Solving a Family Of Multivariate Optimization and Decision Problems on Classes of Bounded Expansion

Daniel Mock
RWTH Aachen University, Germany

Peter Rossmanith
RWTH Aachen University, Germany

Abstract

For some time, it has been known that the model checking problem for first-order formulas is fixed-parameter tractable on nowhere dense graph classes, so we shall ask in which direction there is space for improvements. One of the possible directions is to go beyond first-order formulas: Augmenting first-order logic with general counting quantifiers increases the expressiveness by far, but makes the model checking problem hard even on graphs of bounded tree-depth. The picture is different if we allow only "simple" – but arbitrarily nested – counting terms of the form $\#y\,\varphi(\bar{x}, y) > N$. Even then, only approximate model checking is possible on graph classes of bounded expansion. Here, the largest known logic fragment, on which exact model checking is still fpt, consists of formulas of the form $\exists x_1 \dots \exists x_k \#y\,\varphi(\bar{x}, y) > N$, where $\varphi(\bar{x}, y)$ is a first-order formula without counting terms. An example of a problem that can be expressed in this way is partial dominating set: Are there k vertices that dominate at least a given number of vertices in the graph? The complexity of the same problem is open if you replace *at least* with *exactly*. Likewise, the complexity of "are there k vertices that dominate at least half of the blue and half of the red vertices?" is also open. We answer both questions by providing an fpt algorithm that solves the model checking problem for formulas of the more general form $\psi \equiv \exists x_1 \dots \exists x_k\, P(\#y\,\varphi_1(\bar{x}, y), \dots, \#y\,\varphi_\ell(\bar{x}, y))$, where P is an arbitrary polynomially computable predicate on numbers. The running time is $f(|\psi|)n^{\ell+1}\,\mathrm{polylog}(n)$ on graph classes of bounded expansion. Under SETH, this running time is tight up to almost linear factor.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Logic; Mathematics of computing → Graph algorithms

Keywords and phrases bounded expansion, parameterized algorithms, sparsity, counting logic, dominating set, model checking, multivariate optimization

Digital Object Identifier 10.4230/LIPIcs.SWAT.2024.35

Funding Funded by the Deutsche Forschungsgemeinschaft (DFG, German Science Foundation) – RO 927/15-2.

1 Introduction

Many problems on finite structures, in particular on graphs, can be expressed by logical formulas and then solved by meta-algorithms for the model checking problem. One of the best known meta-algorithms is due to Courcelle that solves all problems that can be expressed in monadic second order logic on graph classes with bounded tree-width in linear fpt when parameterized by length of the formula [3], that is, in time $f(|\varphi|)n$ where n is the size of the input graph for some function f. Taking a less expressive logic, for example MSO_1 instead of full MSO, you can solve the model checking problem on larger classes of graphs, i.e., on bounded clique width [4]. Unfortunately, there is some evidence that it is unlikely we can extend these results beyond bounded tree-width, resp. bounded clique-width [20, 14].

First-order logic (FO) is still less expressive, but powerful enough to model many decision problems. For example, the dominating set problem can be expressed by the FO-formula

$$\psi \equiv \exists x_1 \ldots \exists x_k \forall y \bigvee_{i=1}^{k} (y = x_i \vee y \sim x_i),$$

which says "are there vertices $x_1, \ldots, x_k$ such that every vertex y is one of them or adjacent to one of them?" Viewing the dominating set problem as parameterized problem, where the solution size k is the parameter, the parameter for for the model checking problem beocomes the length of the fomula $|\varphi|$. Many other problems like this, for example independent set or chordal deletion, can be modeled in FO-logic, making FO model checking a powerful tool to solve parameterized problems. As FO is less expressive than MSO, it is not surprising that efficient model checking algorithms exist beyond bounded tree-width. There is a series of results proving fpt running times on bounded degree graphs, planar graphs, minor-closed classes, topological minor-closed classes, locally bounded tree-width and many more [26, 13, 6, 18]. All these graph classes are generalized by *bounded expansion* or *nowhere dense* classes which are part of the beautiful sparsity theory introduced by Nešetřil and Ossona de Mendez [24]. Dvořák, Kráľ, and Thomas [11] showed that FO model checking is linear fpt on bounded expansion and Grohe, Kreutzer, and Siebertz [16] showed an almost linear fpt run time on nowhere dense graph classes (the run time is $f(|\varphi|)n^{1+o(1)}$ for some function f depending on the graph class). The latter result is in some sense final for sparse graph classes: If a graph class is monotone, that is, closed under taking subgraphs, then FO model checking is fpt on this class iff the class is nowhere dense. One recent development that reaches beyond sparsity is the result by Dreier, Mählmann, and Siebertz who look at *structurally* sparse graph classes [7].

If we turn from decision problems to optimization problems, the picture changes again. In some cases an optimization problem can be solved by using the underlying decision problem as a subroutine. For example, we can easily find the smallest dominating set for a graph by trying all k's. On the other hand if we fix k and ask what is the best partial dominating set with k vertices, i.e., how many vertices can be dominated by k vertices, a self-reduction seems to be impossible. For such questions, we need a more expressive logic. Kuske and Schweikardt introduced the logic FOC($\mathbf{P}$), which is FO with powerful counting extensions [21] and showed that its model checking problem is fpt on classes of bounded degree. However, FOC($\mathbf{P}$) is so expressive that its model checking problem becomes hard on trees of bounded depth. This stays true even for the fragment FO($\{>0\}$), which is still very expressive, but at least allows for an *approximative* model checking algorithm in fpt time [10] on classes of bounded expansion; the algorithms gives the correct answer or responds with "I do not know". The latter answer is allowed only if small pertubations in the number symbols of the formula make it both satisfied and unsatisfied.

The most general type of counting formulas known so far that can efficiently and exactly be evaluated on graph classes of bounded expansion are of the form

$$\exists x_1 \ldots \exists x_k \, \#y \, \varphi(y, x_1, \ldots, x_k) > N, \tag{1}$$

where N is a number that can depend on the order of the graph [8]. The semantics of the counting term $\#y \, \varphi(y, u_1, \ldots, u_k)$ for $u_i \in V(G)$ is the number of vertices $v \in V(G)$ with $G \models \varphi(v, u_1, \ldots, u_k)$. In this way, partial dominating set can be expressed as $\exists x_1 \ldots \exists x_k \, \#y \bigvee_{i=1,\ldots,k}(y = x_i \vee y \sim x_i) > N$, where $u \sim v$ denotes adjacency in the graph. Instead of graphs we consider colored graphs, where nodes can have colors. In a formula we talk about colors by using a unary predicate for each color. In that way we can express questions like "are there k nodes dominating at least N of the blue nodes?" A corresponding formula might look like

$$\exists x_1 \ldots \exists x_k \, \#y \bigvee_{i=1}^{k} \big(blue(y) \wedge (y = x_i \vee y \sim x_i)\big) > N. \tag{2}$$

While we know that formulas like that can be evaluated efficiently on classes of bounded expansion, as it is of the form of (1), not much is known how much further we can extend the fragment of counting formulas beyond (1) without losing fixed parameter tractability. One might expect that changing "$> N$" into "$= N$" in (2) should not make the problem W-hard, yet neither is such a result known, nor does it follow from a simple self-reducibility argument such as using (2) and binary search. Similarly, asking for k vertices that simultaneously dominate at least N blue and N red vertices seems not fundamentally harder than the same problem with only one color class, but cannot be expressed in the form of (1) either.

For the sake of completeness, we mention two other fragments, which are orthogonal to the ones discussed before. The fragment $\mathrm{FOC}_1(\mathbf{P})$ of $\mathrm{FOC}(\mathbf{P})$ introduced by Grohe and Schweikardt extends FO by allowing to formulate cardinality constraints on counting terms that have at most one free variable. The model checking problem for $\mathrm{FOC}_1(\mathbf{P})$ is fpt on nowhere dense classes [17]. For bounded expansion classes, Toruńczyk introduced a stronger query language (also orthogonal to ours) extending FO logic by aggregation over semirings [27].

