On the Complexity of Enumerating the Answers to Well-Designed Pattern Trees (Ext. Abstract)*

Markus Kröll, Reinhard Pichler, and Sebastian Skritek

Institute for Information Systems, TU Wien,
lastname@dbai.tuwien.ac.at

1 Introduction

With the steadily increasing amount of incomplete data on the web, the need for partial matching as an extension of Conjunctive Queries (CQs) is gaining more and more importance. Therefore, the OPTIONAL operator is a crucial feature in the semantic web query language SPARQL, and has also been studied for arbitrary relational vocabulary recently [3]. Intuitively, this operator (which corresponds to the left outer join in the Relational Algebra) allows the user to extend CQs by optional parts for which answers are retrieved from the data if available, but which do not cause the query to return no answer at all otherwise.

In [9], based on a query fragment with desirable properties presented in [10], so-called *well-designed pattern trees* (wdPTs) were introduced as a convenient graphical representation of CQs extended by this optional matching feature. Intuitively, the nodes in a wdPT correspond to CQs while the tree structure represents the optional extensions.

Several computational problems of wdPTs have been studied in recent years, such as the evaluation problem, the counting problem, as well as static analysis tasks including the containment and equivalence problems. In contrast, despite recent interest in the enumeration problem (i.e., computing one solution after the other without outputting duplicates) for FO queries and CQs [5, 2, 11], this natural problem has been hardly considered for wdPTs so far. A notable exception is [9]. Similar ideas were pursued in [4] for the enumeration of the answers to full disjunctions. However, being associative and commutative, full disjunctions differ significantly from wdPTs. Hence, the results in [4] do not apply to wdPTs.

In this work, we embark on a systematic study of the complexity of the enumeration problem of wdPTs. We identify several tractable and intractable cases of this problem both from a classical complexity point of view and from a parameterized complexity point of view.

2 Well-Designed Pattern Trees

A *well-designed pattern tree* (wdPT) p over a schema σ is a triple $(T, \lambda, \boldsymbol{x})$ s.t.:

1. T is a rooted tree; λ maps each node in T to a set of relational atoms over σ.

* This is an extended abstract of a paper published at ICDT 2016 [8].

2. For every variable y from T, the set of nodes where y occurs is connected.
3. The tuple $\boldsymbol{x}$ of distinct variables from T denotes the *free variables* of p.

Assume $p = (T, \lambda, \boldsymbol{x})$ is a wdPT over σ. We write r to denote the root of T. For a subtree T' of T rooted in r, we define $q_{T'}$ to be the CQ $\text{Ans}(\boldsymbol{y}) \leftarrow R_1(\boldsymbol{v}_1), \ldots, R_m(\boldsymbol{v}_m)$, where $\{R_1(\boldsymbol{v}_1), \ldots, R_m(\boldsymbol{v}_m)\} = \bigcup_{t \in T'} \lambda(t)$, and $\boldsymbol{y}$ are all the variables that are mentioned in T'. That is, all variables in $q_{T'}$ appear free.

The intuition behind the semantics of a wdPT $p = (T, \lambda, \boldsymbol{x})$ is as follows. A mapping h is an answer to (T, λ) over a database $\mathcal{D}$, if h is a solution to $q_{T'}$ and there is no way to "extend" h to a solution of some $q_{T''}$, where T', T'' are subtrees of T rooted in r. The evaluation of p over $\mathcal{D}$ corresponds then to the projection of the answers to (T, λ) over $\mathcal{D}$ to $\boldsymbol{x}$. Formally: Let $\mathcal{D}$ be a database and $p = (T, \lambda, \boldsymbol{x})$ a wdPT over schema σ. Let $dom(\mathcal{D})$ be the set of elements in the active domain of $\mathcal{D}$ and $\mathbf{X}$ the variables mentioned in p. Then:

- A *homomorphism* from p to $\mathcal{D}$ is a partial mapping $h : \mathbf{X} \to dom(\mathcal{D})$, for which it is the case that there is a subtree T' of T rooted in r such that h is a *homomorphism* from the CQ $q_{T'}$ to $\mathcal{D}$.
- The homomorphism h is *maximal* if there is no homomorphism h' from p to $\mathcal{D}$ such that h' *extends* h.

