Counting Answers to Existential Questions

Holger Dell

Cluster of Excellence (MMCI), Saarland Informatics Campus (SIC), Saarbrücken, Germany
hdell@mmci.uni-saarland.de

Marc Roth

Cluster of Excellence (MMCI), Saarland Informatics Campus (SIC), Saarbrücken, Germany
mroth@mmci.uni-saarland.de

Philip Wellnitz

Max Planck Institute for Informatics, Saarland Informatics Campus (SIC), Saarbrücken, Germany
wellnitz@mpi-inf.mpg.de

Abstract

Conjunctive queries select and are expected to return certain tuples from a relational database. We study the potentially easier problem of *counting* all selected tuples, rather than enumerating them. In particular, we are interested in the problem's parameterized and data complexity, where the query is considered to be small or even fixed, and the database is considered to be large. We identify two structural parameters for conjunctive queries that capture their inherent complexity: The dominating star size and the linked matching number. If the *dominating star size* of a conjunctive query is large, then we show that counting solution tuples to the query is at least as hard as counting dominating sets, which yields a fine-grained complexity lower bound under the Strong Exponential Time Hypothesis (SETH) as well as a #W[2]-hardness result in parameterized complexity. Moreover, if the *linked matching number* of a conjunctive query is large, then we show that the structure of the query is so rich that arbitrary queries up to a certain size can be encoded into it; in the language of parameterized complexity, this essentially establishes a #A[2]-completeness result.

Using ideas stemming from Lovász (1967), we lift complexity results from the class of conjunctive queries to arbitrary existential or universal formulas that might contain inequalities and negations on constraints over the free variables. As a consequence, we obtain a complexity classification that refines and generalizes previous results of Chen, Durand, and Mengel (ToCS 2015; ICDT 2015; PODS 2016) for conjunctive queries and of Curticapean and Marx (FOCS 2014) for the subgraph counting problem. Our proof also relies on graph minors, and we show a strengthening of the Excluded-Grid-Theorem which might be of independent interest: If the linked matching number (and thus the treewidth) is large, then not only can we find a large grid somewhere in the graph, but we can find a large grid whose diagonal has disjoint paths leading into an assumed node-well-linked set.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Problems, reductions and completeness

Keywords and phrases Conjunctive queries, graph homomorphisms, counting complexity, parameterized complexity, fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.113

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version A full version of this work appears as preprint arXiv:1902.04960.

Funding *Philip Wellnitz*: Partially funded by the Saarbrücken Graduate School of Computer Science.

Acknowledgements We thank Cornelius Brand, Karl Bringmann, Radu Curticapean, Reinhard Diestel, Joshua Erde, Stephan Kreutzer, Stefan Mengel, Daniel Weißauer, and an anonymous reviewer for discussions and advice.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 113; pp. 113:1–113:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Conjunctive query evaluation is a core problem in database theory. Using first-order logic, conjunctive queries can be expressed by formulas of the form

$$x_1 \dots x_k \exists y_1 \dots \exists y_\ell (a_1 \wedge \dots \wedge a_m) \,, \tag{1}$$

where the x_i are the *free variables*, the y_i are the (existentially) *quantified variables*, and the a_i are *atomic formulas* (such as edge $E(x_1, y_4)$ or relational $R(x_7, y_3, y_6)$ constraints on the variables). Conjunctive queries exactly correspond to select-project-join queries; a detailed introduction can be found in the textbook of Abiteboul, Hull, and Vianu [1]. The *conjunctive query evaluation problem* is given a conjunctive query and a relational database, and is tasked to compute the set of all assignments to the free variables such that the formula is satisfied. Since enumerating all solution tuples $s_1 \dots s_k$ can be costly for reasons not inherent to the problem's complexity, it is more meaningful to consider the decision problem (Does there exist a solution tuple?) or the more general counting problem (How many solution tuples exist?). The decision problem is equivalent to setting $k = 0$ and also called the constraint satisfaction problem (CSP). In this paper, we study the problem of counting the number of all solution tuples for conjunctive and more general queries.

Perhaps the most naïve way to study the complexity of this problem is via its *combined complexity*, in which both the query and the database are considered to be worst-case inputs. Since conjunctive queries generalize the clique problem on graphs, the problem is clearly NP-hard in this setting [2]. In the real world, however, the database is much larger than the query, and thus the combined complexity may fixate on instances that we do not care about. Instead, we consider two other models in this paper: the data complexity and the parameterized complexity of conjunctive query evaluation.

The *data complexity* considers the query to be completely fixed and only the database to be worst-case input. If the query is fixed, the number of variables $k + \ell$ is a constant, and so the problem is polynomial-time solvable: even the exhaustive search algorithm just needs to try out and check all $n^{k+\ell}$ possible assignments to the variables, where n is the size of the universe. Unsurprisingly, exhaustive search is not the best strategy for every query. For example, Chekuri and Rajaraman [3] showed that the decision and counting problems can be solved in time $O(n^{t+1})$ where t is the treewidth of the query's Gaifman graph, that is, the graph containing a vertex for every variable and an edge between two vertices whenever the corresponding variables are contained in a common constraint. Since $t + 1$ is typically much smaller than $k + \ell$, this algorithm is better than exhaustive search. For each fixed query Q, the guiding question for a fine-grained understanding of data complexity is this: What is the smallest constant c_Q such that the query evaluation problem can be solved in time $O(n^{c_Q})$?

Parameterized complexity offers a third vantage point from which conjunctive query evaluation can be studied. Here the query isn't completely fixed, but it's also not completely free either. Instead, it is assumed that only certain types of queries will be used, meaning that the class of queries that are allowed as input is restricted. As a hybrid between data complexity and combined complexity, the parameterized complexity of query evaluation is more difficult to study than the combined complexity, but easier than the data complexity, while still offering some insight. For example, Grohe, Schwentick, and Segoufin [18] show that the conjunctive query evaluation problem is W[1]-hard if the class of allowed input queries has Gaifman graphs of unbounded treewidth.