Algorithmic Results and Techniques. The main result (Theorem 6) of this paper is to answer these questions affirmatively and to go significantly further. We develop an algorithm that evaluates formulas of the form

$$\psi \equiv \exists x_1 \ldots \exists x_k \, P\big(\#y \, \varphi_1(x_1, \ldots, x_k, y), \ldots, \#y \, \varphi_\ell(x_1, \ldots, x_k, y)\big), \tag{3}$$

where P is an arbitrary polynomially computable predicate on numbers and the φ_i's are FO formulas. If P can be evaluated in constant or at least in $g(\ell)$ time[1] (in the uniform cost model for some function g) then evaluating ψ can be done in time $f(|\psi|)n^{\ell+1} \operatorname{polylog}(n)$ for some function f on graph classes of bounded expansion. If P is a boolean combination of $\geq$- or $\leq$-comparisons to constants, then the running time of the model checking problem improves to $f(|\psi|)n^{\ell} \operatorname{polylog}(n)$ (see Lemma 11). This subsumes the result of [10, Theorem 1.2] up to polylogarithmic factors. Later, we will extend the result to counting tuples with the counting quantifier, which means that $\#y$ can be replaced by $\#(y_1, \ldots, y_p)$ in formula 3.

Note that formulas of the form (3) are quite expressive. They capture many multivariate optimization problems where you look for k vertices that optimize a possibly quite complicated target function. For example, you might want to find a set of k vertices U such that every other vertex is reachable from U by a path of length at most ten. You pay a penalty for each vertex v proportional to the distance of v to U. Finding the k vertices with the lowest penalty is fpt on every graph class of bounded expansion because the problem can be expressed by a formula of the form (3). Moreover, the algorithm will not depend on the graph class (but its running time will). In a multivariate optimization problem we can also find witnesses for each Pareto-optimal solution size. For example, if we look for k vertices that dominate as many red and as many blue vertices as possible, there might exist $\Omega(n)$ many different Pareto-optimal ways.

[1] Technically it is sufficient that there is a pseudo-polynomial fpt algorithm that evaluates $P(n_1, \ldots, n_\ell)$ in time $f(\ell)(n_1 + \cdots + n_\ell)^{O(1)}$ in order to achieve an fpt running time for evaluating ψ.

In a nutshell, the results about formulas of the form (1) were proved as follows (see [10] for more details): First, the innermost existential and universal quantifiers of the input formula are treated with quantifier-elimination, resulting in a simpler formula. In this process, the graph is augmented with additional colored edges while still being part of (another) class of bounded expansion. Then, a counting term $\#y\,\varphi(y\bar{x})$, where φ is now quantifier-free, is replaced in a sequence of transformations by a sum of simpler counting terms. These are simple enough to be directly evaluated, and the evaluation is represented as a family of vertex weight functions. Then, the original task is reduced to finding a certain kind of induced subgraph of maximal weight (which is described by one of the quantifier-free formulas). Then one can use low treedepth colorings to reduce the problem to the case of bounded treedepth and use LinEMSOL optimization [4] as a black box to find the result.

This approach breaks down at several places if we want to evaluate a formula of the more complicated form (3). For example, LinEMSOL optimization always finds the biggest weighted solution and cannot be used to establish the existence of a given weight. Moreover, it can handle only univariate weights. In order to prove our main result, we use dynamic programming on the treedepth decomposition using vectors of weights. This is achieved by formulating the problem of finding a subgraph described by a quantifier-free formula with constraints on the multivariate weights as an *MSO-evaluation problem*. This means that the problem can be expressed as a homomorphism that maps the satisfying assignments of the MSO-formula into a semiring. After applying treedepth colorings as in [10] those problems can be evaluated quickly on graphs of bounded treedepth by the result of Courcelle and Mosbah [2].

Lower Bounds. The second result is a conditional lower bound for formulas of the form (3) (Theorem 18). Under SETH we cannot evaluate formulas of the restricted form

$$\exists x_1 \dots \exists x_k \bigwedge_{i=1}^{\ell} \#y\,\varphi_i(x_1,\dots,x_k,y) = N_i,$$

on star forests[2] where the φ_i's are quantifier-free formulas and the N_i's are numbers, in time $f(k+\ell)n^{\ell-\varepsilon}$ for any $\varepsilon > 0$ and function f. We reduce from the k-Sum problem, using a tight conditional lower bound on dense k-Sum instances under SETH [1]. The upper and lower bounds given by Theorems 6 and 18 are tight up to an almost linear factor in n.

In Theorem 20 we show that introducing an additional universal quantifier between the existential quantifiers and the counting quantifier makes evaluating those formulas on forests of depth 2 W[1]-hard. The same holds if one has two nested counting quantifiers.

2 Preliminaries

We write $\mathbf{Z}$ and $\mathbf{N}$ for the integers and natural numbers (including 0), $[k] = \{1,\dots,k\}$.

Graphs. We consider labeled graphs $G = (V, E, P_1, \dots, P_m)$ where V is the vertex set, E the edge set and $P_1,\dots,P_m \subseteq V$ the labels of G. The order $|G|$ of G equals $|V|$. The size $\|G\|$ of G is $|V| + |E| + |P_1| + \dots + |P_m|$. We often write $V(G)$ and $E(G)$ for the vertex and edge sets of G. Unless stated otherwise, graphs are undirected. An *expansion* G' of a graph G is a graph on the same vertex set as G and its edges form a superset of $E(G)$.

[2] A star forest is a disjoint union of stars.

Sparse graph classes. A *treedepth decomposition* of a graph G is a rooted forest F on the same vertices as G, such that for every edge $uv \in E(G)$ either u is an ancestor of v in F or vice versa. The *treedepth* of a graph G is the minimum depth any treedepth decomposition of G.

A graph G is a *topological depth-r minor* of another graph H if an $\leq r$-subdivision of G is isomorphic to a subgraph of H [24, 5].

▶ **Definition 1** ([24]). A graph class $\mathcal{C}$ has *bounded expansion* if for all $r \in \mathbf{N}$ there exists $t = t(r) \in \mathbf{N}$ such that for all $G \in \mathcal{C}$, and all topological depth-r minors H of G, $\|H\|/|H| \leq t$.

It is *nowhere dense* if for all $r \in \mathbf{N}$ there exists a $t \in \mathbf{N}$ such that no $G \in \mathcal{C}$ contains K_t as a topological depth-r minor. If a graph class is not nowhere dense it is *somewhere dense.*

▶ Remark 2. Following [16], a class has *effectively bounded expansion* if $t(r)$ is computable. If the class $\mathcal{C}$ in Lemmas 3 and 5 and Theorem 18 has effectively bounded expansion, then the algorithms in those statements are uniform and their running times are computable.

Logic. The notation $\bar{x}$ stands for $x_1 \ldots x_{|\bar{x}|}$. Usually $|\bar{x}|$ is denoted by k. We write $\varphi(\bar{x})$ if φ has $\bar{x}$ as free variables. Let G be a structure and β be the assignment with $\beta(x_i) = u_i$ for $i \in [k]$. For simplicity, we write $G \models \varphi(\bar{u})$ instead of the satisfaction relation $(G, \beta) \models \varphi(\bar{x})$ and $[\![\varphi(\bar{u})]\!]^G$ instead of the interpretation $[\![\varphi(\bar{x})]\!]^{(G,\beta)}$ which can be a number if φ is a counting term from FOC($\mathbf{P}$). A formula is *quantifier-free* if it contains no $\exists, \forall$ or $\#$ quantifiers. A *conjunctive clause* is a conjunction over (possibly negated) predicates. The *length* of a formula φ is denoted by $|\varphi|$ and is the number of symbols in φ. In particular, the length of any number-symbol $N \in \mathbf{Z}$ is one (and should not be confused with the length of the binary encoding of N). Adjacency between two variables x, y is denoted by $x \sim y$. For two signatures we write $\sigma \subseteq \rho$ to indicate that ρ extends σ. All signatures are finite and the cardinality $|\sigma|$ of a signature equals its number of symbols.

We interpret a labeled graph $G = (V, E, P_1, \ldots, P_m)$ as a logical structure with universe $V(G) \coloneqq V$, binary relation $E(G) \coloneqq E$ and unary relations $P_1, \ldots, P_m$.

Counting Logic. We are interested in fragments of the counting logic FO($\mathbf{P}$), introduced by Kuske and Schweikardt [21], which is a fragment of FOC($\mathbf{P}$). Compared to traditional first-order logic, it introduces counting over one variable and comparing the number of witnesses using numerical predicates from a set $\mathbf{P}$. We give an informal definition. Given a FO($\mathbf{P}$) formula φ and a variable y, $\#y\, \varphi(y\bar{x})$ is a *counting term.* The semantics of $[\![\#y\, \varphi(y\bar{u})]\!]^G$ are the number of vertices v satisfying $[\![\varphi(v\bar{u})]\!]^G$. Multiple counting terms $t_1, \ldots, t_m$ with a predicate $P \in \mathbf{P}$ yield an FO($\mathbf{P}$) formula $P(t_1, \ldots, t_m)$. The semantics of this formula is true iff the evaluation of the counting terms is in $[\![P]\!]$. This is generalized by FOC($\mathbf{P}$), which allows counting tuples $\#\bar{y}\, \varphi(\bar{y}\bar{x})$ and allows addition and multiplication of counting terms. For more details, we refer the reader to [21]. The logic FO($\{>0\}$) [10] is a special case of FO($\mathbf{P}$) where the unary predicates are of the form $\geq t$ for $t \in \mathbf{N}$. That is, counting terms can be used only in the form $\#y\, \varphi(y\bar{x}) \geq t$. This logic is also known as FO(C) [12].