If h is a homomorphism from p to $\mathcal{D}$, let $h_{\boldsymbol{x}}$ denote the restriction of h to the variables in $\boldsymbol{x}$. The *evaluation* of p over $\mathcal{D}$, denoted $p(\mathcal{D})$, corresponds to all mappings of the form $h_{\boldsymbol{x}}$, such that h is a maximal homomorphism from p to $\mathcal{D}$.

3 Enumeration of wdPTs

Let $\mathcal{C}$ be a class of wdPTs. Given a wdPT $p \in \mathcal{C}$ and a database $\mathcal{D}$, we denote with $\textsc{Enum}(\mathcal{C})$ the problem to enumerate $p(\mathcal{D})$. Because of projection, it may happen that both, a mapping h and a proper extension of h are in $p(\mathcal{D})$. However, in some settings only the *maximal solutions* are of interest, e.g. [1]. We thus also study the problem $\textsc{Max-Enum}(\mathcal{C})$ of enumerating $p_m(\mathcal{D})$, the set of maximal solutions in $p(\mathcal{D})$.

We aim at identifying the boundary between tractability and intractability for these problems. In [6], various notions of tractable enumeration are defined. We concentrate on two such notions, namely polynomial delay (the time until the next output or termination is bounded by a polynomial in the size of the input) and output polynomial time (the total runtime of the algorithm is bounded by a polynomial in the size of the input *plus* the output). We denote with DelayP and $\mathsf{OutputP}$ the corresponding enumeration complexity classes.

The following decision problem $\textsc{ExtSol}(\mathcal{C})$ (where $\mathcal{C}$ is a class of wdPTs) is closely related to the enumeration problems defined above: Given a wdPT $p = (T, \lambda, \boldsymbol{x}) \in \mathcal{C}$, a database $\mathcal{D}$, a partial mapping h and a set $\boldsymbol{x}' \subseteq \boldsymbol{x}$, does there exist some mapping $h' \in p(\mathcal{D})$ s.t. $h \subseteq h'$ and $dom(h') \cap \boldsymbol{x}' = \emptyset$? The relationship with enumeration is given by the following property.

Table 1. Summary of the main results on evaluation and enumeration of wdPTs. Negative results are shown under the assumptions $\mathsf{P} \neq \mathsf{NP}$ ([1]) resp. $\mathsf{FPT} \neq \mathsf{W[1]}$ ([2]).

	ℓ-$\mathsf{TW}(k) \cap \mathsf{BI}(c)$	g-$\mathsf{TW}(k) \cap \mathsf{SBI}(c)$	ℓ-$\mathsf{TW}(k) \cap \mathsf{SBI}(c)$	g-$\mathsf{TW}(k)$
p-Eval($\mathcal{C}$)	in P [3]	in FPT	W[1]-complete	W[2]-hard
p-Max-Eval($\mathcal{C}$)	in P [3]	in P [3]	W[1]-hard	in P [3]
Enum($\mathcal{C}$)	DelayP	*open*	not OutputP[2]	not OutputP[1]
Max-Enum($\mathcal{C}$)	not OutputP[1]	not OutputP[1]	not OutputP[1]	not OutputP[1]
p-Enum($\mathcal{C}$)	DelayP	DelayFPT	not OutputFPT[2]	not OutputFPT[2]
p-Max-Enum($\mathcal{C}$)	DelayFPT	DelayFPT	not OutputFPT[2]	DelayFPT

Lemma 1. *Let $\mathcal{C}$ be a class of wdPTs, $p = (T, \lambda, \boldsymbol{x}) \in \mathcal{C}$, and $\mathcal{D}$ a database. If there is a computable function f s.t. for every partial mapping μ and every subset $\boldsymbol{x}'$ of $\boldsymbol{x}$, the problem* ExtSol($\mathcal{C}$) *can be decided in $\mathcal{O}(f(|p|, |\mathcal{D}|))$, then there exists an algorithm enumerating $p(\mathcal{D})$ with delay $\mathcal{O}(f(|p|, |\mathcal{D}|) \cdot |\mathcal{D}| \cdot |p|)$.*