1.1 Context and Previous Work

When only one constraint type E of arity two is allowed, the conjunctive query evaluation problem specializes to the graph homomorphism problem: The decision problem (where $k = 0$) is given two graphs H, G to decide whether there is a homomorphism from H to G. Dalmau et al. [9] prove that this problem can be solved in polynomial time if the homomorphic core of H has bounded treewidth, and conversely, Grohe [17] shows that the graph homomorphism problem is W[1]-complete even if H is restricted to be from an arbitrary class of graphs whose homomorphic cores have unbounded treewidth. Taken together, these two results yield a *dichotomy theorem* for the complexity of detecting graph homomorphisms: Depending on the class of allowed graphs H, the problem is either polynomial-time computable or W[1]-complete, and in particular there are not infinitely many cases of intermediate complexity. For the counting problem without quantified variables (where $\ell = 0$), such a dichotomy is also known: Dalmau and Jonsson [8] show that the number of homomorphisms from H to G is polynomial-time computable if H itself has bounded treewidth, and it is #W[1]-complete if H comes from any class of unbounded treewidth. In the mixed situation when both free and quantified variables may exist (and thus $k, \ell > 0$), then the resulting counting problem actually counts *partial homomorphisms*, that is, homomorphisms from k vertices of H that can be extended to a homomorphism on all $k + \ell$ vertices of H. A line of work [27, 25], culminating in Durand and Mengel [12] and Chen and Mengel [4], studies the parameterized complexity of this mixed problem, and depending on the class of graphs H that are allowed, they classify the complexity either as polynomial-time, W[1]-equivalent, or #W[1]-hard. A corollary to the present work is a finer classification that splits up the #W[1]-hard cases into three classes.

One way to go beyond homomorphisms is to consider *injective* homomorphisms, which leads to the corresponding decision problem that is given H, G to decide whether H is a subgraph of G – this problem can be solved in time $f(H)n^{O(t)}$ if t is the treewidth of H (e.g., [15]), that is, it is *fixed-parameter tractable* when parameterized by $|H|$ and if the treewidth is bounded. However, it is an important open problem [24] whether the subgraph detection problem is W[1]-hard when H is restricted to be from an arbitrary class of unbounded treewidth. The counting problem is better understood: Vassilevska Williams and Williams [33] (also cf. [21, 7]) show that the number of times H occurs as a subgraph in G can be computed in time $f(H)n^{\mathsf{vc}(H)+O(1)}$ where $\mathsf{vc}(H)$ is the size of the smallest vertex cover, but Curticapean and Marx [7] (also cf. [6]) show that the problem is #W[1]-complete if H is from any class of graphs whose minimum vertex cover is not bounded. Now, what do injective homomorphisms have to do with conjunctive queries? As it turns out, what we are doing is to add *inequalities* as an additional, but very restricted constraint type: Injective homomorphisms correspond to queries without quantified variables that have edge constraints and are augmented with inequalities $(x_i \neq x_j)$ for all distinct i, j. If some, but not all, inequality constraints are present, we obtain partially injective homomorphisms, the complexity of which has a known dichotomy theorem for the counting version [30], and has been studied to some extent for the decision version [20]. As part of the present work, we are able to classify the mixed situation with free and quantified variables $(k, \ell > 0)$ *as well as* some inequalities on the free variables.

The mentioned complexity classification for counting partial homomorphisms into three cases [12, 4] was actually proved in the more general setting of conjunctive queries. Chen and Mengel [5] extended their classification to queries that are monotone, but not necessarily conjunctive. That is, the corresponding formula is supposed to be an existential positive formula, which may contain existential quantifiers ∃, logical ands ∧, and ors ∨.

In the present work, we are able to further extend (our finer version of) the classification to existential formulas that may have negations on constraints involving only free variables; we truly study the complexity of *counting answers to existential questions.*

1.2 Our Contributions

As already indicated in Section 1.1, we make simultaneous progress on two fronts: Our complexity classifications are finer than previous work, and we can prove the classification for more general classes of queries. An important feature of our work is that the proofs are modular and largely self-contained: We first prove the complexity results for counting partial homomorphisms, then lift them to conjunctive queries, and then further to a more general class of queries. So what is the most general class of queries that we study? We allow queries φ of the form

$$x_1 \dots x_k \exists y_1 \dots \exists y_\ell : \psi \,, \tag{2}$$

where ψ is a quantifier-free formula in first-order logic and all negations in ψ must be directly applied to constraints that only involve free variables (e.g. $E(x_1, x_7) \vee (R(x_7, y_7, y_9) \wedge \neg R(x_1, x_4, x_9)))$. Constraints of the form $\neg R(x_1, x_4, x_9)$ are referred to as *non-monotone constraints* in the remainder of the paper. Furthermore φ may be equipped with a set of inequalities over the free variables (eg. $x_3 \neq x_5$).

All of our theorems also apply to the corresponding *universal* queries, where each $\exists$ in (2) is replaced with $\forall$, but for the sake of readability we will often omit this fact. We are able to generalize from conjunctive queries to queries of the form (2) by using ideas that go back to Lovász's work from 1967 [22] (also cf. [23]): We prove that queries φ of the form (2) can be expressed in a meaningful way as an abstract linear combination of conjunctive queries (which are of the form (1)); positive results (algorithms) as well as negative results (hardness) for each "summand" translate to the abstract linear combination and thus to φ.

