Efficient Evaluation of Well-designed Pattern Trees (Extended Abstract)*

Pablo Barceló[1], Reinhard Pichler[2], and Sebastian Skritek[2]

[1] Center for Semantic Web Research & Department of Computer Science, University of Chile
[2] Faculty of Informatics, Vienna University of Technology

Abstract. Conjunctive queries (CQs) constitute the core of the query languages for relational databases and also the most intensively studied querying mechanism in the database theory community. But CQs suffer from a serious drawback when dealing with incomplete information: If it is not possible to match the complete query with the data, they return no answer at all. The semantic web therefore provides a formalism - known as well-designed pattern trees (WDPTs) - that tackles this problem. In particular, WDPTs allow us to match patterns over the data if available, but do not fail to give an answer otherwise. Here, we abstract away the specifics of semantic web applications and study WDPTs over arbitrary relational schemas. Since our language properly subsumes the class of CQs, the evaluation problem associated with it is intractable. In this paper we identify natural structural properties of WDPTs that lead to tractability of various variants of the evaluation problem.

1 Introduction

Conjunctive queries (CQs) constitute the core of the query languages for relational databases and also the most intensively studied querying mechanism in the database theory community. But CQs suffer from a serious drawback when dealing with incomplete information: they fail to provide an answer when the pattern described by the query cannot be matched completely into the data.

The semantic web therefore provides formalisms to overcome this problem. One simple such formalism corresponds to the $\{\mathsf{AND},\mathsf{OPT}\}$-fragment of SPARQL – the standard query language for RDF, the semantic web data model. The OPT-operator extends the AND-operator by the possibility to return partial answers. I.e., instead of returning no answer at all if not the complete query can be matched into the data, it allows to match parts of the query. Pérez et al. noticed that a non-constrained interaction of these two operators may lead to undesired behavior [11]. This motivated the definition of a better behaved syntactic restriction of the language, known as *well-designed* $\{\mathsf{AND},\mathsf{OPT}\}$-SPARQL. Queries in this fragment allow for a natural tree representation, called *well-designed pattern trees (WDPTs)* [10]. Here, we abstract away from the specifics of RDF and define WDPTs over arbitrary relational schemas.

Despite the importance of WDPTs, very little is known about some fundamental problems related to them. In particular, no in-depth study has been carried out regarding

* This is an extended abstract of [3]

efficient evaluation of these queries, a problem that permeates the literature on CQs and its extensions [13, 8, 9]. The main goal of this work is to initiate a systematic study of tractable fragments of WDPTs for the different variants of query evaluation that have been studied in the literature. We explain this in more detail below.

2 Well-designed Pattern Trees

We define the class of WDPTs below. Intuitively, the nodes of a WDPT represent CQs (called "basic graph patterns" in the semantic web context) while the nesting of optional matching is represented by the tree structure of a WDPT.

A *well-designed pattern tree* (WDPT) over schema σ is a pair $(T, \lambda, \bar{x})$, such that:

1. T is a rooted tree and λ maps each node t in T to a set of relational atoms over σ.
2. For every variable y mentioned in T, the set of nodes where y occurs is connected.
3. The tuple $\bar{x}$ of distinct variables from T denotes the *free variables* of the WDPT.

We say that $(T, \lambda, \bar{x})$ is *projection-free*, if $\bar{x}$ contains all variables mentioned in T.

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

We define the semantics of WDPTs by naturally extending their interpretation under semantic web vocabularies [10]. The intuition behind the semantics of a WDPT $(T, \lambda, \bar{x})$ is as follows. A mapping h is an answer to (T, λ) over a database $\mathcal{D}$, if it is "maximal" among the mappings that satisfy the patterns $q_{T'}$ defined by the subtrees T' of T. This means, h is a solution to $q_{T'}$ and there is no way to "extend" h to a solution of some $q_{T''}$ for some bigger subtree T'' of T. The evaluation of a WDPT $(T, \lambda, \bar{x})$ over $\mathcal{D}$ corresponds then to the projection over the variables in $\bar{x}$ of the mappings h that satisfy (T, λ) over $\mathcal{D}$. We formalize this next: Let $\mathcal{D}$ be a database and $p = (T, \lambda, \bar{x})$ a WDPT over schema σ. Assume that $dom(\mathcal{D})$ is the set of elements in the active domain of $\mathcal{D}$ and $\mathbf{X}$ are 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 $p = (T, \lambda, \bar{x})$ to $\mathcal{D}$ we denote by $h_{\bar{x}}$ the restriction of h to the variables in $\bar{x}$. The *evaluation* of p over $\mathcal{D}$, denoted $p(\mathcal{D})$, corresponds to all mappings of the form $h_{\bar{x}}$, such that h is a maximal homomorphism from p to $\mathcal{D}$.