Also ExtSol($\mathcal{C}$) is a generalization of the evaluation problems (Max-)Eval($\mathcal{C}$): Given $p \in \mathcal{C}$, a database $\mathcal{D}$, and a mapping h, is $h \in p(\mathcal{D})$ (or $p_m(\mathcal{D})$)? Since these problems are known to be intractable in general, so is ExtSol($\mathcal{C}$). As a result, the evaluation problem for fragments of wdPTs has been studied in [3]. We use the same classes of wdPTs, and also introduce further restrictions to get a better understanding of the sources of complexity of enumeration.

Fragments of wdPTs have been defined in two different ways: On the one hand by restrictions on some associated CQs, and on the other hand by restricting the number of variables shared between nodes. Let $\mathsf{TW}(k)$ be the class of CQs with *bounded treewidth*. A wdPT $p = (T, \lambda, \boldsymbol{x})$ is in the class ℓ-$\mathsf{TW}(k)$ if for every node n in T the CQ $\mathrm{Ans}() \leftarrow \lambda(n)$ is in $\mathsf{TW}(k)$, and in the class g-$\mathsf{TW}(k)$ if for every subtree T' of T rooted in r the CQ $q_{T'}$ is in $\mathsf{TW}(k)$. Furthermore, we say p is in $\mathsf{BI}(c)$ if every node n in T contains at most c variables that also occur elsewhere in T, while p is in $\mathsf{SBI}(c)$ if no two nodes in T share more than c variables.

Table 1 summarizes our results. Observe that beside the classical complexity, we also study the parameterized complexity of the enumeration problems. I.e., p-Enum($\mathcal{C}$) and p-Max-Enum($\mathcal{C}$) denote the parameterized variants (with $|p|$ as the parameter) of Enum($\mathcal{C}$) and Max-Enum($\mathcal{C}$), respectively. Also, DelayFPT and OutputFPT are defined analogously to DelayP and OutputP. Since the parameterized complexity of wdPTs has not yet been studied, by p-Eval($\mathcal{C}$) and p-Max-Eval($\mathcal{C}$) we also consider the parameterized variant of the common evaluation problem for wdPTs, i.e., given $p, \mathcal{D}, h$, is $h \in p(\mathcal{D})$ (resp. $p_m(\mathcal{D})$)?

We would like to point out that by introducing the class $\mathsf{SBI}(c)$ and performing a parameterized complexity analysis, we are able to draw a much more fine grained picture of the complexity of the decision problem. So far, only Eval(ℓ-$\mathsf{TW}(k) \cap \mathsf{BI}(c)$) and Eval($g$-$\mathsf{TW}(k)$) have been known to be tractable and NP-complete, respectively [3].

For the enumeration problems, we observe a surprising discrepancy between enumerating all answers and enumerating only the maximal ones. For the former setting, in most of the cases the corresponding decision problem is at least as hard

as for the second setting (this is also the case for the non-parameterized problem [3]). A similar behaviour is observed for p-Enum($\mathcal{C}$) and p-Max-Enum($\mathcal{C}$). This, however, completely changes for Enum($\mathcal{C}$) and Max-Enum($\mathcal{C}$), where Max-Enum($\mathcal{C}$) is not even tractable in the most restricted case.

In the remainder, we would like to sketch some of the tools used to show the results in Table 1. Lemma 1 established a relationship between ExtSol($\mathcal{C}$) and enumeration. This relationship is put to use by the following result.