Data Complexity

To study the data complexity of the problem, we employ the Strong Exponential Time Hypothesis (SETH) by Impagliazzo and Paturi [19], which was developed in the context of fine-grained complexity. The k-dominating set problem can be easily expressed as a (universal) conjunctive query, and Williams and Pătraşcu [28] show that this problem cannot be solved in time $O(n^{k-\varepsilon})$ unless SETH is false. We are able to lift this hardness result to all queries φ that have the k-dominating set query as a *query minor*, a notion that we translate from graphs and formalize later. The *dominating star size* $\mathsf{dss}(\varphi)$ of a conjunctive query φ is the maximum number k such that the k-dominating set query is a query minor. Equivalently, this means that some connected component in the quantified variables of φ has k neighbors in the free variables.[1] We obtain the following result:

▶ **Theorem 1.** *Let φ be a fixed query of the form* (2)*. Given a logical structure B with a domain of size n, we wish to compute the number of solutions of φ in B. If SETH holds, this problem cannot be solved in time $O(n^{\mathsf{dss}(\varphi)-\varepsilon})$ for any $\varepsilon > 0$.*

In the full version, we also obtain an algorithm for the problem in Theorem 1, with a running time of $O(n^{\mathsf{dss}(\varphi)+t+1} + n^{t'+1})$, where t and t' are treewidths related to the query φ. Neglecting many technical details, the proof of Theorem 1 reduces the k-dominating set

[1] The dominating star size coincides with the *strict star size* from [4].

problem to the model counting problem for φ by following operations of the query minor. If φ is a query of the form (2), then it can be represented by an abstract linear combination of conjunctive queries φ'; in this case, we define $\mathsf{dss}(\varphi)$ as the maximum $\mathsf{dss}(\varphi')$ over all constituents φ' that occur in this abstract linear combination.

Theorem 1 is similar in spirit to other known conditional lower bounds for first-order model checking, such as the one of Williams [32] and Gao et al. [16]. One of their results is that first-order sentences with $k+1$ variables cannot be decided in time $O(m^{k-\varepsilon})$, where m is the size of the structure, unless SETH fails. However, these results are incomparable to Theorem 1 for several reasons: The results in [32, 16] allow negations and consider the decision problem, while we allow only limited negations and consider the counting problem. More fundamentally, however, Theorem 1 gives a hardness result for every fixed query φ, while the results in [32, 16] show that there exists a query φ that is hard. Moreover, the lower bounds in [32, 16] are in terms of the size m of the structure, not merely the size n of the domain.

Parameterized Complexity

We refine the classification of Chen and Mengel [4] for counting answers to conjunctive queries. For every class of allowed queries they show the problem to be either fixed-parameter tractable, W[1]-equivalent or #W[1]-hard. Here, W[1]-equivalent means that there are parameterized Turing reductions from and to the decision version of the k-Clique problem. Understanding the parameterized complexity of problems even beyond the usual classes W[1] and #W[1] is interesting from a structural complexity point of view, and it also provides meaningful information about the studied problem. Indeed we show that the dominating star size, i.e., the parameter considered in Theorem 1, is a structural parameter for conjunctive queries that, if unbounded, makes the problem #W[2]-hard and that, if bounded, keeps the problem #W[1]-easy.

This extension to #W[2]-hard cases only partially resolves the parameterized complexity of the problem of counting answers to conjunctive queries. It is known that the general problem of counting answers to formulas of the form

$$x_1 \dots x_k \exists y_1 \dots \exists y_\ell : \psi\,, \qquad \text{where } \psi \text{ is a quantifier-free first-order formula,} \tag{3}$$

is #A[2]-equivalent.[2] For which families of *conjunctive* queries is the counting problem as hard as for unrestricted queries as in (3)? Such families have the hardest counting problems, even harder than the #W[2]-hard cases unless #A[2] = #W[2] holds, which seems unlikely.[3] We prove that families of conjunctive queries are #A[2]-hard if their *linked matching number* is unbounded. Intuitively a conjunctive query φ with free variables X and quantified variables Y has a large linked matching if there is a large well-linked set in Y that cannot be separated from X by removing a small number of variables. We obtain the following refined complexity classification.

[2] Due to a technicality in the original definition of #A[2], we cannot establish #A[2]-completeness and will instead only talk about *equivalence* to a #A[2]-complete problem under parameterized Turing reductions.

[3] See Chapt. 8 and 14 in [14] for a discussion.

▶ **Theorem 2.** *Let Φ be a family of conjunctive queries. Given a formula φ from Φ and a logical structure B, we wish to compute the number of solutions of φ in B. When parameterized by $|\varphi|$ this problem is*

1. #W[1]*-easy if the dominating star size of Φ is bounded,*
2. #W[2]*-hard if the dominating star size of Φ is unbounded, and*
3. #A[2]*-equivalent if the linked matching number of Φ is unbounded.*

It is instructive to provide examples for the application of the above theorem. First consider the problem of, given a graph G without self-loops and a natural number k, computing the number of cliques of size k that are not maximal. While the problem of counting cliques of size k is #W[1]-complete, adding the non-maximality constraint makes the problem hard for #W[2]. To see this, we will express the problem as a conjunctive query

$$\varphi_k := x_1 \dots x_k \exists y : \bigwedge_{1 \le i < j \le k} E(x_i, x_j) \wedge \bigwedge_{1 \le i \le k} E(x_i, y)\,. \tag{4}$$

Note that the number of solutions to φ_k in G is precisely $k!$ times the number of non-maximal cliques of size k in G. Furthermore, it holds that φ_k has dominating star size k and hence that $\Phi = \{\varphi_k \mid k \in \mathbb{N}\}$ has unbounded dominating star size. By Theorem 2 the problem of counting answers to queries in Φ is #W[2]-hard. Furthermore, invoking Theorem 1, we obtain that counting non-maximal cliques of size k cannot be done in time $O(n^{k-\varepsilon})$ for any $\varepsilon > 0$. Note that this is also in sharp contrast to the problem of counting (not necessarily non-maximal) cliques of size k which can be done in time $O(n^{\omega k/3})$ [26]. Furthermore deciding the existence of a non-maximal clique of size k is equivalent to deciding the existence of a clique of size $k+1$ and hence the lower bound under SETH crucially depends on the fact that we count the solutions.

On the other hand, counting non-maximal cliques of size k is most likely not #A[2]-hard as it is #W[2]-easy[4]. An example for a #A[2]-hard problem would be the following. Assume a graph G and a natural number k are given. Then the goal is to compute the number of k-vertex sets that can be (perfectly) matched to a k-clique. Let us express the problem as a conjunctive query

$$\psi_k := x_1 \dots x_k \exists y_1 \dots \exists y_k : \bigwedge_{1 \le i < j \le k} E(y_i, y_j) \wedge \bigwedge_{1 \le i \le k} E(x_i, y_i)\,. \tag{5}$$

We point out that ψ_k does not correspond directly to the vertex sets we would like to count as x_i and x_j could be the same vertex in G. However, it can be shown that an oracle for counting answers to ψ_k allows us to compute the desired number efficiently and vice versa. Finally, as the linked matching number of ψ_k is not bounded for $k \to \infty$, #A[2]-hardness follows from Theorem 2.

