every C2RPQ Φ of semantic TW k can be evaluated in time $O(f(|\Phi|)|D|^{k+1})$, where f is a singly exponential function.*

5. Outlook

There are several open questions regarding efficient evaluation of C2RPQs. The first one concerns the precise complexity of deciding semantic treewidth: for now it is known that it is ExpSpace-hard and in 2ExpSpace. Another major step would be achieving a full characterisation of fixed parameter tractability of UC2RPQs, in particular establishing lower bounds.

Acknowledgments

Supported by Poland's NCN grant 2018/30/E/ST6/00042.

References

[1] S. Sakr, A. Bonifati, H. Voigt, A. Iosup, K. Ammar, R. Angles, W. Aref, M. Arenas, M. Besta, P. A. Boncz, K. Daudjee, E. D. Valle, S. Dumbrava, O. Hartig, B. Haslhofer, T. Hegeman, J. Hidders, K. Hose, A. Iamnitchi, V. Kalavri, H. Kapp, W. Martens, M. T. Özsu, E. Peukert, S. Plantikow, M. Ragab, M. R. Ripeanu, S. Salihoglu, C. Schulz, P. Selmer, J. F. Sequeda, J. Shinavier, G. Szárnyas, R. Tommasini, A. Tumeo, A. Uta, A. L. Varbanescu, H.-Y. Wu, N. Yakovets, D. Yan, E. Yoneki, The future is big graphs: A community view on graph processing systems, Commun. ACM 64 (2021) 62–71.

[2] R. Angles, C. Gutierrez, Survey of graph database models, ACM Comput. Surv. 40 (2008) 1:1–1:39.

[3] A. O. Mendelzon, P. T. Wood, Finding regular simple paths in graph databases, SIAM Journal on Computing 24 (1995) 1235–1258. doi:10.1137/S009753979122370X.

[4] M. Yannakakis, Algorithms for acyclic database schemes, in: VLDB, 1981, pp. 82–94.

[5] C. Chekuri, A. Rajaraman, Conjunctive query containment revisited, Theor. Comput. Sci. 239 (2000) 211–229.

[6] G. Gottlob, N. Leone, F. Scarcello, Hypertree decompositions and tractable queries, Journal of Computer and System Sciences 64 (2002) 579–627.

[7] M. Grohe, D. Marx, Constraint solving via fractional edge covers, ACM Trans. Algorithms 11 (2014) 4:1–4:20.

[8] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, J. ACM 54 (2007) 1:1–1:24.

[9] D. Marx, Tractable hypergraph properties for constraint satisfaction and conjunctive queries, in: STOC, 2010, pp. 735–744.

[10] H. Chen, G. Gottlob, M. Lanzinger, R. Pichler, Semantic width and the fixed-parameter tractability of constraint satisfaction problems, in: C. Bessiere (Ed.), IJCAI, 2020, pp. 1726–1733.

[11] M. Romero, P. Barceló, M. Y. Vardi, The homomorphism problem for regular graph patterns, in: 2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2017, pp. 1–12.

[12] P. Kolaitis, M. Vardi, On the expressive power of datalog: Tools and a case study, Journal of Computer and System Sciences 51 (1995) 110–134.

[13] D. Figueira, R. Morvan, Approximation and semantic tree-width of conjunctive regular path queries, in: ICDT, volume 255 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 15:1–15:19.

[14] P. Barceló, M. Romero, M. Y. Vardi, Semantic acyclicity on graph databases, SIAM Journal on Computing 45 (2016) 1339–1376.