Computational model. We use the RAM model with a uniform cost measure.

If a function $f\colon Z^\ell \to \mathbf{Z}$ is part of the output of an algorithm, it is represented by a list of its non-zero entries. Thus, even if the domain has infinite size, the representation is finite if f's support is. In most of our cases, the support of f will be $\{-N, \ldots, N\}^\ell$ for some N, which usually is linear in the input.

3 Applications

Before we get into the main part of this work, we want to illustrate in detail its possible applications with a few examples. Let us consider a variation of the partial dominating set problem, the EXACTLY t-DOMINATING SET PROBLEM [19]. We are interested in set D of size k that dominates *exactly* t vertices in a graph G. By deciding $G \models \psi$ using Theorem 6 for the formula

$$\psi \equiv \exists x_1 \ldots \exists x_k \# y \, \Big(\bigvee_{i=1}^{k} y = x_i \vee y \sim x_i \Big) = t$$

we can solve the EXACTLY t-DOMINATING SET PROBLEM on classes of bounded expansion in time $f(k)n^2 \operatorname{polylog} n$. As a byproduct we get a list of all t such that there exists such a set of size k. Hence, one could also solve the problem of finding a partial dominating set of size k where the number of dominated vertices t has to satisfy some (easily computable) numerical predicate P, e.g., that t has to be prime.

Another variation: Given a graph where the vertices are colored red or blue, we want to find a set D of k vertices that dominates at least t_1 many red and at least t_2 many blue vertices. This can be expressed by the following formula:

$$\exists x_1 \ldots \exists x_k (\# y \, \big(\mathit{blue}(y) \wedge \bigvee_{i=1}^{k} (y = x_i \vee y \sim x_i) \big) \geq t_1 \wedge \# y \, \big(\mathit{red}(y) \wedge \bigvee_{i=1}^{k} (y = x_i \vee y \sim x_i) \big) \geq t_2)$$

Thus, we can evaluate this problem in $f(k)n^2 \operatorname{polylog} n$ time on classes of bounded expansion by Lemma 11. If "$\geq$" is replaced by "$=$" in the formula above, by Theorem 6 we can even compute all pairs (t_1, t_2) where the adjusted formula holds, but in time $f(k)n^3 \operatorname{polylog} n$. Additionally, it is also possible to express that the vertices $x_1, \ldots, x_k$ are, e.g., connected or independent. This increases the run time only in the function $f(k)$.

4 Algorithms

In this section we show the algorithmic results, most importantly Theorem 6 in Sections 4.1 and 4.2. Namely, that both the query evaluation and query counting problem for formulas of the form $\psi \equiv \exists x_1 \ldots \exists x_k \, P(\# y \, \varphi_1(\bar{x}, y), \ldots, \# y \, \varphi_\ell(\bar{x}, y))$ can be solved in $f(|\psi|)n^{\ell+1} \operatorname{polylog} n$ time on graph classes of bounded expansion. This is followed by an improvement in the running time by a factor n for the special case of Lemma 11 that deals with $\leq$-constraints in Section 4.3. In a nutshell, we put the information needed to evaluate general counting terms depending on multiple variables into vertex weights that depend on only one variable. The resulting combinatorial problem is then reduced to the case of bounded treedepth. Solving this problem on bounded treedepth is then deferred to its own Section 4.2 as it is the main technical contribution of this paper. There, we introduce Mosbah's and Courcelle's concept of MSO-evaluation problems. Then we show that our problem can indeed be modeled as an MSO-evaluation problem and that we get the desired running time. In Section 4.3 we consider the special case of $\leq$-constraints, which can be handled more efficiently. At last, in Section 4.4 we show that the previous results also hold if the counting quantifier can count tuples instead of single variables.

4.1 Reduction to Weighted Sets

We build upon the tools used in [10][3] as there a similar but weaker fragment of $\mathrm{FO}(\{>0\})$ was considered, namely formulas of the form $\exists x_1 \dots \exists x_k \ \#y\, \varphi(y, x_1, \dots, x_k) > N$, (see (1)). However, we need to adapt them to the multivariate case, i.e., where we consider multiple counting terms $\#y\, \varphi(y\bar{x})$ at once.

The result of the following lemma is similar to [9, Theorem 4]. The only difference is that equality between $[\![\#y\, \varphi(y\bar{u})]\!]^{\vec{G}}$ and $\sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i)$ is generalized to the ℓ formulas $\varphi_1, \dots, \varphi_\ell$ and ℓ sums over "families" of weight functions $c_{\omega,i}^{(j)}$. Note that the set Ω is the same for every φ_j and that the running time does not change compared to [9, Theorem 4].

The following lemma allows us to represent counting terms that depend on $k+1$ vertices as a sum of vertex weights which depend only on one variable. This procedure works even for multiple counting terms at once.

▶ **Lemma 3.** *Let $\mathcal{C}$ be a class of bounded expansion with signature σ. One can compute for first-order formulas $\varphi_1(y\bar{x}), \dots, \varphi_\ell(y\bar{x})$ with signature σ a set of conjunctive clauses Ω with free variables $\bar{x}$, and signature $\rho \supseteq \sigma$ that satisfies the following property:*

There exists a class $\mathcal{C}'$ of bounded expansion with signature ρ. such that for every $\vec{G} \in \mathcal{C}$ one can compute in time $O(\|\vec{G}\|)$ an expansion $\vec{G}' \in \mathcal{C}'$ of $\vec{G}$ and functions $c_{\omega,i}^{(j)}(v) \colon V(\vec{G}) \to \mathbf{Z}$ for $\omega \in \Omega$, $i \in [|\bar{x}|]$, and $j \in [\ell]$ with $c_{\omega,i}^{(j)}(v) = O(|\vec{G}|)$ such that for every $\bar{u} \in V(\vec{G})^{|\bar{x}|}$ there exists exactly one formula $\omega \in \Omega$ with $\vec{G}' \models \omega(\bar{u})$. For this formula

$$\Big([\![\#y\, \varphi_1(y\bar{u})]\!]^{\vec{G}}, \dots\dots, [\![\#y\, \varphi_\ell(y\bar{u})]\!]^{\vec{G}}\Big) = \Big(\sum_{i=1}^{|\bar{x}|} c_{\omega,i}^{(1)}(u_i), \dots\dots, \sum_{i=1}^{|\bar{x}|} c_{\omega,i}^{(\ell)}(u_i)\Big).$$

Proof. When we look into the proof of [9, Theorem 4], we see that the expansion $\vec{G}'$ of $\vec{G}$ does not depend on any formula, but only on G. Moreover, the set of conjunctive clauses Ω depends only on the signature ρ. Thus, when applying [9, Theorem 4] to the formulas $\varphi_1, \dots, \varphi_\ell$ sequentially both the graph extension $\vec{G}'$ and the set of clauses Ω are identical for each application. Only the weight functions depend on $\varphi_1, \dots, \varphi_\ell$. Thus, we get our result. ◀

For simplicity, let us define $C_\omega^{(j)}(\bar{u}) \coloneqq \sum_{i=1}^{|\bar{x}|} c_{\omega,i}^{(j)}(u_i)$ and abbreviate it as $C^{(j)}(\bar{u})$ if ω is clear from context.

With Lemma 3, the original task boils down to finding a vertex tuple $\bar{u}$ that satisfies a simple quantifier-free formula ω and has "correct" vertex weights, i.e., $C^{(j)}(\bar{u}) = w_j$ for all $j \in \ell$. Let us give a formal definition.

▶ **Definition 4.**

Counting ℓ-Weighted k-Set Problem
Input: $k, \ell \in \mathbf{N}$, a graph G, a quantifier-free FO-formula ω with variables $x_1 \dots x_k$ and weight functions $c_i^{(j)} \colon V(G) \to \mathbf{Z}$ for $i \in [k], j \in [\ell]$
Output: a partial function $a \colon \mathbf{Z}^\ell \to \mathbf{N}$ where $a(w)$ is the number of $\bar{u} \in V(G)^k$ satisfying
1. $G \models \omega(\bar{u})$
2. $C^{(j)}(\bar{u}) = \sum_{i \in [k]} c_i^{(j)}(u_i) = w_j$ for all $j \in [\ell]$

[3] The relevant results for our approach can be found in more detail in the full version on arXiv [9].