Building up on Theorem 2 and using the framework of linear combinations, we obtain the following, extensive classification result.

[4] If there is a constant bound on the number of quantified variables then the problem of counting answers to conjunctive queries is reducible to a #W[2]-complete problem w.r.t. parameterized Turing reductions. We omit a proof of this statement but point out that it can be done by lifting the results of Chapt. 7.4 in [14] to the realm of counting problems.

▶ **Theorem 3.** *Let Φ be a family of existential or universal positive formulas with inequalities and non-monotone constraints, both over the free variables. Given a formula φ from Φ and a logical structure B, we wish to compute the number of solutions of φ in B.*

When parameterized by $|\varphi|$, this problem is either fixed parameter tractable, W[1]*-equivalent,* #W[1]*-equivalent,* #W[2]*-hard or* #A[2]*-equivalent.*

Note that allowing the inequalities and non-monotone constraints over all variables, not just the free ones, would in particular include the subgraph decision problem. However, the parameterized complexity of finding a subgraph in G that is isomorphic to a small pattern graph P is a long-standing open question in parameterized complexity [11, Chapt. 33.1].

1.3 Techniques and Overview

Our paper brings together questions and techniques from a wide variety of areas, such as parameterized and fine-grained complexity, logics, database theory, matroid theory, lattice theory, graph minor theory, and the theory of graph limits. The interested reader should not be alarmed, however, as we put considerable effort into making the presentation as self-contained and smooth as possible, introducing the required background material carefully and only once needed in both, the extended abstract as well as in the full version this paper: After reviewing some basic preliminaries (Section 2), we formally present our refined complexity classifications in Section 3 for the special case of partial graph homomorphisms, rather than the full query evaluation problem. Due to the incompatibility of the space constraints and the amount of results and techniques required, we deferred all proofs as well as the treatment of the full query evaluation problem over arbitrary signatures of bounded arity to the full version of this paper.

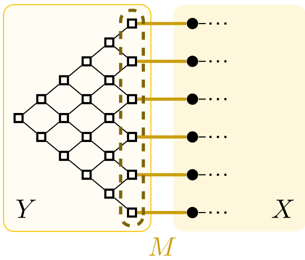
Colors and Query Minors

We will mainly work with a color-prescribed variant of the problem of counting answers to conjunctive queries. Here we assume that the elements of a given database B are colored according to the variables of the given conjunctive query φ and the goal is to compute the number of solutions that are additionally color-preserving. For this variant we will show and heavily exploit that the problem of counting answers to a conjunctive query φ is at least as hard as counting answers to any query that is a minor of φ. Minors of a query are defined via the (graph theoretic) minors of its Gaifman graph. It is then required to show that the color-prescribed variant and the uncolored variant are interreducible for all minimal queries. Intuitively, a query is minimal if it does not contain a proper subquery that produces the same set of solutions for each database. The proof of the interreducibility relies on the theory of homomorphic equivalence.

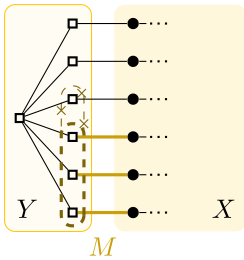
For Theorem 1 and the second case of Theorem 2 we construct a tight reduction from the problem of counting dominating sets of size k which cannot be solved in time $O(n^{k-\varepsilon})$ for any $\varepsilon > 0$ unless SETH fails [28] and which is hard for #W[2] [13].

Minor Theory

For #A[2]-hardness in Theorem 2 we take a detour to graph minor theory: Given a graph G, we call a set $S \subseteq V(G)$ *node-well-linked* if, for every pair of disjoint, equal-sized subsets A, B of S, there are $|A| = |B|$ vertex disjoint paths in G that connect the vertices in A with the vertices in B. Now, we obtain the following strengthening of the Excluded-Grid-Theorem, which might be of independent interest.



(a) A query (H, X) with a large linked matching number ($\mathsf{lmn}(H, X) = 6$): There is a large matching M (gold-colored) connecting vertices in X with a node-well-linked set in Y (enclosed within the dashed line).



(b) A query (H, X) with a small linked matching number ($\mathsf{lmn}(H, X) = 3$): While there may be larger matchings between vertices of X and Y, a largest matching M into a node-well-linked set in Y has size 3.

Figure 1 Examples for queries with large and small linked matching number.

▶ **Theorem 4** (Intuitive version). *There exists an unbounded function f such that every graph with a node-well-linked set S of k vertices has an $(f(k) \times f(k))$-grid minor with the property that there are $f(k)$ vertex-disjoint paths leading from S to the $f(k)$ vertices of the first column of the grid and without touching the grid minor elsewhere.*

If we drop the requirement that the minor model can be reached by disjoint paths from S, then this theorem is well-known and due to Diestel et al. [10].

Intuitively, we use Theorem 4 in the following way: If the quantified variables of a query contain a node-well-linked set S, we obtain a large grid-like structure that is connected to S by many vertex-disjoint paths. Next, we show that if that set S also has a large matching to a subset of the free variables of the query, then the query becomes #A[2]-hard. For this last step, we use an #A[2]-normalization theorem, which we will provide at the end of the paper.

Formally, we define the *linked matching number* of a query and prove #A[2]-hardness if this parameter is unbounded. Consider Figure 1 for examples for the linked matching number.

