

Kernelization Lower Bounds through Colors and IDs*

Michael Dom[†]

Daniel Lokshтанov[‡]

Saket Saurabh[§]

Abstract

In parameterized complexity each problem instance comes with a parameter k , and a parameterized problem is said to admit a *polynomial kernel* if there are polynomial time preprocessing rules that reduce the input instance to an instance with size polynomial in k . Many problems have been shown to admit polynomial kernels, but it is only recently that a framework for showing the non-existence of polynomial kernels for specific problems has been developed by Bodlaender et al. [6] and Fortnow and Santhanam [17]. With few exceptions, all known kernelization lower bounds result have been obtained by directly applying this framework. In this paper we show how to combine these results with combinatorial reductions which use colors and IDs in order to prove kernelization lower bounds for a variety of basic problems. Below we give a summary of our main results. All results are under the assumption that the polynomial hierarchy does not collapse to the third level.

- We show that the STEINER TREE problem parameterized by the number of terminals and solution size k , and the CONNECTED VERTEX COVER and CAPACITATED VERTEX COVER problems do not admit a polynomial kernel. The two latter results are surprising because the closely related VERTEX COVER problem admits a kernel of with at most $2k$ vertices.
- Alon and Gutner obtain a $k^{poly(h)}$ kernel for DOMINATING SET IN H -MINOR FREE GRAPHS parameterized by $h = |H|$ and solution size k , and ask whether kernels of smaller size exist [3]. We partially resolve this question by showing that DOMINATING SET IN H -MINOR FREE GRAPHS does not admit a kernel with size polynomial in $k + h$.
- Harnik and Naor obtain a “compression algorithm” for the SPARSE SUBSET SUM problem [21]. We show that their algorithm is essentially optimal by showing that the instances cannot be compressed further.
- The HITTING SET and SET COVER problems are among the most studied problems in algorithmics. Both problems admit a kernel of size $k^{O(d)}$ when parameterized by solution size k and maximum set size d . We show that neither of them, along with the UNIQUE COVERAGE and BOUNDED RANK DISJOINT SETS problems, admits a polynomial kernel.

The existence of polynomial kernels for several of the problems mentioned above were open problems explicitly stated in the literature [3, 4, 19, 20, 26]. Many of our results also rule out the existence of compression algorithms, a notion similar to kernelization defined by Harnik and Naor [21], for the problems in question.

1 Introduction

Polynomial time preprocessing to reduce instance size is one of the most widely used approaches to tackle computationally hard problems. A natural question in this regard is how to measure

*A preliminary version of this paper appeared in the proceedings of ICALP 2009.

[†]Friedrich-Schiller-Universität Jena, 07743 Jena, Germany. michael.dom@uni-jena.de

[‡]University of Bergen, 5020 Bergen, Norway. daniello@ii.uib.no

[§]The Institute of Mathematical Sciences, Chennai 600 113, India. saket@imsc.res.in

the quality of preprocessing rules proposed for a specific problem. Parameterized complexity provides a natural mathematical framework to give performance guarantees of preprocessing rules. In parameterized complexity each problem instance comes with a parameter k and the parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm, called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k , while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. We refer to [13, 15, 27] for further introduction to parameterized complexity and kernelization.

Kernelization has been extensively studied, resulting in polynomial kernels for a variety of problems. Notable examples include a $2k$ vertex kernel for VERTEX COVER [9], a $355k$ vertex kernel for DOMINATING SET in planar graphs [2], which later was improved to a $67k$ vertex kernel [8], and a $O(k^2)$ kernel for FEEDBACK VERTEX SET [28] parameterized by the solution size. While positive kernelization results have appeared regularly over the last two decades, the first results establishing infeasibility of polynomial kernels for specific problems have appeared only recently. In particular, Bodlaender et al. [6] and Fortnow and Santhanam [17] have developed a framework for showing that a problem does not admit a polynomial kernel unless the polynomial hierarchy collapses to the third level ($PH = \Sigma_p^3$), a collapse which is deemed unlikely.

Previous Results. Bodlaender et al. [6] observed that their framework can be directly applied to show kernelization lower bounds for many parameterized problems, including LONGEST PATH and LONGEST CYCLE. To the authors’ best knowledge, the only non-trivial applications of this framework are in a recent result of Fernau et al. [14] showing that the DIRECTED MAX LEAF OUTBRANCHING problem does not have a polynomial kernel, and a result by Bodlaender et al. [7] showing that the DISJOINT PATHS and DISJOINT CYCLES problems do not admit a polynomial kernel unless $PH = \Sigma_p^3$.

Our Results & Techniques. At present, there are two ways of showing that a particular problem does not admit a polynomial kernel unless $PH = \Sigma_p^3$. One is to give a “composition algorithm” for the problem in question. The other is to reduce from a problem for which a kernelization lower bound is known to the problem in question, such that a polynomial kernel for the considered problem would transfer to a polynomial kernel for the problem we reduced from. Such a reduction is called a *polynomial parameter transformation* and was introduced by Bodlaender et al. [7]. In order to show our results, we apply both methods. First, we present a “cookbook” approach for showing kernelization lower bounds by using composition algorithms together with polynomial parameter transformations. We apply our approach to show that UNIQUE COVERAGE parameterized by solution size k and HITTING SET and SET COVER parameterized by solution size k and universe size $|U|$ do not admit polynomial kernels unless $PH = \Sigma_p^3$. These problems turn out to be useful starting points for polynomial parameter transformations. By reducing from these problems, we show that a variety of basic problems do not have a polynomial kernel. Below, we give a summary of our main results. All our results are under the assumption that $PH \neq \Sigma_p^3$ and unless explicitly stated otherwise, all the problems considered are parameterized by the solution size.

Connectivity and Covering Problems: We show that the SET COVER problem parameterized by solution size k and the size $|U|$ of the universe does not have a polynomial kernel. Using this result, we prove that STEINER TREE parameterized by the number of terminals and solution size k does not have a polynomial kernel, resolving an open problem stated in [4]. We proceed to show that the CONNECTED VERTEX COVER and CAPACITATED VERTEX COVER problems do not admit a polynomial kernel for the parameter k . The existence of polynomial kernels for these

problems was an open problem explicitly stated in the literature [19, 20], and the negative answer to these questions is bit unexpected because the closely related VERTEX COVER problem admits a vertex kernel of size $2k$. Finally, we show that the BOUNDED RANK DISJOINT SETS and UNIQUE COVERAGE problems do not admit a polynomial kernel. The latter result resolves an open problem of Moser et al. [26].