Notice that WDPTs properly extend CQs. In fact, assume $q(\bar{x})$ is a CQ of the form $\text{Ans}(\bar{x}) \leftarrow R_1(\bar{v}_1), \ldots, R_m(\bar{v}_m)$. Then $q(\bar{x})$ is equivalent to WDPT $p = (T, \lambda, \bar{x})$, where T consists of a single node r and $\lambda(r) = \{R_1(\bar{v}_1), \ldots, R_m(\bar{v}_m)\}$. In other words, $q(\mathcal{D}) = p(\mathcal{D})$, for each database $\mathcal{D}$.

3 Efficient evaluation of WDPTs

3.1 Evaluation of WDPTs:

We study the complexity of the evaluation problem $\text{EVAL}(\mathcal{C})$ for different classes $\mathcal{C}$ of WDPTs. This problem is formally defined as follows: Given a database $\mathcal{D}$ and a WDPT p over σ, as well as a partial mapping $h : \mathbf{X} \to dom(\mathcal{D})$, where $\mathbf{X}$ is the set of variables mentioned in p, is it the case that h belongs to $p(\mathcal{D})$?

The complexity of $\text{EVAL}(\mathcal{C})$ has been studied for the case when $\mathcal{C}$ is the class $\mathcal{C}_{\mathsf{all}}$ of all WDPTs or the class $\mathcal{C}_{\mathsf{pf}}$ of projection-free WDPTs. In particular, $\text{EVAL}(\mathcal{C}_{\mathsf{all}})$ is Σ_2^P-complete [10] and $\text{EVAL}(\mathcal{C}_{\mathsf{pf}})$ is coNP-complete [11]. This raises the need for understanding which classes of WDPTs can be evaluated in polynomial time.

Evaluation of WDPTs is defined in terms of CQ evaluation, which is an intractable problem in general. Therefore, our goal of identifying tractable classes of WDPTs naturally calls for a restriction of the classes of CQ patterns allowed in them. In particular, there has been a flurry of activity around the topic of determining which classes of CQs admit efficient evaluation that could be reused in our scenario [13, 8, 9]. These include classes of bounded *treewidth* [6], *hypertreewidth* [9], etc. We concentrate on the first two. From now on, we denote by $\mathsf{TW}(k)$ (resp., $\mathsf{HW}(k)$), for $k \geq 1$, the class of CQs of treewidth (resp., hypertreewidth) at most k.

A condition that has been shown to help identifying relevant tractable fragments of WDPTs is *local tractability* [10]. This refers to restricting the CQ defined by each node in a WDPT to belong to a tractable class. Formally, let $\mathcal{C}$ be either $\mathsf{TW}(k)$ or $\mathsf{HW}(k)$, for $k \geq 1$. A WDPT $(T, \lambda, \bar{x})$ is *locally in* $\mathcal{C}$, if for each node $t \in T$ such that $\lambda(t) = \{R_1(\bar{v}_1), \ldots, R_m(\bar{v}_m)\}$ the CQ $\text{Ans}() \leftarrow R_1(\bar{v}_1), \ldots, R_m(\bar{v}_m)$ is in $\mathcal{C}$. We write ℓ-$\mathcal{C}$ for the set of all WDPTs that are locally in $\mathcal{C}$.

It is known that local tractability leads to tractability of evaluation for projection-free WDPTs [10]. On the other hand, this result does not hold in the presence of projection, even when $\mathcal{C}$ is of bounded treewidth. Formally, EVAL*(*ℓ-$\mathsf{TW}(k)$*)* and EVAL*(*ℓ-$\mathsf{HW}(k)$*)* are NP-complete for every $k \geq 1$ [10].

This raises the question of which further restrictions on WDPTs are needed to achieve tractability. Here we identify a natural such restriction, called *bounded interface*. Intuitively, this restricts the number of variables shared between a node in a WDPT and its children. Formally, a WDPT $(T, \lambda, \bar{x})$ has *c-bounded interface*, for $c \geq 1$, if for each node $t \in T$ with children $t_1, \ldots, t_k$ it is the case that the number of variables that appear both in a relational atom in $\lambda(t)$ and in a relational atom in $\lambda(t_i)$, for some $1 \leq i \leq k$, is at most c. We denote by $\mathsf{BI}(c)$ the set of WDPTs of c-bounded interface. Interestingly, similar restrictions on the number of variables shared by different atoms of CQs have been recently applied for obtaining reasonable bounds for the problem of containment of Datalog into unions of CQs [5]. One of our main results states that local tractability and bounded interface yield tractability of WDPT evaluation:

Theorem 1. *Let $\mathcal{C}$ be $\mathsf{TW}(k)$ or $\mathsf{HW}(k)$ and $c \geq 1$. Then* $\text{EVAL}(\ell\text{-}\mathcal{C} \cap \mathsf{BI}(c))$ *is in* PTIME.