▶ **Definition 5** (Linked matching number). *Let (H, X) be graphical conjunctive query, let $Y = V(H) \setminus X$ be the set of quantified variables, and let M be a matching from X to Y. We call the matching M* linked *if the set $V(M) \cap Y$ is node-well-linked in $H[Y]$. The* linked matching number lmn *of (H, X) is defined as the size of the largest linked matching of H.*

Abstract Linear Combinations

To prove Theorem 3, we use abstract linear combinations that are called *quantum graphs* (or rather, quantum queries in our setting) and were developed in the theory of graph limits [23]. For our computational questions, the *complexity monotonicity property* [6] is the useful phenomenon that the quantum graph and its constituents (i.e., its abstract summands) often lead to computational problems that have precisely the same complexity. Using elementary linear-algebraic and polynomial interpolation arguments, we prove that this property holds, and we use Rota's NBC Theorem from lattice theory [29] to determine which graphs are constituents of the relevant quantum graphs. The complexity monotonicity property has been used (implicitly) by Chen and Mengel [5] for their extension from conjunctive queries to monotone queries; and the extension from homomorphisms to partially injective homomorphism [30] used Rota's Theorem in a similar fashion as we do in the present work.

2 Preliminaries

We use the notation $[n] = \{1, \dots, n\}$ and $[m, n] = \{m, \dots, n\}$ for natural numbers with $m < n$. We write $\#M$ for the cardinality of a finite set M. We write $f|_M$ for the restriction of a function f to elements of M. For a function $f\colon A \times B \to C$ and $a \in A$, we write $f(a, \star)$ for the function $b \mapsto f(a, b)$.

Graphs, Homomorphisms, and Formulas

Graphs in this paper are unlabeled, undirected, simple and without self-loops, unless stated otherwise. Let $V(G)$ denote the set of vertices and $E(G)$ denote the set of edges of G. We define the *size* of a graph G to be the number of vertices. Given a subset Y of $V(G)$, we write $G[Y]$ for the subgraph induced by the vertices of Y. The *complement graph* $\overline{G}$ has the same vertices as G and contains an edge uv if and only if $u \neq v$ and $uv \notin E(G)$. A *homomorphism* h from a graph F to a graph G is a mapping from $V(F)$ to $V(G)$ that is edge-preserving, that is, all $uv \in E(F)$ satisfy $h(u)h(v) \in E(G)$. We write $\mathsf{Hom}(F \to G)$ for the set of all homomorphisms from F to G. A bijective homomorphism whose inverse is also a homomorphism is called an *isomorphism*, and a homomorphism from F to F itself is called *endomorphism*. An endomorphism that is also an isomorphism is called an *automorphism*. We write $\mathsf{Aut}(F)$ for the set of all automorphisms of F.

Parameterized Counting Complexity

A *counting problem* is a function $P\colon \{0,1\}^* \to \mathbb{N}$, and a *parameterized counting problem* is a pair (P, π) where $\pi\colon \{0,1\}^* \to \mathbb{N}$ is computable and called a *parameterization*. Parameterized *decision* problems are defined likewise for decision problems $P\colon \{0,1\}^* \to \{0,1\}$. A parameterized (decision or counting) problem is *fixed-parameter tractable* if there is a computable function $t\colon \mathbb{N} \to \mathbb{N}$ such that, for every input $x \in \{0,1\}^*$, the function P can be computed in time $t(\pi(x)) \cdot \mathsf{poly}(|x|)$. We denote the class of all fixed-parameter tractable problems as FPT.

A *parameterized Turing-reduction* from (P, π) to (P', π') is an algorithm $\mathbb{A}$ with oracle access to P' that solves P, such that $\mathbb{A}$ runs in fixed-parameter tractable time when parameterized by π and there exists a computable function r such that, for every input x, the parameter $\pi'(y)$ of every query y is bounded by $r(\pi(x))$. A *parameterized parsimonious reduction* is a parameterized Turing-reduction with the additional requirement that $\mathbb{A}$ is only allowed to query the oracle a single time at the very end of the computation and then outputs the result of the query without further modification.

Clique is the parameterized (decision) problem to decide whether a given graph G contains a k-clique. Similarly, DomSet is to decide whether G has a dominating set of size k. The parameterized counting problems $\#\mathsf{Clique}$ and $\#\mathsf{DomSet}$ count the number of the respective objects. We define the parameterized complexity classes that appear in this paper by their well-known complete problems: W[1] contains all parameterized problems that are reducible to Clique with respect to parameterized parsimonious reductions. Similarly, #W[1], W[2], and #W[2] contain all problems reducible to $\#\mathsf{Clique}$, DomSet, and $\#\mathsf{DomSet}$, respectively. Furthermore #A[2] is the class of all parameterized counting problems that are expressible as model counting problem with one quantifier alternation. It is known that

$$\mathrm{FPT} \leq^T \mathrm{W}[1] \leq^T \#\mathrm{W}[1] \subseteq \#\mathrm{W}[2] \subseteq \#\mathrm{A}[2]\,,$$

where $\mathrm{C} \leq^T \mathrm{D}$ denotes that every problem in C can be reduced to a problem in D with respect to parameterized Turing-reductions.

(a) **(b)**

Figure 2 (a) Graphical representation of the conjunctive query in (6). **(b)** A graphical conjunctive query that is "equivalent" to the example on the left in the sense that an assignment $a\colon \{x_1, \dots, x_k\} \to V(G)$ is a partial homomorphism from the left graph to G if and only if it is a partial homomorphism from the right graph to G.

For further background on parameterized counting complexity, see [14, Chapter 14]. While the parameterized complexity classes are defined via parsimonious reductions, we will rely on Turing reductions. Hence we cannot speak of completeness but instead of equivalence.

▶ **Definition 6.** *Let* C *be a parameterized complexity class. A parameterized counting problem* (P, π) *is* C-easy *if it can be reduced to a problem in* C *and it is* C-hard *if every problem in* C *reduces to* (P, π)*, both with respect to parameterized Turing-reductions. A problem is* C-equivalent *if it is* C-easy *and* C-hard*.*

Exponential-Time Hypotheses

The strong exponential time hypothesis (SETH) asserts that for all $\delta > 0$ there is some $k \in \mathbb{N}$ such that k-SAT cannot be computed in time $O(2^{(1-\delta)n})$, where n is the number of variables of the input formula [19]. A dominating set of size k in an n-vertex graph cannot be computed in time $O(n^{k-\varepsilon})$ for any $\varepsilon > 0$ unless SETH is false [28]. The exponential time hypothesis (ETH) asserts that 3-SAT cannot be computed in time $\exp(o(m))$, where m is the number of clauses of the input formula [19].