Domination and Transversals: We show that the HITTING SET problem parameterized by solution size k and the size $|U|$ of the universe does not have a polynomial kernel. This implies that the DOMINATING SET problem parameterized by solution size k and the size of a minimum vertex cover of the input graph does not admit a polynomial kernel. The latter result in turn implies that DOMINATING SET IN H -MINOR FREE GRAPHS parameterized by $h = |H|$ and k does not admit kernel with size polynomial in $k + h$, partially resolving an open problem by Alon and Gutner [3], who obtain a $k^{\text{poly}(h)}$ kernel for DOMINATING SET IN H -MINOR FREE GRAPHS and ask whether kernels of smaller size exist. Another implication of our results is that the HITTING SET and SET COVER problems parameterized by solution size k and maximum set size d do not have a kernel polynomial in k, d . Both HITTING SET and SET COVER admit a $k^{O(d)}$ kernel [1], and our results can help drawing the tractability borderline for these basic problems.

Numeric Problems: Harnik and Naor obtain a “compression algorithm” for the SPARSE SUBSET SUM problem [21]. Essentially, Harnik and Naor show that if the input instance to SUBSET SUM is a relatively small set of huge numbers, the instance can be reduced. We show that in contrast, if the input instance is a huge set of relatively small numbers, the instance cannot be reduced.

It is implicit from the discussion in [17] that for a large class of parameterized problems the notions of kernelization and *compression algorithms*, defined by Harnik and Naor [21], are equivalent. This result, together with our results about the non-existence of polynomial kernels, rule out the existence of compression algorithms for many problems. These include STEINER TREE, CONNECTED VERTEX COVER, CAPACITATED VERTEX COVER and UNIQUE COVERAGE, among others.

Finally, we would like to point out that HITTING SET and SET COVER are among the basic hard problems in parameterized complexity, and that many W -hardness reductions start from these problems. In order to obtain our results we give several polynomial parameter transformations starting from the bounded universe version of these problems. It seems like the bounded universe version of these problems could turn out to be as useful for showing kernelization lower bounds, as their unrestricted counterparts have been to show W -hardness.

Organization of the paper: The paper is organized as follows. In the next section we collect all the definitions and results which we make use of in the paper. In Section 3 we give a systematic approach to prove kernelization and compressibility lower bounds using composition algorithms. Sections 4, 5 and 6 form the main technical part of the paper where we show the non-existence of polynomial kernels for many different problems, either by taking the approach described in Section 3 or by showing a polynomial parameter transformation. Our results are thematically organized under connectivity and covering problems, domination and transversals and numeric problems. In Section 7, we show a connection between compression algorithms and kernelization and rule out compression algorithms for various problems. Finally we conclude with some remarks and open problems in Section 8. Our appendix includes a compendium of the problems considered in the paper. There one can find definitions, known results and our contributions.

2 Preliminaries

A parameterized problem L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ . An instance of a parameterized problem consists of (x, k) , where k is called the parameter. A central notion in parameterized complexity is *fixed parameter tractability (FPT)*, which means for a given instance (x, k) solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. The notions of *kernelization* and *composition* are formally defined as follows.

Definition 2.1. A *kernelization algorithm*, or in short, a *kernel* for a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that (a) $(x, k) \in Q$ if and only if $(x', k') \in Q$ and (b) $|x'| + k' \leq g(k)$, where g is an arbitrary computable function. The function g is referred to as the size of the kernel. If g is a polynomial function then we say that Q admits a polynomial kernel.

Definition 2.2 (Composition [6]). A composition algorithm for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, k), \dots, (x_t, k))$, with $(x_i, k) \in \Sigma^* \times \mathbb{N}^+$ for each $1 \leq i \leq t$, uses time polynomial in $\sum_{i=1}^t |x_i| + k$, and outputs $(x', k') \in \Sigma^* \times \mathbb{N}^+$ with (a) $(x', k') \in L \iff (x_i, k) \in L$ for some $1 \leq i \leq t$ and (b) k' is polynomial in k . A parameterized problem is compositional if there is a composition algorithm for it.

A composition algorithm as described in Definition 2.2 is also called an “Or-Composition”. We utilize a recent result of Bodlaender et al. [6] and Fortnow and Santhanam [17] that states that any compositional parameterized problem does not have a polynomial kernel unless the polynomial hierarchy collapses to the third level. We define the *unparameterized version* \tilde{L} of a parameterized problem L as the language $\tilde{L} = \{x\#1^k \mid (x, k) \in L\}$, that is, the mapping of parameterized problems to unparameterized problems is done by mapping an instance (x, k) to the string $x\#1^k$, where 1 is an arbitrary fixed letter in Σ and $\# \notin \Sigma$.

Theorem 2.1 ([6, 17]). *Let L be a compositional parameterized problem whose unparameterized version \tilde{L} is NP-complete. Then, unless $\text{PH} = \Sigma_p^3$, there is no polynomial kernel for L .*

Finally we define the notion of *polynomial parameter transformations*.

Definition 2.3 ([7]). Let P and Q be parameterized problems. We say that P is polynomial parameter reducible to Q , written $P \leq_{ppt} Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ and a polynomial p , such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ (a) $(x, k) \in P$ if and only if $(x', k') = f(x, k) \in Q$ and (b) $k' \leq p(k)$. The function f is called *polynomial parameter transformation*.

Proposition 2.1 ([7]). *Let P and Q be the parameterized problems and \tilde{P} and \tilde{Q} be the unparameterized versions of P and Q respectively. Suppose that \tilde{P} is NP-hard and \tilde{Q} is in NP. Furthermore if there is a polynomial parameter transformation from P to Q , then if Q has a polynomial kernel then P also has a polynomial kernel.*

Proposition 2.1 shows how to use polynomial parameter transformations to show kernelization lower bounds. A notion similar to polynomial parameter transformation was independently used by Fernau et al. [14] albeit without being explicitly defined.