Notice that CQs are a special case of WDPTs consisting of the root node only. Hence, $\mathsf{TW}(k) \subseteq \ell\text{-}\mathsf{TW}(k) \cap \mathsf{BI}(c)$ and $\mathsf{HW}(k) \subseteq \ell\text{-}\mathsf{HW}(k) \cap \mathsf{BI}(c)$ hold for each $c \geq 1$.

Therefore, Theorem 1 tells us that ℓ-$\mathsf{TW}(k) \cap \mathsf{BI}(c)$ and ℓ-$\mathsf{HW}(k) \cap \mathsf{BI}(c)$ define relevant extensions of $\mathsf{TW}(k)$ and $\mathsf{HW}(k)$, respectively, that preserve tractability of evaluation.

3.2 Partial evaluation of WDPTs:

Given the nature of WDPTs, it is also interesting to check whether a mapping h is a *partial* answer to the WDPT p over $\mathcal{D}$ [11, 1], i.e., whether h can be extended to some answer h' to p over $\mathcal{D}$. This gives rise to the partial evaluation problem PARTIAL-EVAL($\mathcal{C}$) for a class $\mathcal{C}$ of WDPTs defined as follows: Given a database $\mathcal{D}$ and a WDPT $p \in \mathcal{C}$ over σ, as well as a partial mapping $h : \mathbf{X} \to \mathbf{U}$, where $\mathbf{X}$ is the set of variables mentioned in p, does there exists some $h' \in p(\mathcal{D})$ such that h' extends h?

If projection is allowed, then partial evaluation is NP-complete even under local tractability, i.e., even for the classes ℓ-$\mathsf{TW}(k)$ and ℓ-$\mathsf{HW}(k)$, for each $k \geq 1$ [10].

It is easy to modify the proof of Theorem 1 to show that adding bounded interface to local tractability yields efficient partial evaluation; that is, PARTIAL-EVAL(ℓ-$\mathsf{TW}(k) \cap \mathsf{BI}(c)$) and PARTIAL-EVAL($\ell$-$\mathsf{HW}(k) \cap \mathsf{BI}(c)$) are in PTIME. However, partial evaluation is seemingly easier than exact evaluation. Hence, the question naturally arises if tractability of partial evaluation of WDPTs can be ensured by a weaker condition. Indeed, we give a positive answer to this question below. This condition will be referred to as *global tractability*. Intuitively, it states that there is a bound on the treewidth (resp., hypertreewidth) of the CQs defined by the different subtrees of a WDPT $(T, \lambda, \bar{x})$ rooted in r. Formally, let $\mathcal{C}$ be $\mathsf{TW}(k)$ or $\mathsf{HW}(k)$, for $k \geq 1$. A WDPT $(T, \lambda, \bar{x})$ is *globally in* $\mathcal{C}$, if for each subtree T' of T rooted in r it is the case that the CQ $q_{T'}$ is in $\mathcal{C}$. We denote by g-$\mathcal{C}$ the set of all WDPTs that are globally in $\mathcal{C}$.

The following proposition formally states that global tractability is a strictly weaker condition than the conjunction of local tractability and bounded interface.

Proposition 1. *1. Let $k, c \geq 1$. Then ℓ-$\mathsf{TW}(k) \cap \mathsf{BI}(c) \subseteq g$-$\mathsf{TW}(k + 2c)$ and ℓ-$\mathsf{HW}(k) \cap \mathsf{BI}(c) \subseteq g$-$\mathsf{HW}(k + 2c)$.*
2. For every $k \geq 1$ there is a family C_k of WDPTs in g-$\mathsf{TW}(k)$ (resp., in g-$\mathsf{HW}(k)$) such that $C_k \not\subseteq \mathsf{BI}(c)$, for each $c \geq 1$.

We can now formally state that global tractability leads to tractability of the partial evaluation problem for WDPTs:

Theorem 2. PARTIAL-EVAL(g-$\mathsf{TW}(k)$) *and* PARTIAL-EVAL(g-$\mathsf{HW}(k)$) *are in* PTIME *for every $k \geq 1$.*

It remains to answer the question if global tractability also suffices to ensure tractability of (exact) evaluation for WDPTs. It turns out that this is not the case.