3 Formal Statements of Our Results

It is instructive to first focus on conjunctive queries with one relation symbol E of arity two. An example of such a query is the following formula:

$$x_1 \dots x_k \exists y : Ex_1y \wedge \dots \wedge Ex_ky \,. \tag{6}$$

The relation E corresponds to a graph G and the free and quantified variables will be assigned vertices of G. In this example, an assignment $a_1, \dots, a_k \in V(G)$ to the free variables satisfies the formula if and only if the vertices $a_1, \dots, a_k$ have a common neighbor in G. It will be convenient for us to view the formula as a graph H as depicted in Figure 2. The vertices of H are partitioned into a set $X = \{x_1, \dots, x_k\}$ of free variables and a set $Y = \{y\}$ of quantified variables. An assignment to the free variables corresponds to a function $a : X \to V(G)$, and such an assignment satisfies the formula if it can be consistently extended to a homomorphism from H to G. This motivates the following definition, where we only consider simple graphs without loops, so we do not allow atomic subformulas of the form Ezz.

▶ **Definition 7.** *A* graphical conjunctive query (H, X) *consists of a graph* H *and a set* X *of vertices of* H*. We let* $\mathsf{Hom}(H, X \to G)$ *be the set of all mappings from* X *to* $V(G)$ *that can be extended to a homomorphism from* H *to* G*, and we call these mappings partial homomorphisms. Formally, the set of partial homomorphisms is defined via*

$$\mathsf{Hom}(H, X \to G) = \left\{ a : X \to V(G) \;\middle|\; \exists h \in \mathsf{Hom}(H \to G) : h|_X = a \right\}. \tag{7}$$

Given two different graphical conjunctive queries (H, X) and $(\hat{H}, \hat{X})$ it might be the case that $\#\mathsf{Hom}(H, X \to \star)$ and $\#\mathsf{Hom}(\hat{H}, \hat{X} \to \star)$ are the same functions. An example for this is given in Figure 2. In this case, we say that (H, X) and $(\hat{H}, \hat{X})$ are *equivalent*, denoted as $(H, X) \sim (\hat{H}, \hat{X})$, and the subgraph-minimal elements of the induced equivalence classes are called *minimal*. In what follows, we classify the complexity of counting homomorphisms for classes of graphical conjunctive queries. More precisely, we consider the parameterized counting problem $\#\mathsf{Hom}(\Delta)$ for each fixed class Δ of graphical conjunctive queries. This problem is given as input a query $(H, X) \in \Delta$ and a graph G and the task is to compute the number $\#\mathsf{Hom}(H, X \to G)$. The problem is parameterized by the size of H. We start with the formal definitions of the different structural parameters of graphical conjunctive queries and present the classification theorem thereafter. All parameters, along with five example classes, are depicted in Figure 3.

▶ **Definition 8** (Contract). *The* contract *of a graphical conjunctive query (H, X) is a graph on the vertex set X, obtained by adding an edge between two vertices u and v in X if uv is an edge of H or if there exists a connected component C in $H \setminus X$ that is adjacent to both u and v. Given a class Δ of conjunctive queries, we write* $\mathsf{contract}(\Delta)$ *for the set of all of its contracts.*

▶ **Definition 9** (Dominating star size). *Let (H, X) be graphical conjunctive query and let $Y_1, \dots, Y_\ell$ be the connected components of the subgraph $H[V(H) \setminus X]$ induced by the quantified variables. Further, let k_i be the number of vertices $x \in X$ for which there exists a vertex $y \in Y_i$ that is adjacent to x. The* dominating star size *of (H, X) is defined via*

$$\mathsf{dss}(H, X) = \max\{\, k_i \mid i \in \ell \,\}\,.$$

We are now in position to state our main result, the full classification for counting answers to conjunctive queries. Note that Theorem 2 is subsumed by the full classification in the case of graphs. The general version, that is, the case of arbitrary logical signatures with bounded arity, is stated and proved in the full version.

▶ **Theorem 10.** *Let Δ be a recursively enumerable class of minimal conjunctive queries.*

1. *If the treewidth of Δ and* $\mathsf{contract}(\Delta)$ *is bounded, then $\#\mathsf{Hom}(\Delta)$ can be computed in polynomial time.*
2. *If the treewidth of Δ is unbounded and the treewidth of* $\mathsf{contract}(\Delta)$ *is bounded, then $\#\mathsf{Hom}(\Delta)$ is* W[1]*-equivalent.*
3. *If the treewidth of* $\mathsf{contract}(\Delta)$ *is unbounded and the dominating star size of Δ is bounded, then $\#\mathsf{Hom}(\Delta)$ is* #W[1]*-equivalent.*
4. *If the dominating star size of Δ is unbounded, then $\#\mathsf{Hom}(\Delta)$ is* #W[2]*-hard. Moreover, for any fixed query δ with $\mathsf{dss}(\delta) \geq 3$, the problem $\#\mathsf{Hom}(\delta \to \star)$ cannot be computed in time $O(n^{\mathsf{dss}(\delta)-\varepsilon})$ for any $\varepsilon > 0$ unless SETH fails.*
5. *If the linked matching number of Δ is unbounded, then $\#\mathsf{Hom}(\Delta)$ is* #A[2]*-equivalent.*

In proving the last case of Theorem 10, we establish the following generalization of the Excluded-Grid-Theorem which applies for conjunctive queries with a large linked matching number and is essentially equivalent to Theorem 4; consult Figure 3 for the notion of a *grate*.