We close with some definitions from graph theory. Let $G = (V, E)$ be a graph. For a vertex v in G , we write $N_G(v)$ to denote the set of v 's neighbors in G , and we write $\deg_G(v)$ to denote the *degree* of v , that is, the number of v 's neighbors in G . If it is clear from the context which graph is meant, we write $N(v)$ and $\deg(v)$, respectively, for short. A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. The subgraph G' is called an *induced subgraph* of G if $E' = \{\{u, v\} \in E \mid u, v \in V'\}$, in this case, G' is also called the subgraph *induced by V'* and denoted with $G[V']$. A vertex v *dominates* a vertex u if $u \in N(v)$.

3 A Systematic Approach to Prove Kernelization Lower Bounds

In this section we describe a “cookbook” for showing kernelization lower bounds. To show that a problem does not admit a polynomial size kernel we go through the following steps.

1. Find a suitable parameterization of the problem considered. Quite often parameterizations that impose extra structure make it easier to give a composition algorithm.
2. Define a suitable colored version of the problem. This is in order to get more control over how solutions to problem instances can look.
3. Show that the unparameterized version of the considered problem is in NP and that the unparameterized version of the colored version of the problem is NP-hard.
4. Give a polynomial parameter transformation from the colored to the uncolored version. This will imply that if the uncolored version has a polynomial kernel then so does the colored version. Hence kernelization lower bounds for the colored version directly transfer to the original problem.
5. Show that the colored version parameterized by k is solvable in time $2^{k^c} \cdot n^{O(1)}$ for a fixed constant c .
6. Finally, show that the colored version is compositional and thus has no polynomial kernel. To do so, proceed as follows.
 - (a) If the number of instances in the input to the composition algorithm is at least 2^{k^c} then running the parameterized algorithm on each instance takes time polynomial in input size. This automatically yields a composition algorithm.
 - (b) If the number of instances is less than 2^{k^c} , every instance receives a unique identifier. Notice that in order to uniquely code the identifiers (ID) of all instances, k^c bits per instance is sufficient. The IDs are coded either as an integer, or as a subset of a $\text{poly}(k)$ sized set.
 - (c) Use the coding power provided by colors and IDs to complete the composition algorithm.

In the following sections we show how to apply this approach to show kernelization lower bounds for a variety of problems.

4 Connectivity and Covering Problems

4.1 Set cover, Steiner Tree, and Variants of Vertex Cover

The problems STEINER TREE, CONNECTED VERTEX COVER (CONVC), CAPACITATED VERTEX COVER (CAPVC), and SMALL UNIVERSE SET COVER are defined as follows. In STEINER TREE we are given a graph $G = (T \cup N, E)$ and an integer k and asked for a vertex set $N' \subseteq N$ of size at most k such that $G[T \cup N']$ is connected. In CONVC we are given a graph $G = (V, E)$ and an integer k and asked for a vertex cover of size at most k that induces a connected subgraph in G . A *vertex cover* is a set $C \subseteq V$ such that each edge in E has at least one endpoint in C . The problem CAPVC takes as input a graph $G = (V, E)$, a capacity function $\text{cap} : V \rightarrow \mathbb{N}^+$ and an integer k , and the task is to find a vertex cover C and a mapping from E to C in such a way that at most $\text{cap}(v)$ edges are mapped to every vertex $v \in C$. Finally, an instance of SMALL UNIVERSE SET COVER consists of a set family \mathcal{F} over a universe U with $|U| \leq d$ and a positive integer k .

The task is to find a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that $\cup_{S \in \mathcal{F}'} S = U$. All four problems are known to be NP-complete (e.g., see [18] and the proof of Theorem 4.1); in this section, we show that the problems do not admit polynomial kernels for the parameter $(|T|, k)$ (in the case of STEINER TREE), k (in the case of CONVC and CAPVC), and (d, k) (in the case of SMALL UNIVERSE SET COVER), respectively. To this end, we first use the framework presented in Section 3 to prove that another problem, which is called RBDS, does not have a polynomial kernel. Then, by giving polynomial parameter transformations from RBDS to the above problems, we show the non-existence of polynomial kernels for these problems.

In RED-BLUE DOMINATING SET (RBDS) we are given a bipartite graph $G = (T \cup N, E)$ and an integer k and asked whether there exists a vertex set $N' \subseteq N$ of size at most k such that every vertex in T has at least one neighbor in N' . We show that RBDS parameterized by $(|T|, k)$ does not have a polynomial kernel. In the literature, the sets T and N are called “blue vertices” and “red vertices”, respectively. In this paper we will call the vertices “terminals” and “nonterminals” in order to avoid confusion with the colored version of the problem that we are going to introduce. RBDS is equivalent to SET COVER and HITTING SET and is, therefore, NP-complete [18].

In the colored version of RBDS, denoted by COLORED RED-BLUE DOMINATING SET (COL-RBDS), the vertices of N are colored with colors chosen from $\{1, \dots, k\}$, that is, we are additionally given a function $col : N \rightarrow \{1, \dots, k\}$, and N' is required to contain exactly one vertex of each color. We will now follow the framework described in Section 3.

Lemma 4.1. (1) *The unparameterized version of RBDS is in NP, and the unparameterized version of COL-RBDS is NP-hard.* (2) *There is a polynomial parameter transformation from COL-RBDS to RBDS.* (3) *COL-RBDS is solvable in $2^{|T|+k} \cdot |T \cup N|^{O(1)}$ time.*

Proof. (1) RBDS is known to be NP-complete. To prove the NP-hardness of COL-RBDS, we reduce RBDS to COL-RBDS: Given an instance $(G = (T \cup N, E), k)$ of RBDS, we construct an instance $(G' = (T \cup N', E'), k, col)$ of COL-RBDS where the vertex set N' consists of k copies v^1, \dots, v^k of every vertex $v \in V$, one copy of each color. That is, $N' = \cup_{a \in \{1, \dots, k\}} \{v^a \mid v \in N\}$, and the color of every vertex $v^a \in N_a$ is $col(v^a) = a$. The edge set E' is given by

$$E' = \bigcup_{a \in \{1, \dots, k\}} \{\{u, v^a\} \mid u \in T \wedge a \in \{1, \dots, k\} \wedge \{u, v\} \in E\}.$$

We omit the proof for the correctness of the construction, which is easy to see.