To solve the Counting ℓ-Weighted k-Set Problem, we follow the approach in [9, Theorem 2] and reduce the case for graphs of bounded expansion to graphs of bounded treedepth. These graphs are structurally much simpler and have a rich landscape of meta-theorems. The reduction is achieved by a so-called low treedepth coloring [24].

For now, let us assume we can solve the problem above in time $f(|\omega|, d)n^{\ell+1} \operatorname{polylog} n$ on graphs of tree-depth d, as stated in the following lemma. We give the proof in Section 4.2.

▶ **Lemma 5.** *Assume we are given $k, \ell \in \mathbf{N}$, a quantifier-free first-order formula $\omega(x_1 \dots x_k)$, a treedepth-decomposition of a graph G of depth d and weight functions $c_i^{(j)} : V(G) \to \mathbf{Z}$ with $|C^{(j)}(v_1, \dots, v_k)| \leq N$ for $j \in [\ell]$ and $i \in [k]$. Then, we can solve the Counting ℓ-Weighted k-Set Problem with G, ω and $c_i^{(j)}$ as input in time $f(|\omega|, d)nN^\ell \log N$ for some function f.*

With this result at hand, we use a similar technique as in the proof of [9, Theorem 2].

▶ **Theorem 6.** *Let $\mathcal{C}$ be a class of bounded expansion. There exists a function f such that for all graphs $G \in \mathcal{C}$ and first-order formulas $\varphi_1(y\bar{x}), \dots, \varphi_\ell(y\bar{x})$, one can compute a function $a\colon \mathbf{Z}^\ell \to \mathbf{N}$ where $a(w_1, \dots, w_\ell)$ is the number of solutions $\bar{u} \in V(G)^k$ with*

$$G \models \psi(\bar{u}) \qquad \text{where } \psi(\bar{x}) \equiv \bigwedge_{j \in [\ell]} \#y\, \varphi_j(y\bar{x}) = w_j$$

in time $f(|\psi|)n^{\ell+1} \operatorname{polylog}(n)$.

Proof. We apply Lemma 3 to G and $\varphi_1, \dots, \varphi_\ell$ yielding a functional graph $\vec{G}'$ from a class of bounded expansion, a set of conjunctive clauses Ω over an extended signature, and weight functions $c_{\omega,i}^{(j)}\colon V(G) \to \mathbf{Z}$ for $\omega \in \Omega, i \in [k], j \in [\ell]$. Then

$$G \models \bigwedge_{j \in [l]} \#y\, \varphi_j(y\bar{u}) = w_j$$

iff there exists an $\omega \in \Omega$ such that both

$$\vec{G}' \models \omega(\bar{u}) \text{ and } C_\omega^{(j)}(\bar{u}) = w_j \text{ for all } j \in [\ell].$$

The latter is an instance of the Counting ℓ-Weighted k-Set Problem. Using the techniques from the proof of [9, Theorem 2] we reduce this problem from graphs of bounded expansion to graphs of bounded treedepth with the help of low treedepth colorings.

Before we continue, it will be easier to transform $\vec{G}'$ into a relational structure G': Every function f is replaced by a binary relation E_f with $E_f(G') = \{(v, f^{\vec{G}'}(v)) \mid v \in V(\vec{G}')\}$, and we keep unary predicates. For every (functional) conjunctive clause $\omega(\bar{x})$ we construct a relational conjunctive clause $\omega'(\bar{x}\bar{z})$ with $\vec{G}' \models \omega(\bar{u})$ iff $G' \models \exists \bar{z} \omega'(\bar{x}\bar{z})$ for every $\bar{u} \in V(\vec{G}')^{|\bar{x}|}$.

There exists a vertex coloring χ of G, so called *k-treedepth colorings*, such that each subgraph of G induced by at most k colors has at most treedepth k. Denote the set of all such subgraphs with $\mathcal{H}$. As G is from a class of bounded expansion $\mathcal{C}$, one can compute in $f_\mathcal{C}(k)n$ time a k-treedepth coloring that uses at most $f_\mathcal{C}(k)$ colors [24]. Hence, if we want to "find" a fixed k-vertex tuple $\bar{u} \in V(G)^k$, there must exist k colors such that $\bar{u}$ is contained in the subgraph induced by those k colors. So, we need to consider only $\binom{f(k)}{k}$ such subgraphs when looking for a solution to the ℓ-Weighted k-Set Problem and solve the problem on such a subgraph which has treedepth k. For more details, see the proof of [9, Theorem 2].

Thus, for every $\omega \in \Omega$, possible colorings of $\bar{u}$ with colors from χ and $H \in \mathcal{H}$ we apply Lemma 5 to solve the counting problem. Then, for every weight tuple $(w_1, \dots, w_\ell) \in \{-N, \dots, +N\}^\ell$ we add up the counts over all subgraphs $H \in \mathcal{H}$, possible colorings of $\bar{u}$, and $\omega \in \Omega$, and output this sum for $(w_1, \dots, w_\ell)$. In total, the computation time is $O\big(|\Omega|\binom{f(k)}{k}knN^\ell \operatorname{poly}(\ell) \operatorname{polylog}(N) + f'(|\psi|)n\big) = O\big(g(|\psi|)n^{\ell+1} \operatorname{polylog}(n)\big)$. ◀

By using Theorem 6 and looking for a tuple $(w_1, \dots, w_\ell)$ which is contained in P and has strictly positive output we get the following corollary.

▶ **Corollary 7.** *Let $\mathcal{C}$ be a class of bounded expansion. There exists a function f such that for all graphs $G \in \mathcal{C}$, first-order formulas $\varphi_1(y\bar{x}), \dots, \varphi_\ell(y\bar{x})$, and all computable relations $P \subseteq \mathbf{N}^\ell$ one can decide*

$$G \models \exists x_1 \dots \exists x_k P(\#y\, \varphi_1(y\bar{x}), \dots, \#y\, \varphi_\ell(y\bar{x}))$$

in time $O\big(f(|\varphi_1| + \cdots + |\varphi_\ell|)n^{\ell+1} \operatorname{polylog}(n) + rn^\ell\big)$ where r is the time needed to decide membership in P.

4.2 Solving the ℓ-Weighted k-Set Problem on bounded treedepth

In order to construct an algorithm satisfying Lemma 5 (that is, solving the ℓ-Weighted k-Set Problem on bounded treedepth graphs), we will use the machinery of *monadic second-order evaluations* on graphs of bounded treewidth (on bounded treedepth even), introduced by Courcelle and Mosbah [2]. (This should not be confused with Courcelle's theorem for MSO model checking on graphs of bounded treewidth [3]). Monadic second order evaluations can be used to compute a function of an optimal solution if the function can be computed iteratively on a tree decomposition. In the case of a join node the values of the children have to be combined in some way and in the case of introduce and forget nodes the values have to be updated in the right way. When you use this machinery, and typically in other DP algorithms as well, the values you use carry more information than is needed in the end. Therefore, the needed information has to be extracted by an *homomorphism*. Updating the values is done by operations on a *semiring*. For example, if you want to find a vertex cover of minimal size in a node-weighted graph, your semiring would be $(\mathbf{N} \cup \{\infty\}, \min, +, \infty, 0)$ and the homomorphism would map a (partial) solution to its weight, which is the sum of the weights of each contained vertex.

We have to show that Counting ℓ-Weighted k-Set Problem is an MSO-evaluation problem. We refer the reader to [22, Section 3.3.3] for definitions of semirings, weak homomorphism, MSO logic, assignments and the semiring of assignments. We also follow their notation.

Let us very briefly mention the most relevant definitions. Let G be a graph and φ a first-order formula.[4] Then, $\operatorname{sat}(\varphi, G)$ denotes the set of assignments to variables of φ over G, and $\operatorname{assignring}(\varphi, G) = (2^{\operatorname{assignments}(\varphi,G)}, \cup, \hat{\cup}, \emptyset, \hat{\emptyset})$ denotes the *semiring over the power set of assignments* where $\hat{\cup}$ is the Cartesian product of assignments and $\hat{\emptyset}$ the set containing only the empty assignment. The operation $\hat{\cup}$ combines assignments from G_1 and G_2 when a graph G can be "decomposed" into two graphs G_1 and G_2, e.g., as in tree-decompositions.