Proposition 1. *The following complexity results hold for* ExtSol($\mathcal{C}$)*:*

1. *Let $k, c \geq 1$ and $\mathcal{C}$ be a class of CQs for which the evaluation problem is in* P*. Then* ExtSol(ℓ-$\mathcal{C} \cap \mathsf{BI}(c)$) *is in* P*.*
2. ExtSol(g-$\mathsf{TW}(k)$) *is* NP*-complete for every $k \geq 1$.*
3. *For $k, c \geq 1$,* ExtSol(g-$\mathsf{TW}(k) \cap \mathsf{SBI}(c)$) *parameterized by $|p|$ is in* FPT*.*

A useful tool to show negative results is Theorem 1. We call a class $\mathcal{C}$ *robust* if it is closed under some simple transformations (like replacing all occurrences of a variable by the same constant). All classes considered here are robust.

Theorem 1. *Let $\mathcal{C}$ be a robust class of wdPTs. If p-*Enum($\mathcal{C}$) *is in* $\mathsf{OutputFPT}$*, then p-*Eval($\mathcal{C}$) *is in* FPT*.*

Another way of providing intractability results for enumeration problems is by showing that, given a set of solutions, deciding if there exists another one is not possible in polynomial time (cf. [7]). We used this relationship to show the intractability of Max-Enum($\mathcal{C}$) (assuming $\mathsf{P} \neq \mathsf{NP}$) by proving the decision problem to be NP-complete for $\mathcal{C} = \ell$-$\mathsf{TW}(k) \cap \mathsf{BI}(c)$.

4 Further Results and Future Work

In addition to the results presented above, following an active line of research [2, 11], we also had a brief look at the data complexity of the enumeration problems. In this case, the goal is to find an enumeration algorithm that works with constant delay after some linear time preprocessing. We show that even for very simple settings such algorithms are very unlikely to exist, as for certain wdPTs without projection consisting of only two nodes, containing one respectively two binary atoms. In the presence of projection, this already holds for wdPTs consisting of two nodes with one atom each.

Besides closing the open case in Table 1, future work also includes the search for further tractable fragments. E.g., we do not know any interesting tractable classes for Max-Enum($\mathcal{C}$) yet.

Acknowledgments. This work was supported by the Vienna Science and Technology Fund (WWTF) through project ICT12-015 and by the Austrian Science Fund (FWF): P25207-N23. Markus Kröll was also supported by FWF project W1255-N23.

References

1. S. Ahmetaj, W. Fischl, R. Pichler, M. Simkus, and S. Skritek. Towards reconciling SPARQL and certain answers. In *Proc. WWW 2015*, pages 23–33. ACM, 2015.
2. G. Bagan, A. Durand, and E. Grandjean. On acyclic conjunctive queries and constant delay enumeration. In *Computer Science Logic*, pages 208–222. Springer, 2007.
3. P. Barceló, R. Pichler, and S. Skritek. Efficient evaluation and approximation of well-designed pattern trees. In *Proc. PODS 2015*, pages 131–144. ACM, 2015.
4. S. Cohen, I. Fadida, Y. Kanza, B. Kimelfeld, and Y. Sagiv. Full disjunctions: Polynomial-delay iterators in action. In *Proc. VLDB 2006*, pages 739–750. ACM, 2006.
5. A. Durand, N. Schweikardt, and L. Segoufin. Enumerating answers to FO queries over databases of low degree. In *Proc. PODS 2014*, pages 121–131. ACM, 2014.
6. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.
7. B. Kimelfeld and P. G. Kolaitis. The complexity of mining maximal frequent subgraphs. *ACM Trans. Database Syst.*, 39(4):32:1–32:33, 2014.
8. M. Kröll, R. Pichler, and S. Skritek. On the complexity of enumerating the answers to well-designed pattern trees. In *Proc. ICDT 2016*, volume 48 of *LIPIcs*, pages 22:1–22:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
9. A. Letelier, J. Pérez, R. Pichler, and S. Skritek. Static analysis and optimization of semantic web queries. *ACM Trans. Database Syst.*, 38(4):25, 2013.
10. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009.
11. L. Segoufin. Enumerating with constant delay the answers to a query. In *Proc. ICDT 2013*, pages 10–20. ACM, 2013.