(2) Given an instance $(G = (T \cup N, E), k, col)$ of COL-RBDS, we construct an instance $(G' = (T' \cup N, E'), k)$ of RBDS. Thereby, the set T' consists of all vertices from T plus k additional vertices z_1, \dots, z_k . The edge set E' consists of all edges from E plus the edges

$$\{\{z_a, v\} \mid a \in \{1, \dots, k\} \wedge v \in N \wedge col(v) = a\}.$$

We omit the proof for the correctness of the construction.

(3) To solve COL-RBDS in the claimed running time, we first use the reduction given in (2) from COL-RBDS to RBDS. The number $|T'|$ of terminals in the constructed instance of RBDS is $|T| + k$. Next, we transform the RBDS instance (G', k) into an instance (\mathcal{F}, U, k) of SET COVER where the elements in U one-to-one correspond to the vertices in T' and the sets in \mathcal{F} one-to-one correspond to the vertices in N . Since SET COVER can be solved in $O(2^{|U|} \cdot |U| \cdot |\mathcal{F}|)$ time [16, Lemma 2], statement (3) follows. \square

Lemma 4.2. *COL-RBDS parameterized by $(|T|, k)$ is compositional.*

Proof. Given a sequence

$$(G_1 = (T_1 \cup N_1, E_1), k, col_1), \dots, (G_t = (T_t \cup N_t, E_t), k, col_t)$$

of instances of COL-RBDS with $|T_1| = |T_2| = \dots = |T_t| = p$, we show how to construct a COL-RBDS instance $(G = (T \cup N, E), k, col)$ as described in Definition 2.2.

For $i \in \{1, \dots, t\}$, let $T_i := \{u_1^i, \dots, u_p^i\}$ and $N_i := \{v_1^i, \dots, v_{q_i}^i\}$. We start with adding p vertices u_1, \dots, u_p to the set T of terminals to be constructed. (We will add more vertices to T later.) Next, we add to the set N of nonterminals all vertices from the vertex sets N_1, \dots, N_t , preserving the colors of the vertices. That is, we set $N = \bigcup_{i \in \{1, \dots, t\}} N_i$, and for every vertex $v_j^i \in N$ we define $col(v_j^i) = col_i(v_j^i)$. Now, we add the edge set $\bigcup_{i \in \{1, \dots, t\}} \left\{ \{u_{j_1}^i, v_{j_2}^i\} \mid \{u_{j_1}^i, v_{j_2}^i\} \in E_i \right\}$ to G (see Figure 1). The graph G and the coloring col constructed so far have the following property: If at least one of the COL-RBDS instances $(G_1, k, col_1), \dots, (G_t, k, col_t)$ is a *yes*-instance, then (G, k, col) is also a *yes*-instance because if for any $i \in \{1, \dots, t\}$ a size- k subset from N_i dominates all vertices in T_i , then the same vertex set selected from N also dominates all vertices in T . However, (G, k, col) may even be a *yes*-instance in the case where all instances $(G_1, k, col_1), \dots, (G_t, k, col_t)$ are *no*-instances, because in G one can select vertices into the solution that originate from different instances of the input sequence.

To ensure the correctness of the composition, we add more vertices and edges to G . We define for every graph G_i of the input sequence a unique identifier $ID(G_i)$, which consists of a $(p+k)$ -bit binary number. Since we can assume that the input sequence does not contain more than 2^{p+k} instances, $(p+k)$ bits are enough to assign unique identifiers to all instances of the input sequence. (Note that if there are more than 2^{p+k} instances, then we can solve all these instances in $\sum_{i=1}^t 2^{p+k} \cdot (p+q_i)^{O(1)} \leq t \cdot \sum_{i=1}^t (p+q_i)^{O(1)}$ time, which yields a composition algorithm.) For each color pair $(a, b) \in \{1, \dots, k\} \times \{1, \dots, k\}$ with $a \neq b$, we add a vertex set $W_{(a,b)} = \{w_1^{(a,b)}, \dots, w_{p+k}^{(a,b)}\}$ to T , and we add to E the edge set

$$\bigcup_{\substack{i \in \{1, \dots, t\}, \\ j_1 \in \{1, \dots, q_i\}}} \left\{ \{v_{j_1}^i, w_{j_2}^{(a,b)}\} \mid \begin{array}{l} a = col(v_{j_1}^i) \wedge \\ b \in \{1, \dots, k\} \setminus \{a\} \wedge \\ \text{the } j_2\text{th bit in } ID(G_i) \text{ is } 1 \end{array} \right\} \cup$$

$$\bigcup_{\substack{i \in \{1, \dots, t\}, \\ j_1 \in \{1, \dots, q_i\}}} \left\{ \{v_{j_1}^i, w_{j_2}^{(a,b)}\} \mid \begin{array}{l} b = col(v_{j_1}^i) \wedge \\ a \in \{1, \dots, k\} \setminus \{b\} \wedge \\ \text{the } j_2\text{th bit in } ID(G_i) \text{ is } 0 \end{array} \right\}$$

(see Figure 1).

Note that the construction conforms to the definition of a composition algorithm; in particular, k remains unchanged and the size of T is polynomial in p, k because $|T| = p+k(k-1) \cdot (p+k)$. To prove the correctness of the construction, we show that (G, k, col) has a solution $N' \subseteq N$ if and only if at least one instance (G_i, k, col_i) from the input sequence has a solution $N'_i \subseteq N_i$.

In one direction, if $N'_i \subseteq N_i$ is a solution for (G_i, k, col_i) , then the same vertex set chosen from N forms a solution for (G, k, col) . To see this, first note that the vertices u_1, \dots, u_p are dominated by the chosen vertices. Moreover, for every color pair $(a, b) \in \{1, \dots, k\} \times \{1, \dots, k\}$ with $a \neq b$, each vertex from $W_{(a,b)}$ is either connected to all vertices v from N_i with $col(v) = a$ or to all vertices v from N_i with $col(v) = b$. Since N'_i contains one vertex of each color class from N_i , each vertex in $W_{(a,b)}$ is dominated by a vertex from N chosen into the solution.