▶ **Definition 8.** A graph problem P is an *MSO-evaluation problem* if there is an MSO-formula φ and a semiring $R = (U_R, \oplus, \otimes, \hat{0}, \hat{1})$ such that P can be stated as computing $h(sat(\varphi, G))$, where $G = (V, E)$ and the weak semiring homomorphism h between $\operatorname{assignring}(\varphi, G)$ and R are part of the input.

The following fact follows from Proposition 3.1 and Theorem 2.10 from [2].

▶ **Proposition 9.** *Let P be an MSO-evaluation problem, expressed by a weak homomorphism h into a semiring R. Then, P can be computed on a graph G given a tree decomposition $\mathcal{T}$ of width w in time $O(f_P(w)|\mathcal{T}|\eta)$ where η is the time complexity of evaluating the homomorphism and performing the semiring operations.*

---

[4] The formula can also be from MSO in general. For our needs, FO suffices.

To express the Counting ℓ-Weighted k-Set Problem as an MSO-evaluation problem, we define the semiring ℓ-WeightedSolCount $:= (U, \oplus, \otimes, \hat{0}, \hat{1})$ where elements of $U = \mathbf{N}^W$ are infinite sequence of natural numbers indexed by ℓ-tuples in $W = \mathbf{Z}^\ell$. An entry $a \in U$ means that, for a tuple $w \in W$, there are a_w many assignments $\bar{u} \in V(G)^k$ with multivariate weight w.

More formally, given a graph G with weight functions $c_i^{(j)}$ and a first-order formula φ, we define a weak homomorphism $h\colon \text{assignring}(\varphi, G) \to \ell\text{-WeightedSolCount}$ by

$$h(A) := a \text{ and for } w \in W \text{ with } a_w := \left|\{\, \bar{u} \in A \mid C^{(j)}(\bar{u}) = w_j \text{ for all } j \in [\ell] \,\}\right|$$

where A is some set of assignments[5] $\bar{u} \in V(G)^k$ to φ.

Hence, this problem is an MSO-evaluation problem by Definition 8. Equivalently, instead of a being a vector indexed by ℓ-tuples, we can imagine a being a function mapping an ℓ-tuple w to the number $a(w)$ of solutions with multivariate weight w.

But first, we need to complete the definition of ℓ-WeightedSolCount by defining the operations and constants. The addition $\oplus$ is the element-wise addition defined as

$$(a \oplus b)_w := a_w + b_w \text{ for } w \in W$$

with the neutral element $\hat{0} := (0)^W$, the all-zero vector indexed by W. The multiplication $\otimes$ is defined by the convolution

$$(a \otimes b)_w := \sum_{r+s=w} a_r \cdot b_s \text{ for } w \in W$$

with neutral element $\hat{1}$, where $\hat{1}_{(0,\ldots,0)} = 1$ and $\hat{1}_w = 0$ for $w \neq (0, \ldots, 0)$.

The weak homomorphism h is then defined by

$$h(\{\bar{u}\}) = (a_w)_{w \in W} \text{ with } a_w = \begin{cases} 1 & \text{if } C^{(j)}(\bar{u}) = w_j \text{ for all } j \in [\ell] \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{u}$ is an assignment from $V(G)^k$. The image of h for other sets of assignments is derived from the singleton sets and semiring properties.

One can easily verify that ℓ-WeightedSolCount is a semiring. Indeed, ℓ-WeightedSolCount is similar to the CardCounting semiring in [22, Example 28.4]. However, we are not interested in the number of solutions of some cardinality but of some certain weight. Moreover, the weight is multivariate and not univariate. Also, it should not be confused with the evaluation problem described in [2, Section 4.8]. There, a linear combination of multiple weight functions is considered, which is fundamentally different from our approach.

▶ **Example 10.** To show the Counting ℓ-Weighted k-Set Problem and ℓ-WeightedSolCount in action, let us consider the problem of finding an independent set of certain weight on a graph with two vertex weight functions, $c(v)$ and $c'(v)$. Let us say we are interested in an independent set with weights 24 and 96 w.r.t to c and c' respectively,

This problem can be easily modeled as Counting ℓ-Weighted k-Set Problem with $\ell = 2$ and $\omega(\bar{x}) = \bigwedge_{i<j} x_i \nsim x_j$. The output function $a(24, 96)$ tells us the number of (ordered) independent sets $\bar{u}$ with weight $\sum_i c(u_i) = 24$ and $\sum_i c'(u_i) = 96$.

---

[5] Here, we only consider assignments of k vertex variables as φ is constrained to have only those as free variables. For MSO formulas, we would need to consider set variables, but this is out of scope for our work.

The weak homomorphism h maps a set $A \subseteq V(G)^k$ of independent sets to a vector $a \in \mathbf{Z}^{\mathbf{N}^\ell}$. The entry $h(A)_{(24,96)}$ tells us the number of independent sets in A that have weight 24 and 96 w.r.t. c and c'.

Before we apply the result about MSO-evaluation problems, discuss the complexity of the operations on ℓ-WeightedSolCount. Even though elements of U have a priori infinite size, in our application their size will be bounded. The weights in the graph are finite, even bounded by $O(n)$ by Lemma 3. Thus, the highest weight possible of a k-vertex tuple is $O(kn)$ which we will denote by N. So let us assume we are given two elements a, b from ℓ-WeightedSolCount, which are N-*bounded*. That is, $a_w = 0$ if $\|w\|_\infty > N$. Such vectors can be represented naturally in $O(N^\ell)$ space. Then the addition $a \oplus b$ trivially needs time $O(N^\ell)$. As $a \otimes b$ is a convolution, it can be computed in time $O(\ell N^\ell \log N)$ using DFT [28].

The following lemma follows from Counting ℓ-Weighted k-Set Problem being an MSO-evaluation problem and applying the result of Courcelle and Mosbah.

▶ **Lemma 5.** *Assume we are given $k, \ell \in \mathbf{N}$, a quantifier-free first-order formula $\omega(x_1 \dots x_k)$, a treedepth-decomposition of a graph G of depth d and weight functions $c_i^{(j)} \colon V(G) \to \mathbf{Z}$ with $|C^{(j)}(v_1, \dots, v_k)| \le N$ for $j \in [\ell]$ and $i \in [k]$. Then, we can solve the Counting ℓ-Weighted k-Set Problem with G, ω and $c_i^{(j)}$ as input in time $f(|\omega|, d)nN^\ell \log N$ for some function f.*

Proof. First, from a treedepth decomposition $\mathcal{T}$ of depth d we can easily construct a tree decomposition of width d. Also, we know the ℓ-Weighted k-Set Problem is an MSO-evaluation problem, which can be expressed by evaluating $h(sat(\varphi, G))$ as described above in the definition of ℓ-WeightedSolCount. Hence, we can apply Proposition 9 on G, φ and the weight functions $c_i^{(j)}$ for this problem. This yields us an algorithm solving the given problem in time $O(f(d)|\mathcal{T}| \cdot \eta) = O(f(d)n\ell N^\ell \log N)$ where η is the complexity of the operations in the semiring ℓ-WeightedSolCount. Indeed, η is bounded by $O(\ell N^\ell \log N)$ as the vectors appearing during the computation are N-bounded and the image of $C^{(j)}$ is bounded by N as well. ◀

4.3 Run Time Improvements for $\le$-Relations

For the special case, where $\mathbf{P}$ consists of boolean combinations of ℓ $\le$-relations, i.e., $\#y\, \varphi(y\bar{x}) \ge t$ for constant $t \in N$, we shave of a factor of n in the running time.

This case can easily be reduced to a bounded number of conjunctions of counting terms $\#y\, \varphi(y\bar{x}) \ge t$ of length at most ℓ, by transforming the boolean combination into disjunctive normal form (DNF). Then each conjunctive clause is regarded separately by pushing the existential quantifier into the disjunction.