Proposition 2. EVAL(g-$\mathsf{TW}(k)$) *and* EVAL(g-$\mathsf{HW}(k)$) *are* NP-*complete for all $k \geq 1$.*

3.3 Semantics based on maximal mappings:

The semantics of projection-free WDPTs is only based on *maximal* mappings, i.e., mappings that are not subsumed by any other mapping in the answer. This is no longer the case in the presence of projection [10]. Recent work on query answering for SPARQL under entailment regimes has established the need for a semantics for WDPTs that is uniquely based on maximal mappings [1]. This semantics is formalized as follows. Assume $\mathcal{D}$ is a database and p is a WDPT over σ. The *evaluation of p over $\mathcal{D}$ under maximal mappings*, denoted $p_m(\mathcal{D})$, corresponds to the restriction of $p(\mathcal{D})$ to those mappings $h \in p(\mathcal{D})$ that are not extended by any other mapping $h' \in p(\mathcal{D})$. This naturally leads to the decision problem MAX-EVAL($\mathcal{C}$) defined as follows: Given a database $\mathcal{D}$ and a WDPT $p \in \mathcal{C}$ over σ, as well as a partial mapping $h : \mathbf{X} \to \mathbf{U}$, where $\mathbf{X}$ is the set of variables mentioned in p, is $h \in p_m(\mathcal{D})$?

It follows from [1] that MAX-EVAL($\mathcal{C}$) is clearly intractable for the class $\mathcal{C}$ of all WDPTs. Analogously to PARTIAL-EVAL, local tractability is not sufficient to ensure tractability of MAX-EVAL:

Proposition 3. *For every $k \geq 1$ the problems* MAX-EVAL(ℓ-$\mathsf{TW}(k)$) *and* MAX-EVAL(ℓ-$\mathsf{HW}(k)$) *are* NP*-hard.*

To obtain tractability in this case it is however sufficient to impose global tractability, which is exactly the same condition that yields tractability of partial evaluation for WDPTs (as stated in Theorem 2):

Theorem 3. MAX-EVAL(g-$\mathsf{TW}(k)$) *and* MAX-EVAL(g-$\mathsf{HW}(k)$) *are in* PTIME *for every $k \geq 1$.*

4 Further Results

Taking these results as a starting point, we were also able to show that in several cases the complexity of static analysis tasks, like deciding containment and equivalence [12], decreases. Next, we also studied the problem of testing if some WDPT is equivalent to one from a tractable class (cf. e.g. [2, 4, 7]), and showed that evaluating such queries is fixed-parameter tractable w.r.t. the size of the query. Finally, we also studied the problem of approximating WDPTs by one from a tractable class (a problem that is now well-understood in the context of CQs [2]).

Acknowledgments. The work of Pablo Barceló is funded by the Millenium Nucleus Center for Semantic Web Research under grant NC120004. Part of this work was done while Reinhard Pichler and Sebastian Skritek were visiting Pablo Barceló on invitation by the Millenium Nucleus Center for Semantic Web Research. Reinhard Pichler and Sebastian Skritek were supported by the Vienna Science and Technology Fund (WWTF) through project ICT12-015 and by the Austrian Science Fund (FWF):P25207-N23.

References

1. S. Ahmetaj, W. Fischl, R. Pichler, M. Simkus, and S. Skritek. Towards reconciling sparql and certain answers. In *WWW'15*, 2015. Accepted for publication.
2. P. Barceló, L. Libkin, and M. Romero. Efficient approximations of conjunctive queries. *SIAM J. Comput.*, 43(3):1085–1130, 2014.

3. P. Barcelo, R. Pichler, and S. Skritek. Efficient evaluation and approximation of well-designed pattern trees. In *PODS'15*, 2015. Accepted for publication.
4. P. Barceló, M. Romero, and M. Y. Vardi. Semantic acyclicity on graph databases. In *PODS'13*, pages 237–248, 2013.
5. P. Barceló, M. Romero, and M. Y. Vardi. Does query evaluation tractability help query containment? In *PODS'14*, pages 188–199, 2014.
6. C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. *Theor. Comput. Sci.*, 239(2):211–229, 2000.
7. V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *CP'02*, pages 310–326, 2002.
8. G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. *J. ACM*, 48(3):431–498, 2001.
9. G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002.
10. 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.
11. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009.
12. R. Pichler and S. Skritek. Containment and equivalence of well-designed SPARQL. In *PODS'14*, pages 39–50, 2014.
13. M. Yannakakis. Algorithms for acyclic database schemes. In *VLDB'81*, pages 82–94, 1981.