In the other direction, to show that any solution $N' \subseteq N$ for (G, k, col) is a solution for at least one instance (G_i, k, col_i) , we prove that N' cannot contain vertices originating from different instances of the input sequence. To this end, first note that each two vertices in N'

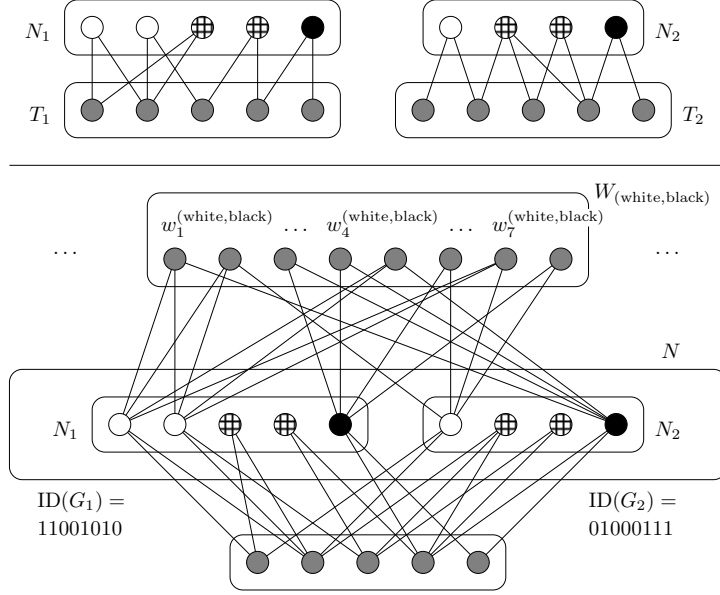


Figure 1: Example for the composition algorithm for COL-RBDS. The upper part of the figure shows an input sequence consisting of two instances with $k = 3$ (there are three colors: white, checkered, and black). The lower part of the figure shows the output of the composition algorithm. For the sake of clarity, only the vertex set $W_{(\text{white},\text{black})}$ is displayed, whereas five other vertex sets $W_{(a,b)}$ with $a, b \in \{\text{white}, \text{checkered}, \text{black}\}$ are omitted. Since $k = 3$ and $p = 5$, each ID consists of an eight-bit binary number, and $W_{(\text{white},\text{black})}$ contains eight vertices. The order of the vertices in $W_{(\text{white},\text{black})}$ from left to right corresponds to the order of the bits in the IDs from left to right, that is, the vertex $w_1^{(\text{white},\text{black})}$ corresponds to the leftmost bit in the IDs.

must have different colors. Assume, for the sake of a contradiction, that N' contains a vertex $v_{j_1}^{i_1}$ with $\text{col}(v_{j_1}^{i_1}) = a$ originating from the instance $(G_{i_1}, k, \text{col}_{i_1})$ and a vertex $v_{j_2}^{i_2}$ with $\text{col}(v_{j_2}^{i_2}) = b$ originating from a different instance $(G_{i_2}, k, \text{col}_{i_2})$. Since $\text{ID}(G_{i_1}) \neq \text{ID}(G_{i_2})$, there are vertices in $W_{(a,b)} \cup W_{(b,a)}$ (namely, all vertices $w_j^{(a,b)}$ such that the j th bit in $\text{ID}(G_{i_1})$ is 0 and the j th bit in $\text{ID}(G_{i_2})$ is 1, and all vertices $w_j^{(b,a)}$ such that the j th bit in $\text{ID}(G_{i_2})$ is 0 and the j th bit in $\text{ID}(G_{i_1})$ is 1) that are neither adjacent to $v_{j_1}^{i_1}$ nor to $v_{j_2}^{i_2}$. Therefore, N' does not dominate all vertices from T , which is a contradiction to the fact that N' is a solution for (G, k, col) . \square

Theorem 4.1. *The problems RED-BLUE DOMINATING SET and STEINER TREE, both parameterized by $(|T|, k)$, the problems CONNECTED VERTEX COVER and CAPACITATED VERTEX COVER, both parameterized by k , the problem SMALL UNIVERSE SET COVER parameterized by (k, d) , and the problem SET COVER parameterized by solution size k and the maximum size of any set in \mathcal{F} do not admit polynomial kernels unless $\text{PH} = \Sigma_p^3$.*

Proof. For RBDS the statement of the theorem follows directly by Theorem 2.1 together with Lemmata 4.1 and 4.2.

To show that the statement is true for the other four problems, we give polynomial parameter transformations from RBDS to each of them—due to Proposition 2.1, this suffices to prove the statement. Let $(G = (T \cup N, E), k)$ be an instance of RBDS. To transform it into an instance $(G' = (T' \cup N, E'), k)$ of STEINER TREE, define $T' = T \cup \{\tilde{u}\}$ where \tilde{u} is a new vertex and let $E' = E \cup \{\{\tilde{u}, v_i\} \mid v_i \in N\}$. It is easy to see that every solution for STEINER TREE on (G', k) one-to-one corresponds to a solution for RBDS on (G, k) .

To transform (G, k) into an instance $(G'' = (V'', E''), k'')$ of CONVC, first construct the graph $G' = (T' \cup N, E')$ as described above. The graph G'' is then obtained from G' by attaching a leaf to every vertex in T' . Now, G'' has a connected vertex cover of size $k'' = |T'| + k = |T| + 1 + k$ if and only if G' has a Steiner tree containing k vertices from N if and only if all vertices from T can be dominated in G by k vertices from N .