▶ **Lemma 11.** *Let $\mathcal{C}$ be a graph class of bounded expansion. There exists a function f such that for all graphs $G \in \mathcal{C}$, first-order formulas $\varphi_1(y\bar{x}), \dots, \varphi_\ell(y\bar{x})$, and numbers $t_1, \dots, t_\ell \in \mathbf{N}$ one can decide in time $O(f(|\varphi_1| + \cdots + |\varphi_\ell|)n^\ell \operatorname{polylog}(n))$ whether*

$$G \models \exists x_1 \dots \exists x_k (\#y\, \varphi_1(y\bar{x}) \ge t_1 \wedge \cdots \wedge \#y\, \varphi_\ell(y\bar{x}) \ge t_\ell).$$

Proof. In the proof of Theorem 6, the subroutine of Lemma 5 computes semiring operations $O(n)$ times which determines the overall run time. The run time of such operations is almost linear in the table size of the dynamic program. There, the number of tuples was assumed to be the worst-case, namely $O(N^\ell)$, resulting in a running time for the semiring operations of $O(\mathrm{poly}(\ell)N^\ell \,\mathrm{polylog}(N))$. In our case, we do not need all tuples, only the Pareto-optimal ones. That is, for any (partial) solutions $\bar{u}, \bar{v} \in V(G)^k$ if $\bar{u}$ dominates $\bar{v}$, that is, $[\![\#y\, \varphi_i(y\bar{u})]\!]^G \geq [\![\#y\, \varphi_i(y\bar{v})]\!]^G$ for all $i \in [\ell]$, then $\bar{v}$ can be disregarded further on.

The number of Pareto-optimal solutions in the domain $\{-N, \dots, +N\}^\ell$ is bounded by $O(N^{\ell-1})$. (Fix values for the first $\ell - 1$ dimensions, then there can be at most one Pareto-optimal tuple agreeing with the first $\ell - 1$ values.)

Thus, the application of the semiring operations in Lemma 5 takes only $O(\ell N^{\ell-1} \log N)$ time instead of $O(\ell N^\ell \log N)$. Continuing the run time analysis as in Lemma 5 and Theorem 6, we get the desired result. ◀

Note that this only improves the run time for the decision problem for this fragment. This approach does not work for the counting problem.

We can achieve the same run time improvement for *modulo counting*. If the predicate is a boolean combination of modulo counting terms, that is, the predicate checks if $\#y\, \varphi(y\bar{x}) \equiv a \pmod b$ then both the decision and even the counting problem is in time $O(f(|\varphi_1| + \cdots + |\varphi_\ell|)n^\ell \,\mathrm{polylog}(n))$ for classes of bounded expansion. However, Nešetřil, Ossona de Mendez and Siebertz [25] showed recently an even stronger result; they achieve a linear fpt time algorithm for the model checking problem of the logic FO+Mod which contains arbirarily nested modulo counting quantifiers.

4.4 Lifting to Counting Tuples $\#(y_1, \dots, y_p)$

The algorithmic results (Theorem 6, Corollary 7, and Lemma 11) can be lifted to counting *tuples*, that is, to counting quantifiers $\#\bar{y}\, \varphi(\bar{y}\bar{x})$ that are also part of FOC($\mathbf{P}$) (where $\bar{y}$ is tuple of variables $(y_1, y_2, \dots, y_p)$).[6] This comes at a cost: The polynomial factor of the run time increases to $n^{(\ell+1)p}$.

▶ **Corollary 12.** *Let $\mathcal{C}$ be a class of bounded expansion. There exists a function f such that for a given graph $G \in \mathcal{C}$, first-order formulas $\varphi_1(\bar{y}\bar{x}), \dots, \varphi_\ell(\bar{y}\bar{x})$ and all computable relations $P \subseteq \mathbf{N}^\ell$ one can decide*

$$G \models \exists x_1 \dots \exists x_k P(\#\bar{y}\, \varphi_1(\bar{y}\bar{x}), \dots, \#\bar{y}\, \varphi_\ell(\bar{y}\bar{x}))$$

in time $O\big(f(|\varphi_1| + \cdots + |\varphi_\ell|)n^{(\ell+1)|\bar{y}|} \,\mathrm{polylog}(n) + rn^\ell\big)$ where r is time needed to decide membership in P.

We achieve this result by adapting both the statements made in this paper and their dependencies, which originate from [10]. This adaptation is quite straightforward; it suffices to replace every occurrence of $\#y$ with the more general counting quantifier $\#\bar{y}$. Additionally, the monovariate weight functions $c_i^{(j)}$ must be extended to $c_I^{(j)}\colon V(G)^p \to \mathbf{Z}$, where $I \in V(G)^p$.

In the statements [9, Lemma 2, 3, 6, 15, and Theorem 2, 4], we apply the same changes. Furthermore, if, for some $i \in [|\bar{x}|]$, clauses are restricted to the form $\tau(y) \wedge \psi(\bar{x}) \wedge f(y) = g(x_i) \wedge \Delta^{\neq}(y\bar{x})$ (with non-empty or empty $\Delta^{\neq}$) in the statements or proofs, we extend them to the form $\tau(\bar{y}) \wedge \psi(\bar{x}) \wedge f_1(y_1) = g_1(x_{i_1}) \wedge \cdots \wedge f_p(y_p) = g_p(x_{i_p}) \wedge \Delta^{\neq}(y\bar{x})$ for some $I = (i_1, \dots, i_p) \in [k]^p$. Note that the runtime increase in adapting [9] mostly occurs in [9, Lemma 6], where its running time increases to $\|G\|^p$.

[6] We want to thank Michał Pilipczuk as he brought to our attention that this approach may likely extend to counting tuples which we formerly thought to be impossible.

The same applies to Definition 4 and Lemmas 3 and 5 in Section 4.1. It's worth mentioning that the N in Theorem 6 is now $O(n^p)$ instead of being linear in n, which explains the increase in time complexity.

As these changes are rather trivial and clutter the presentation of the results, we decided to omit explicit proofs here.

5 Hardness

We have seen that formulas ψ of the form $\exists x_1 \ldots \exists x_k\, P(\#y\, \varphi_1(\bar{x}, y), \ldots, \#y\, \varphi_\ell(\bar{x}, y))$ can be evaluated in time $O(f(|\psi|)n^{\ell+1} \operatorname{polylog}(n))$ on graph classes of bounded expansion. We now show that this cannot be improved by more than an almost linear factor in n under SETH.

For this, let us recall the Subset Sum Problem, which is often used as a starting point in fine-grained complexity theorem. Here, we need a recent conditional lower bound for this problem.

▶ **Definition 13** (k-Sum). Given n integers $x_1, \ldots, x_n \in \mathbf{N}$ and a target value T, the task in the SubsetSum problem is to decide whether there is a subset of the numbers above which sums to T. For the k-Sum problem, the task for the same instance is to decide where there are k numbers which add up to T.

▶ **Proposition 14** ([1]). *Assuming SETH, for any $\varepsilon > 0$ there exists a $\delta > 0$ such that Subset Sum is not in time $O(T^{1-\varepsilon}2^{\delta n})$, and k-Sum is not in time $O(T^{1-\varepsilon}n^{\delta k})$.*

Note that δ does not depend on k, only on ε.

We will show a conditional lower bound of the following problem on star forests. Note that those graphs have treedepth 2.

▶ **Definition 15** (ℓ-Variate k-Satisfaction). Given a graph G, quantifier-free FO formulas $\varphi_1(x_1, \ldots, x_k, y), \ldots, \varphi_\ell(x_1, \ldots, x_k, y)$ and integers $w_1, \ldots, w_\ell$ the problem ℓ-Variate k-Satisfaction asks whether

$$G \models \exists x_1 \ldots \exists x_k \bigwedge_{i=1}^{\ell} \varphi_i(x_1, \ldots, x_k, y) = w_i.$$

First, we make a simple observation that any number can be uniquely represented in any (natural) base. The following is tailored to our use case.

▶ **Observation 16.** *Observe, that for every $\ell, T \in \mathbf{N}$, every number $0 \le x \le T$ can be uniquely expressed as $x = \sum_{j=1}^{\ell} a_j \cdot \tau^{j-1}$ for some $a_j \in \{0, \ldots, \tau - 1\}$ and where $\tau = \lceil \sqrt[\ell]{T} \rceil$.*

Now, we relax the conditions on the base and allow an "overlap". As an example, consider the number 35 in base-10. If we allowed the digits to range from 0 to 15 (represented by $0, \ldots, 9, A, \ldots, F$), 35 can then be expressed also by 2F ($= 2 \cdot 10^1 + 15 \cdot 10^0$ in "base-10 with A-F"). The representation is then of course not unique, but the number of such representations is small.

▶ **Lemma 17.** *Consider Observation 16 but where the a_j are allowed to range from 0 to $k\tau - 1$ instead. Then the number of such representations of x is bounded by k^ℓ.*

Proof. We prove the claim by induction over ℓ. First, let us notice that $\sum_{j=1}^{\ell} a_j \cdot \tau^{j-1} < k\tau^\ell$ for $a_j \in \{0, \ldots, k\tau - 1\}$.