▶ **Theorem 11.** *Let Δ be a class of graphical conjunctive queries. If the linked matching number of Δ is unbounded, then Δ contains arbitrarily large grates as minors.*

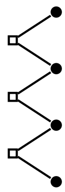
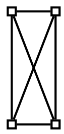
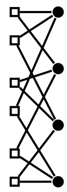
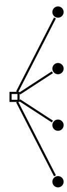
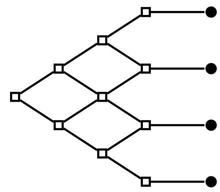
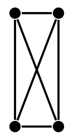
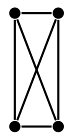
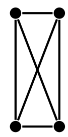
Query Classes	Δ_{poly}	$\Delta_{\text{W[1]}}$	$\Delta_{\#\text{W[1]}}$	$\Delta_{\#\text{W[2]}}$	$\Delta_{\#\text{A[2]}}$
Query for $k = 4$					
contract for $k = 4$		$\emptyset$			
tw	$O(1)$	∞	∞	$O(1)$	∞
tw(contract)	$O(1)$	$O(1)$	∞	∞	∞
dss	$O(1)$	$O(1)$	$O(1)$	∞	∞
lmn	$O(1)$	$O(1)$	$O(1)$	$O(1)$	∞
Complexity	P	W[1]-eq.	#W[1]-eq.	#W[2]-hard	#A[2]-eq.$^{(*)}$

Figure 3 Example problems for each case of the complexity classification (Theorem 10):

$$\Delta_{\text{poly}} = \{\, \varphi_k \mid k \in \mathbb{N} \,\}, \quad \text{where } \varphi_k := x_1 \dots x_k \; \exists y_1 \dots \exists y_{k-1} : \bigwedge_{1 \le i < k} E x_i y_i \wedge E y_i x_{i+1}$$

$$\Delta_{\text{W[1]}} = \{\, \psi_k \mid k \in \mathbb{N} \,\}, \quad \text{where } \psi_k := \exists y_1 \dots \exists y_k : \bigwedge_{1 \le i < j \le k} E y_i y_j$$

$$\Delta_{\#\text{W[1]}} = \{\, \upsilon_k \mid k \in \mathbb{N} \,\}, \quad \text{where } \upsilon_k := x_1 \dots x_k : \bigwedge_{1 \le i < j \le k} \exists y_{ij} : E x_i y_{ij} \wedge E y_{ij} x_j$$

$$\Delta_{\#\text{W[2]}} = \{\, \delta_k \mid k \in \mathbb{N} \,\}, \quad \text{where } \delta_k := x_1 \dots x_k \exists y : \bigwedge_{1 \le i \le k} E x_i y$$

Furthermore, $\Delta_{\#\text{A[2]}}$ is the set of all *grates*. Here, a k-grate is the conjunctive query whose quantified variables constitute half of a $k \times k$ grid whose diagonal is connected to k free variables by a matching of size k. The formal definition is given in the full version.
Depicted is the query (H, X) for $k = 4$, where free variables (i.e., vertices in X) are drawn as solid discs and quantified variables (i.e., vertices in $V(G) \setminus X$) are drawn as hollow squares. We also display the contract (see Definition 8) of each query. We write $O(1)$ whenever a parameter is bounded by a constant in the entire query class, and ∞ whenever it is unbounded. Finally, we show the complexity of counting answers to conjunctive queries in each of the classes in terms of polynomial-time tractability (P) and equivalence (-eq.) or hardness for one of the parameterized complexity classes.

$(*)$ The observant reader might have noticed that a k-grate is not a minimal conjunctive query. For this reason, #A[2]-equivalence in the last column refers to minimal conjunctive queries that contain arbitrary large grates as minors. Alternatively, #A[2]-equivalence is shown to hold for k-grates in the color-prescribed case. Details are given in the full version.

Building upon Theorem 10, we invoke the complexity monotonicity property of linear combinations of conjunctive queries to classify the complexity of counting answers to queries of the form (2), that is, existential or universal first-order queries for which we allow inequalities and non-monotone constraints, both over the free variables. Specifically, we prove the classification as given by Theorem 3. Again, we refer to the full version for the general case of arbitrary logical structures of bounded arity. We also discuss conjunctive queries that may contain inequalities over the free variables. Note that in that case, we are able to give explicit criteria for the five different cases in Theorem 3.

4 Conclusions

We established a comprehensive classification of the complexity of counting answers to conjunctive queries and linear combinations thereof. Depending on the structural parameters of the class of allowed queries, the problem is either polynomial-time solvable, W[1]-equivalent, #W[1]-equivalent, #W[2]-hard or #A[2]-equivalent. This classification, however, leaves out a gap between the latter two cases. More precisely, the following question remains open:

Does a class of conjunctive queries Δ exist for which $\#\mathsf{Hom}(\Delta)$ is #W[2]-hard but neither equivalent for #W[2] nor for #A[2]?

We conjecture a positive answer; the interested reader is encouraged to make themself familiar with the parameterized complexity class $\mathsf{W}_{\mathsf{func}}[2]$ (see e.g. [14, Chapter 8.8]). This class has a canonical counting version which we call $\#\mathsf{W}_{\mathsf{func}}[2]$ and which interpolates between #W[2] and #A[2]. In particular, we conjecture that there exists a class of conjunctive queries Δ for which $\#\mathsf{Hom}(\Delta)$ is $\#\mathsf{W}_{\mathsf{func}}[2]$-equivalent. Consequently, a negative answer to the previous question would imply that either $\#\mathsf{W}_{\mathsf{func}}[2] = \#\mathrm{W}[2]$ or $\#\mathsf{W}_{\mathsf{func}}[2] = \#\mathrm{A}[2]$, which seems to be very unlikely (see e.g. the discussion of $\mathsf{W}_{\mathsf{func}}[2]$ in [14, Chapter 8.8]).

A further question that remains open, and which should be considered a stronger version of the previous question, reads as follows:

Does a class of conjunctive queries Δ exist such that Δ has bounded linked matching number and the problem $\#\mathsf{Hom}(\Delta)$ is #A[2]-equivalent?