Next, we describe how to transform (G, k) into an instance $(G''' = (V''', E'''), \text{cap}, k''')$ of CAPVC. First, for each vertex $u_i \in T$, add a clique to G''' that contains four vertices $u_i^0, u_i^1, u_i^2, u_i^3$. Second, for each vertex $v_i \in N$, add a vertex v_i''' to G''' . Finally, for each edge $\{u_i, v_j\} \in E$ with $u_i \in T$ and $v_j \in N$, add the edge $\{u_i^0, v_j'''\}$ to G''' . The capacities of the vertices are defined as follows: For each vertex $u_i \in T$, the vertices $u_i^1, u_i^2, u_i^3 \in V'''$ have capacity 1 and the vertex $u_i^0 \in V'''$ has capacity $\deg_{G'''}(u_i^0) - 1$. Each vertex v_i''' has capacity $\deg_{G'''}(v_i''')$. Clearly, in order to cover the edges of the size-4 cliques inserted for the vertices of T , every capacitated vertex cover for G''' must contain all vertices $u_i^0, u_i^1, u_i^2, u_i^3$. Moreover, since the capacity of each vertex u_i^0 is too small to cover all edges incident to u_i^0 , at least one neighbor v_j''' of u_i^0 must be selected into every capacitated vertex cover for G''' . Therefore, it is not hard to see that G''' has a capacitated vertex cover of size $k''' = 4 \cdot |T| + k$ if and only if all vertices from T can be dominated in G by k vertices from N .

Finally, the results for SMALL UNIVERSE SET COVER and SET COVER follow from the equivalence of SET COVER and RBDS: To transform (G, k) into an instance (\mathcal{F}, U, k) of SMALL UNIVERSE SET COVER, add one element e_i to U for every vertex $u_i \in T$. For every vertex $v_j \in N$, add one set $\{e_i \mid \{u_i, v_j\} \in E\}$ to \mathcal{F} . The correctness of the construction is obvious; moreover, since $|U| = |T|$, every set in \mathcal{F} contains at most $|T|$ elements, which shows the claimed result for SET COVER. \square

4.2 Unique Coverage

In the UNIQUE COVERAGE problem we are given a universe U , a family of sets \mathcal{F} over U and an integer k . The problem is to find a subfamily \mathcal{F}' of \mathcal{F} and a set S of elements in U such that $|S| \geq k$ and every element of S appears in exactly one set in \mathcal{F}' , that is, the number of elements uniquely covered by \mathcal{F}' is at least k .

In order to obtain our negative results we have to utilize positive kernelization results for the problem. In some sense, we have to compress our instances as much as possible in order to show that what remains is incompressible even though it is big. We utilize the following well-known and simple reduction rules for the problem:

- (a) If any set $S \in \mathcal{F}$ contains at least k elements, then return yes.
- (b) If any element e is not contained in any set in \mathcal{F} , then remove e from U .
- (c) If none of the above rules can be applied and $|U| \geq k(k - 1)$, then return yes.

We show that the UNIQUE COVERAGE problem does not have a polynomial kernel unless $\text{PH} = \Sigma_p^3$. Notice that while the above reduction rules will compress the instance to an instance with at most $O(k^2)$ elements, this is not a polynomial kernel because there is no polynomial bound on the size of \mathcal{F} . Hence, our negative result implies that unless $\text{PH} = \Sigma_p^3$ the size of \mathcal{F} cannot be compressed to a polynomial in k in polynomial time. We start by defining the colorful reduced version COLORED REDUCED UNIQUE COVERAGE (COL-RED-UC) of the UNIQUE COVERAGE problem which is useful for making the composition algorithm. In this version the sets of \mathcal{F} are colored with colors from the set $\{1, \dots, k\}$ and \mathcal{F}' is required to contain exactly one set of each color. Furthermore, in COL-RED-UC every set S in \mathcal{F} has size at most $k - 1$ and $|U| \leq k^2$.

Lemma 4.3. (1) *The unparameterized version of UNIQUE COVERAGE is in NP, and the unparameterized version of COL-RED-UC is NP-hard.* (2) *There is a polynomial parameter transformation from COL-RED-UC to UNIQUE COVERAGE.* (3) *COL-RED-UC is solvable in time $O(k^{2k^2})$.*

Proof. (1) To show that COL-RED-UC is NP-hard we reduce from the UNIQUE COVERAGE problem. For an instance (\mathcal{F}, U, k) of UNIQUE COVERAGE we first apply the reduction rules above. We now assume that the given instance cannot be reduced any further. Now we make k copies of \mathcal{F} , one copy of each color. This new instance has a colored subfamily uniquely covering at least k elements if and only if (\mathcal{F}, U, k) is a yes instance to UNIQUE COVERAGE. Furthermore, even in the new instance we have that $|U| \leq k^2$ and that every set has at most $k - 1$ elements.

(2) We now prove that there is a polynomial parameter transformation from COL-RED-UC to UNIQUE COVERAGE. For an instance (\mathcal{F}, U, k) of COL-RED-UC we make a new instance (\mathcal{H}, U', k') to UNIQUE COVERAGE. Let $k' = k(k^2 + 1) + k$ and for every color i we add a set U_i of $k^2 + 1$ new elements to U and make all sets colored with i contain U_i in addition to what they already contain. Thus we have that $U' = U \cup \bigcup_{i \in \{1, \dots, k\}} U_i$. Notice that in order to cover at least $k(k^2 + 1)$ elements uniquely one has to pick exactly one set of each color. This concludes the polynomial parameter transformation.

(3) Finally, observe that COL-RED-UC can be solved in time $O(k^{2k^2})$ because the size of $|U|$ is bounded by k^2 . \square

Lemma 4.4. *The COL-RED-UC problem is compositional.*

Proof. Given a sequence of COL-RED-UC instances $\mathcal{I}_1 = (U, \mathcal{F}_1, k), \dots, \mathcal{I}_t = (U, \mathcal{F}_t, k)$, we construct a COL-RED-UC instance $\mathcal{I} = (U', \mathcal{F}, k')$. If the number of instances t is at least $2^{2k^2 \log k}$ then running the algorithm from Lemma 4.3 on all instances takes time polynomial in the input size yielding a trivial composition algorithm. Thus we assume that t is at most $2^{2k^2 \log k}$. We now construct ID's for every instance, this is done in two steps. In the first step every instance i gets a unique small id $ID'(\mathcal{I}_i)$ which is a subset of size $k^3/2$ of the set $\{1, \dots, k^3\}$. The identifier of instance i is the set $ID(\mathcal{I}_i)$ which is defined to be $ID(\mathcal{I}_i) = \{x \in \mathbb{N} : \lfloor x/k^3 \rfloor \in ID'(\mathcal{I}_i)\}$. In other words, $ID(\mathcal{I}_i) = \{k^3 \cdot j + j' \mid j \in ID'(\mathcal{I}_i) \wedge j' \in \{0, \dots, k^3 - 1\}\}$. Notice that the identifier of every instance is now a subset of size $k^6/2$ of the set $\{1, \dots, k^6\}$ and that the IDs of two different instances differ in at least k^3 places.