For the induction, let $x \in \{0, \ldots, k\tau^\ell\}$. Consider a representation of x with $x = \sum_{j=1}^{\ell} a_j \cdot \tau^{j-1}$ and $a_j \in \{0, \ldots, k\tau - 1\}$ for all $j \in [\ell]$.

For $\ell = 1$, $x = a_1$ holds. Hence, there is only one representation.

For $\ell > 1$, let us consider furthermore $y = x - a_\ell \tau^{\ell-1}$. Then, by our first note $y = \sum_{j=1}^{\ell-1} a_j \cdot \tau^{j-1} < k\tau^{\ell-1}$. By the induction hypothesis there are at most $k^{\ell-1}$ representations of y in this form (where j ranges from 1 to $\ell - 1$). There are at most k valid choices for a_ℓ, as $x - a_\ell \tau^{\ell-1}$ has to fall into the range $\{0, \dots, k\tau^{\ell-1} - 1\}$. Thus, there are $k \cdot k^{\ell-1} = k^\ell$ possibilities to represent x as described. ◀

▶ **Theorem 18.** *Assuming SETH, for any $\varepsilon > 0$ and $\ell \in \mathbf{N}$ the ℓ-*Variate *k-*Satisfaction *on star forests is not in time $f(\ell, k) \cdot |G|^{\ell-\varepsilon}$ for all $k \in \mathbf{N}$ and functions f.*

A few words about the implications of the theorem are in order: The result does *not* rule out that there exists a pair of ℓ and k and an $\varepsilon > 0$ such that ℓ-Variate k-Satisfaction can be solved in time $f(k, \ell)|G|^{\ell-\varepsilon}$. It only says that (for any fixed ℓ) there cannot be a (family of) algorithms which achieves this "form" of running time. Indeed, if one looks closer into the proof, one can show that for every fixed ℓ there can be at most finitely many k where a faster running time can be achieved. Note especially that for $k < \ell$, one can achieve a running time of $f(k, \ell)n^k < f(k, \ell)n^{\ell+1}$ by essentially brute-forcing all solution candidates.

Before we come to the proof, let us consider the (hopefully easier) case of ℓ being 1 or 2. We can express a given k-Sum instance as a graph where every number x is expressed as a star with x endpoints. Now a set of k numbers from the k-sum instance adds up to T iff there are k stars with T endpoints in total.

Generalizing this to $\ell = 2$, we change the construction such that a star does not have x endpoints but $\lfloor x/\sqrt{T} \rfloor$ blue endpoints and $x - \lfloor x/\sqrt{T} \rfloor$ red endpoints, which represent the most significant bits (MSB) and least significant bits (LSB) of x respectively. Now, a set of k numbers add up to T iff the k corresponding stars have $\lfloor T/\sqrt{T} \rfloor$ blue and $T - \lfloor T/\sqrt{T} \rfloor$ red neighbors in total, where the numbers are the MSB and the LSB of T respectively. Due to carryovers the above statement does not hold technically. This issue can be fixed with a small overhead by guessing the form of the carryovers. The number of possible carryovers is small.

In the proof, the technique is generalized to all $\ell \in \mathbf{N}$ and the technicalities concerning carryovers are addressed as well.

Proof. Let $k, \ell \in \mathbf{N}$. Consider a k-Sum instance with n numbers $x_1, \dots, x_n$ and a target value T. We reduce this to a ℓ-Variate k-Satisfaction instance, where $|G| = \Theta(n\ell\sqrt[\ell]{T})$ and where G is a star forest with n components. The reduction represents the numbers of the k-Sum instance as a graph G with colors $[\ell]$ and each number x_i as a star in G with $a_{i,j}$ many endpoints of color $j \in [\ell]$ where $x_i = \sum_{j=1}^{\ell} a_{i,j} \cdot \tau^{j-1}$, $\tau = \lceil \sqrt[\ell]{T} \rceil$ and $a_{i,j} \in \{0, \dots, \tau - 1\}$ (see Observation 16). Then, there exist k numbers in the k-Sum instance which add up to T if and only if there are k vertices $v_1, \dots, v_k$ in G with (in total) B_j many neighbors of color j for every $j \in [\ell]$ such that $T = \sum_{j=1}^{\ell} B_j \tau^{j-1}$.

After having constructed the graph, let us look at the formula. Each number B_j ranges from 0 to $k(\tau - 1)$ as each vertex v_i has at most $\tau - 1$ neighbors of a given color. Let $\mathcal{T}$ be sets of ℓ-tuples $(w_1, \dots, w_\ell) \in \{0, \dots, k(\tau - 1)\}^\ell$ with $T = \sum_{j=1}^{\ell} w_j \cdot \tau^{j-1}$. By Lemma 17 the size of $\mathcal{T}$ is bounded in a function of k and ℓ. Let $(w_1, \dots, w_\ell) \in \mathcal{T}$. Then, the following formula expresses that there are k (central) vertices whose neighbors of color $j \in [\ell]$ (expressed by the predicate C_j) add up to w_j

$$\psi_{(w_1,\dots,w_\ell)} \equiv \exists x_1 \dots \exists x_k \bigwedge_{j=1}^{\ell} \left(\#y \left(C_j(y) \wedge \bigvee_{i'=1}^{k} y \sim x_{i'} \right) = w_j \right).$$

Thus, G together with the formula above and $(w_1, \dots, w_\ell)$ is the instance of ℓ-Variate k-Satisfaction constructed above.

Hence, for each tuple $(w_1, \dots, w_\ell) \in \mathcal{T}$ we check whether G satisfies $\psi_{(w_1,\dots,w_\ell)}$. If this is the case for some tuple, the k-Sum instance is accepted, otherwise it is not.

The size of the resulting graph G is $n\ell\lceil\sqrt[\ell]{T}\rceil$, the size of each $\psi_{(w_1,\dots,w_\ell)}$ is $O(k\ell)$, the construction of a single instance takes linear time in $|G|$ and $|\psi|$. This construction is repeated $|\mathcal{T}| = f(k,\ell)$ times for each $\psi_{(w_1,\dots,w_\ell)}$.

To show the theorem statement, assume for contradiction that there exist $\ell \in \mathbf{N}$ and $\varepsilon > 0$ such that for every $K \in \mathbf{N}$ there exists $k \geq K$ such that the ℓ-Variate k-Satisfaction on star forests can be solved in time $O(|G|^{\ell(1-\varepsilon)} f(l,k))$ some function f. We consider any k-Sum instance for k. It has n numbers and a target value T. As we have seen above, it can be reduced to $|\mathcal{T}|$ instances of ℓ-Variate k-Satisfaction on star forests of size $n\ell\sqrt[\ell]{T}$.

Then, k-Sum could be solved in time

$$\Theta(|G| + |\varphi|) + |\mathcal{T}|O(|G|^{\ell(1-\varepsilon)} f(k,\ell)) = O((n\ell\sqrt[\ell]{T})^{\ell(1-\varepsilon)} f(k,\ell)) = O(T^{1-\varepsilon} n^{\ell(1-\varepsilon)} f(k,\ell)).$$

Recall Proposition 14 and let us assume that SETH holds. The above run time contradicts the non-existence of an algorithm for k-Sum with run time $T^{1-\varepsilon} n^{\delta k}$ if $\ell(1-\varepsilon) < \delta k$. As δ depends only on ε, the inequality can be satisfied by "big enough" k. To be more precise, it holds for $k \geq \ell(1-\varepsilon)/\delta$. Thus, we get that the existence of an algorithm for ℓ-Variate k-Satisfaction in time $O(|G|^{\ell(1-\varepsilon)} f(l,k))$ contradicts SETH. ◀

From this result, we get directly a lower bound for model checking of formulas with inequality, that is, $\exists x_1 \dots \exists x_k \bigwedge_{i=1}^{\ell} \#y\, \varphi_i(y\bar{x}) \geq N_i$. Note that by Lemma 11 we have an algorithm with a running time of $f(k,\ell)n^\ell \operatorname{polylog} n$.