In other words, the above question asks whether the absence of a bound on the linked matching number is not only sufficient, but also necessary for #A[2]-equivalence. In contrast to the previous question, we conjecture a negative answer. Let us provide some intuition for the latter conjecture: It seems that a constant bound on the linked matching number of a class of conjunctive queries Δ yields a separator decomposition of the quantified variables of queries in Δ in components that have either small treewidth or a small matching number to the free variables. We conjecture that such a decomposition implies the existence of what is called a κ-restricted nondeterministic Turing machine $\mathbb{M}$ such that the number of accepting paths of $\mathbb{M}$ on input $(H, X) \in \Delta$ and a graph G is precisely $\#\mathsf{Hom}(H, X \to G)$ (see e.g. [14, Definition 14.15]). If additionally $\#\mathsf{Hom}(\Delta)$ is #A[2]-equivalent, this would imply that the set of #A[2]-equivalent problems is a subset of the set of #W[P]-equivalent problems; consult e.g. [14, Chapter 3 and 14.2] for a treatment of the class #W[P]. However, the latter inclusion seems to be unlikely and we refer the interested reader to [14, Chapter 8] for a detailed treatment of the corresponding question whether $\mathrm{A}[2] \subseteq \mathsf{W}[\mathsf{P}]$ in the decision world. We conclude with the remark that even a proof of $\mathrm{A}[2] \subseteq \#\mathsf{W}[\mathsf{P}]$ would be a major breakthrough as it constitutes the first step of a parameterized analogue of Toda's theorem [31], which is one of the fundamental open problems in (structural) parameterized counting complexity.

References

1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: http://webdam.inria.fr/Alice/.

2 Ashok K. Chandra and Philip M. Merlin. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90, 1977. doi:10.1145/800105.803397.

3 Chandra Chekuri and Anand Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000. doi:10.1016/S0304-3975(99)00220-0.

4 Hubie Chen and Stefan Mengel. A Trichotomy in the Complexity of Counting Answers to Conjunctive Queries. In *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*, pages 110–126, 2015. doi:10.4230/LIPIcs.ICDT.2015.110.

5 Hubie Chen and Stefan Mengel. Counting Answers to Existential Positive Queries: A Complexity Classification. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 315–326, 2016. doi:10.1145/2902251.2902279.

6 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th ACM Symposium on Theory of Computing, STOC*, pages 210–223, 2017. doi:10.1145/3055399.3055502.

7 Radu Curticapean and Dániel Marx. Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 130–139, 2014. doi:10.1109/FOCS.2014.22.

8 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoretical Computer Science*, 329(1):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

9 Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In Pascal Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002, 8th International Conference, CP 2002, Ithaca, NY, USA, September 9-13, 2002, Proceedings*, volume 2470 of *Lecture Notes in Computer Science*, pages 310–326. Springer, 2002. doi:10.1007/3-540-46135-3_21.

10 Reinhard Diestel, Tommy R. Jensen, Konstantin Yu. Gorbunov, and Carsten Thomassen. Highly Connected Sets and the Excluded Grid Theorem. *Journal of Combinatorial Theory, Series B*, 75(1):61–73, 1999. doi:10.1006/jctb.1998.1862.

11 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

12 Arnaud Durand and Stefan Mengel. Structural Tractability of Counting of Solutions to Conjunctive Queries. *Theory of Computing Systems*, 57(4):1202–1249, 2015. doi:10.1007/s00224-014-9543-y.

13 Jörg Flum and Martin Grohe. The Parameterized Complexity of Counting Problems. *SIAM Journal of Computing*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

14 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. doi:10.1007/3-540-29953-X.

15 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, Saket Saurabh, and B. V. Raghavendra Rao. Faster algorithms for finding and counting subgraphs. *jcss*, 78(3):698–706, 2012. doi:10.1016/j.jcss.2011.10.001.

16 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for First-Order Properties on Sparse Structures with Algorithmic Applications. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2162–2181, 2017. doi:10.1137/1.9781611974782.141.

17 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1, 2007. doi:10.1145/1206035.1206036.

18 Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 657–666, 2001. doi:10.1145/380752.380867.

19 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

20 Anthony C. Klug. On conjunctive queries containing inequalities. *Journal of the ACM*, 35(1):146–160, 1988. doi:10.1145/42267.42273.

21 Miroslaw Kowaluk, Andrzej Lingas, and Eva-Marta Lundell. Counting and Detecting Small Subgraphs via Equations. *SIAM Journal on Discrete Mathematics*, 27(2):892–909, 2013. doi:10.1137/110859798.

22 László Lovász. Operations with structures. *Acta Mathematica Hungarica*, 18(3-4):321–328, 1967.

23 László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Society Providence, 2012.

24 Dániel Marx. Can You Beat Treewidth? *Theory of Computing*, 6(1):85–112, 2010. doi:10.4086/toc.2010.v006a005.

25 Stefan Mengel. *Conjunctive queries, arithmetic circuits and counting complexity.* PhD thesis, University of Paderborn, 2013. URL: http://nbn-resolving.de/urn:nbn:de:hbz:466:2-11944.

26 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.

27 Reinhard Pichler and Sebastian Skritek. Tractable counting of the answers to conjunctive queries. *Journal of Computer and System Sciences*, 79(6):984–1001, 2013. doi:10.1016/j.jcss.2013.01.012.

28 Mihai Pătraşcu and Ryan Williams. On the Possibility of Faster SAT Algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075, 2010. doi:10.1137/1.9781611973075.86.

29 Gian-Carlo Rota. On the foundations of combinatorial theory I. Theory of Möbius functions. *Probability theory and related fields*, 2(4):340–368, 1964.

30 Marc Roth. Counting Restricted Homomorphisms via Möbius Inversion over Matroid Lattices. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, pages 63:1–63:14, 2017. doi:10.4230/LIPIcs.ESA.2017.63.

31 Seinosuke Toda. PP is as Hard as the Polynomial-Time Hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. doi:10.1137/0220053.

32 Ryan Williams. Faster decision of first-order graph properties. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 80:1–80:6, 2014. doi:10.1145/2603088.2603121.

33 Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM Journal of Computing*, 42(3):831–854, 2013. doi:10.1137/09076619X.