We start building the instance \mathcal{I} by letting $U' = U$ and $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \dots \cup \mathcal{F}_t$. The sets have the same color as in their respective instance. For every distinct ordered pair of colors $i, j \leq k$ we add the set $U_{i,j} = \{u_{i,j}^1, \dots, u_{i,j}^{k^6}\}$ to U' . For every instance \mathcal{I}_p we consider the sets colored i and j respectively in \mathcal{F}_p . To every set S with color i in \mathcal{F}_p we add the set $\{u_{i,j}^x : x \in ID(\mathcal{I}_p)\}$. Also, to every set S with color j in \mathcal{F}_p we add the set $\{u_{i,j}^x : x \notin ID(\mathcal{I}_p)\}$. Finally we set $k' = k(k-1)k^6 + k$. This concludes the construction.

If some \mathcal{I}_p has a colored subfamily \mathcal{F}' covering k elements uniquely, we show that the same subfamily covers k' elements uniquely in \mathcal{I} . First note that \mathcal{F}' covers k elements uniquely in U . It remains to prove that for every distinct ordered pair i, j of colors, all elements of $U_{i,j}$ are covered uniquely by \mathcal{F}' in \mathcal{I} . Consider an element $u_{i,j}^q \in U_{i,j}$ and let S_i and S_j be the sets colored i and j respectively in \mathcal{F}' . If $q \in ID(\mathcal{I}_p)$ then S_i contains $u_{i,j}^q$ and S_j does not. Similarly if $q \notin ID(\mathcal{I}_p)$ then S_j contains $u_{i,j}^q$ and S_i does not. Furthermore no other set of \mathcal{F}' contains $u_{i,j}^q$ and thus this element is uniquely covered.

In the other direction, suppose \mathcal{I} has a colored subfamily \mathcal{F}' covering k' elements uniquely. Suppose for contradiction that there is a color i and a color j such that the set $S_i \in \mathcal{F}'$ with color i and the set $S_j \in \mathcal{F}'$ with color j originate from different instances. Observe that in the set $U_{i,j}$ the sets S_i and S_j intersect in at least k^3 elements, and thus these elements are not covered uniquely by \mathcal{F}' . Then the total number of elements that can be uniquely covered by \mathcal{F}' is upper bounded by $k(k-1)k^6 + k^2 - k^3 < k(k-1)k^6 < k'$ yielding a contradiction. Thus all the sets in

\mathcal{F}' come from the same instance \mathcal{I}_p and uniquely cover at least k elements in \mathcal{I}_p . This concludes the proof. \square

Theorem 4.2. *The UNIQUE COVERAGE problem parameterized by k does not admit a polynomial kernel unless $PH = \Sigma_p^3$.*

4.3 Bounded Rank Disjoint Sets

In the BOUNDED RANK DISJOINT SETS problem we are given a family \mathcal{F} over a universe U with every set $S \in \mathcal{F}$ having size at most d together with a positive integer k . The question is whether there exists a subfamily \mathcal{F}' of \mathcal{F} with $|\mathcal{F}'| \geq k$ such that for every pair of sets $S_1, S_2 \in \mathcal{F}'$ we have that $S_1 \cap S_2 = \emptyset$. The problem can be solved in time $2^{O(dk)n^{O(1)}}$ using color-coding and an application of dk -perfect hash families. We outline a proof which shows that this problem does not admit a $\text{poly}(k, d)$ kernel. To do so we define a variation of the PERFECT CODE problem on graphs, which we call BIPARTITE REGULAR PERFECT CODE problem. In BIPARTITE REGULAR PERFECT CODE we are given a bipartite graph $G = (T \cup N, E)$, where every vertex in N has the same degree, and an integer k and asked whether there exists a vertex set $N' \subseteq N$ of size at most k such that every vertex in T has *exactly one neighbor* in N' . The set N' is called a bipartite perfect code. Now we are ready to state the main theorem of this subsection.

Theorem 4.3. *BIPARTITE REGULAR PERFECT CODE parameterized by $(|T|, k)$ and BOUNDED RANK DISJOINT SETS parameterized by (d, k) do not have a polynomial kernel unless $PH = \Sigma_p^3$.*

Proof. We can show that BIPARTITE REGULAR PERFECT CODE parameterized by $(|T|, k)$ does not have a polynomial kernel along the lines of the proof of Theorem 4.1, which shows that RBDS parameterized by $(|T|, k)$ does not have a polynomial kernel. BIPARTITE REGULAR PERFECT CODE is known to be NP-complete even when every vertex in N has degree exactly 3 [22]. The proof showing that the colored version of BIPARTITE REGULAR PERFECT CODE (1) is NP-complete, (2) has a fixed-parameter algorithm of the desired kind, and (3) has a polynomial parameter transformation to BIPARTITE REGULAR PERFECT CODE is just a minor modification as for RBDS. For the composition, it is enough to observe that if the input graphs to the composition algorithm are one sided regular then the composed graph will remain one sided regular in Lemma 4.2. This is true as every vertex is made adjacent to the same number of newly added vertices and newly added vertices are added to T .