▶ **Corollary 19.** *Assuming SETH, for any $\varepsilon > 0$ and $\ell \in \mathbf{N}$ the ℓ-Variate k-Satisfaction with inequalities on star forests is not in time $O(f(\ell,k)|G|^{1/2\ell-\varepsilon})$ for all k and functions f.*

Proof. Note that an equality $\#y\, \varphi(y\bar{x}) = N$ can be expressed by two inequalities $\#y\, \varphi(y\bar{x}) \leq N \wedge \#y\, \varphi(y\bar{x}) \geq N$, and equivalently by $\#y\, \neg\varphi(y\bar{x}) \geq |G| - N \wedge \#y\, \varphi(y\bar{x}) \geq N$. A formula $\exists x_1 \dots \exists x_k \bigwedge_{i\in[\ell]} \#y\, \varphi_i(y\bar{x}) = N_i$ from ℓ-Variate k-Satisfaction with ℓ counting terms is then equivalent to the formula $\exists x_1 \dots \exists x_k \bigwedge_{i\in[\ell]} \#y\, \varphi_i(y\bar{x}) \leq N_i \wedge \#y\, \neg\varphi_i(y\bar{x}) \leq |G| - N_i$ with 2ℓ counting terms using inequalities. The result is weaker in the degree of the polynomial by a factor of $1/2$, as the number ℓ of counting terms doubles when going from equalities to inequalities as discussed above. ◀

Lower Bounds for Other Fragments

In the following we show that our algorithmic results cannot be generalized to slightly more general fragments. Firstly, consider the existential fragment with nested counting, that is, we add another counter formula *inside* the counting formulas considered in Section 4. Secondly, we show that one cannot add even one universal quantifier between the existential quantifiers and the counting quantifier.

Here, we regard the equality predicate $= t$ only (where t is a constant which depends on the graph). This can easily be extended to inequality predicates by replacing $\#y\, \varphi = t$ with both $\#y\, \varphi \geq t \wedge \#y\, \varphi \leq t$. Also note that the constant t depends on the graph. It is also possible to construct a formula which is independent of the graph. However, this formula needs a comparison between two counting terms instead.

The reduction is based on the reduction from [17, Theorem 4.1] and [9, Lemma 16]. The proof is omitted and can be found in the appendix.

▶ **Theorem 20.** *The model checking problem on forests of depth 2 is* W[1]*-hard for* $FO(\{=\})$*-formulas* ψ *of the form*

1. $\exists x_1 \dots \exists x_k \forall z \# y\, \varphi(\bar{x}yz) = t$
2. $\exists x_1 \dots \exists x_k \# z\, [\# y\, \varphi(\bar{x}yz) = t] = s$
3. $\exists x_1 \dots \exists x_k \forall z \forall z' \# y\, \varphi(\bar{x}yzz') = t$

where φ *is a quantifier-free FO-formula.*

Moreover, under ETH there is no algorithm with a running time of $n^{o(\sqrt{|\psi|})}$ *for formulas of the form 1 and 2, and* $n^{o(|\psi|)}$ *for formulas of form 3.*

6 Concluding Remarks

We have shown that on classes of bounded expansion, we can solve the query evaluation and query counting problem of formulas of the form $\exists x_1 \dots \exists x_k\, P(\# y\, \varphi_1(\bar{x}, y), \dots, \# y\, \varphi_\ell(\bar{x}, y))$ in time $f(|\psi|)n^{\ell+1} \operatorname{polylog} n$, and it cannot be improved to $f(|\psi|)n^{\ell-\varepsilon}$ for any $\varepsilon > 0$. For the case of $\leq$-constraints we improved the running time for the query evaluation problem to $f(|\psi|)n^{\ell} \operatorname{polylog} n$.

It would be interesting to close the gap between the lower bound of Theorem 18 and the upper bound of Theorem 6. One approach could be improving the algorithm of Lemma 5. For such approaches it is certainly needed to maintain a table of size $\widetilde{O}(n^{\ell})$ in the worst case. However, it could be possible that in many nodes of the treedepth decomposition, the maintained tables have considerably smaller size, e.g., at the leaves and that together with output sensitive FFT algorithms, e.g., ones that have almost linear run time in the size of the output [23], one could achieve a better run time amortized across all n nodes of the treedepth decomposition. But we do not know if the table size during the dynamic program on the treedepth decomposition permits an improved run time analysis.

Also, for the case that **P** consists of a boolean combination of constraints lower bounding the counting terms by a constant, the gap between lower and upper bound is quite large. As the Pareto-front for those should have size $\Theta(n^{\ell-1})$ it seems that the lower bound could be improved. However, the numerical problems known to us which have strong conditional lower bounds as subset sum are not based on such inequalities.

Moreover, extending our results to the class of nowhere dense graphs would be certainly interesting. The results of [10] for FO($\{>0\}$) are already partially lifted to such classes [8]. They used very different techniques, however. Especially, they did not use (or show) an equivalent result to Lemma 3. On top of that , recently the non-existence of quantifier elimination for FO on nowhere dense classes was shown [15] which implies that different techniques are *needed.*

References

1 Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. SETH-based lower bounds for subset sum and bicriteria path. *ACM Trans. Algorithms*, 18(1):6:1–6:22, 2022. doi:10.1145/3450524.

2 B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1):49–82, 1993. doi:10.1016/0304-3975(93)90064-Z.

3 Bruno Courcelle. The monadic second-order logic of graphs I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

4 Bruno Courcelle, Johann A Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

5 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

6 Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 270–279. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.31.

7 Jan Dreier, Nikolas Mählmann, and Sebastian Siebertz. First-order model checking on structurally sparse graph classes. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 567–580. ACM, 2023. doi:10.1145/3564246.3585186.

8 Jan Dreier, Daniel Mock, and Peter Rossmanith. Evaluating restricted first-order counting properties on nowhere dense classes and beyond. In *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 43:1–43:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ESA.2023.43.

9 Jan Dreier and Peter Rossmanith. Approximate evaluation of first-order counting queries. *CoRR*, abs/2010.14814, 2020. doi:10.48550/arXiv.2010.14814.

10 Jan Dreier and Peter Rossmanith. Approximate evaluation of first-order counting queries. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1720–1739. SIAM, 2021. doi:10.1137/1.9781611976465.104.

11 Zdeněk Dvořák, Daniel Kráľ, and Robin Thomas. Deciding first-order properties for sparse graphs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 133–142. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.20.

12 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995.

13 Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206, 2001. doi:10.1145/504794.504798.

14 Robert Ganian, Petr Hliněný, Alexander Langer, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. Lower bounds on the complexity of MSO_1 model-checking. *J. Comput. Syst. Sci.*, 80(1):180–194, 2014. doi:10.1016/j.jcss.2013.07.005.

15 Mario Grobler, Yiting Jiang, Patrice Ossona de Mendez, Sebastian Siebertz, and Alexandre Vigny. Discrepancy and sparsity, 2021. doi:10.48550/arXiv.2105.03693.

16 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

17 Martin Grohe and Nicole Schweikardt. First-order query evaluation with cardinality conditions. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 253–266. ACM, 2018. doi:10.1145/3196959.3196970.

18 Wojciech Kazana and Luc Segoufin. First-order queries on classes of structures with bounded expansion. *Log. Methods Comput. Sci.*, 16(1), 2020. doi:10.23638/LMCS-16(1:25)2020.

19 Joachim Kneis, Daniel Mölle, and Peter Rossmanith. Partial vs. complete domination: t-dominating set. In *SOFSEM 2007: Theory and Practice of Computer Science, 33rd Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 20-26, 2007, Proceedings*, volume 4362 of *Lecture Notes in Computer Science*, pages 367–376. Springer, 2007. doi:10.1007/978-3-540-69507-3_31.

20 Stephan Kreutzer and Siamak Tazari. Lower bounds for the complexity of monadic second-order logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 189–198. IEEE Computer Society, 2010. doi:10.1109/LICS.2010.39.

21 Dietrich Kuske and Nicole Schweikardt. First-order logic with counting. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005133.

22 Alexander Langer, Felix Reidl, Peter Rossmanith, and Somnath Sikdar. Practical algorithms for MSO model-checking on tree-decomposable graphs. *Comput. Sci. Rev.*, 13-14:39–74, 2014. doi:10.1016/j.cosrev.2014.08.001.

23 Vasileios Nakos. Nearly optimal sparse polynomial multiplication. *IEEE Trans. Inf. Theory*, 66(11):7231–7236, 2020. doi:10.1109/TIT.2020.2989385.

24 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

25 Jaroslav Nešetřil, Patrice Ossona de Mendez, and Sebastian Siebertz. Modulo-counting first-order logic on bounded expansion classes. *Discrete Mathematics*, page 113700, 2023. doi:10.1016/j.disc.2023.113700.

26 Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struct. Comput. Sci.*, 6(6):505–526, 1996. doi:10.1017/s0960129500070079.

27 Szymon Toruńczyk. Aggregate queries on sparse databases. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 427–443. ACM, 2020. doi:10.1145/3375395.3387660.

28 Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013.