Finally, to show that BOUNDED RANK DISJOINT SETS parameterized by (d, k) does not have a polynomial kernel we give a polynomial parameter transformation from BIPARTITE REGULAR PERFECT CODE to BOUNDED RANK DISJOINT SETS. To this end, given an instance $(G = (T \cup N, E), k)$ for BIPARTITE REGULAR PERFECT CODE, we make an instance (U, \mathcal{F}, k', d) for BOUNDED RANK DISJOINT SETS as follows. Let $U = T$, $\mathcal{F} = \{F_v \mid v \in N, F_v = N(v)\}$, $k' = k$ and $d = r$ where r is the degree of any vertex in N . Observe that $k = |T|/r$. From here it easily follows that G has a bipartite perfect code of size k if and only if (U, \mathcal{F}, k, d) has a subfamily \mathcal{F}' of \mathcal{F} of pairwise disjoint sets. \square

5 Domination and Transversals

In the SMALL UNIVERSE HITTING SET problem we are given a set family \mathcal{F} over a universe U with $|U| \leq d$ together with a positive integer k . The question is whether there exists a subset S in U of size at most k such that every set in \mathcal{F} has a non-empty intersection with S . We show that the SMALL UNIVERSE HITTING SET problem parameterized by the solution size k and the size $d = |U|$ of the universe does not have a kernel of size polynomial in (k, d) unless $PH = \Sigma_p^3$. One should notice that while HITTING SET and SET COVER in fact are the same problem, SMALL

UNIVERSE HITTING SET and SMALL UNIVERSE SET COVER are not, because in the former we are restricting the number of potential dominators while in the later we restrict the number of objects to be dominated.

We define the colored version of SMALL UNIVERSE HITTING SET, called COL-SUHS as follows. We are given a set family \mathcal{F} over a universe U with $|U| \leq d$, and a positive integer k . The elements of U are colored with colors from the set $\{1, \dots, k\}$ and the question is whether there exists a subset $S \subseteq U$ containing exactly one element of each color such that every set in \mathcal{F} has a non-empty intersection with S .

Lemma 5.1. (1) *The unparameterized version of SMALL UNIVERSE HITTING SET is in NP, and the unparameterized version of COL-SUHS is NP-hard.* (2) *There is a polynomial parameter transformation from COL-SUHS to SMALL UNIVERSE HITTING SET.* (3) *COL-SUHS parameterized by d, k is solvable in time $O(2^d \cdot n^{O(1)})$.*

Proof. (1) We show that COL-SUHS is NP-hard by reducing from SMALL UNIVERSE HITTING SET, which is easily seen to be NP-complete by a reduction from DOMINATING SET. Given an instance (\mathcal{F}, U, d, k) to SMALL UNIVERSE HITTING SET we make an instance $(\mathcal{F}', U', d \cdot k, k)$ of COL-SUHS by letting U' contain k copies of U , with one element of each color. For every set S in \mathcal{F} we make a set S' in \mathcal{F}' by letting S' contain all copies of all elements in S . From the construction it follows that (\mathcal{F}, U, d, k) has a hitting set of size at most k if and only if $(\mathcal{F}', U', d \cdot k, k)$ has a colored hitting set of size k .

(2) We now give a polynomial parameter transformation from COL-SUHS to SMALL UNIVERSE HITTING SET. For an instance (\mathcal{F}', U, d, k) to COL-SUHS we make an instance (\mathcal{F}, U, d, k) to SMALL UNIVERSE HITTING SET as follows. For every i between 1 and k let U_i be the set of elements in U colored with i . We construct \mathcal{F} by starting with \mathcal{F}' and for every i adding the set U_i to \mathcal{F}' . Now, a subset S of U hits every set of \mathcal{F} if and only if it hits every set of \mathcal{F}' and contains at least one vertex of each color.

(3) Finally, observe that COL-SUHS parameterized by d, k is solvable in time $O(2^d \cdot n^{O(1)})$ by enumerating all subsets of U and for each set checking whether it is a hitting set with at least one vertex of each color. \square

Lemma 5.2. *The problem COL-SUHS is compositional.*

Proof. We have to show how, given a sequence of COL-SUHS instances

$$(\mathcal{F}_1, U, d, k), (\mathcal{F}_2, U, d, k), \dots, (\mathcal{F}_t, U, d, k)$$

where $|U| \leq d$, to construct a COL-SUHS instance $(\mathcal{F}, U', d', k')$ as described in Definition 2.2.

If the number of instances is at least 2^d then running the algorithm from Lemma 5.1 on all instances takes time polynomial in the input size yielding a trivial composition algorithm. Thus we can assume that $t \leq 2^d$. Furthermore, we need the number of instances to be a power of 2. To make this true we add an appropriate number of no-instances. As we never add more than t extra instances in order to make the number of instances a power of 2 this can be done in polynomial time and hence we can safely assume that t is a power of 2, say 2^l . Observe that since $t \leq 2^d$ we have that $l \leq d$. Now, let every instance be identified by a unique number from 0 to $t - 1$.

We let $k' = k + l$ and start building $(\mathcal{F}, U', d', k')$ from $(\mathcal{F}_1, U, d, k), \dots, (\mathcal{F}_t, U, d, k)$ by letting $U' = U$ and letting elements keep their color. For every $i \leq t$ we add the family \mathcal{F}_i to \mathcal{F} . We now add $2l$ new elements $C = \{a_1, b_1, \dots, a_l, b_l\}$ to U' and for every $i \leq l$, $\{a_i, b_i\}$ comprise a new color class. We conclude the construction by modifying the sets in \mathcal{F} that came from the input instances to the composition algorithm. For every $j \leq t$ we consider all sets in \mathcal{F}_j . For every such set S we proceed as follows. Let $\text{ID}(j)$ be the identification number of instance number j .

For every $i \leq l$ we look at the i 'th bit in the binary representation of $\text{ID}(j)$. If this bit is set to 1 we add a_i to S and if the bit is set to 0 we add b_i to S . This concludes the construction.

Now, if there is a colored hitting set S for \mathcal{F}_j with $|S| \leq k$ we construct a colored hitting set S' for \mathcal{F} of size $k + l$ as follows. First we add S to S' and then we consider the identification number $\text{ID}(j)$ of instance j . For every i from 1 to l we consider the i 'th bit of $\text{ID}(j)$. If this bit is set to 1 we add b_j to S' else we add a_j to S' . Clearly S' is a hitting set for \mathcal{F}_i , has size $k + l$ and contains one vertex of each color. It remains to show that S' hits all other sets of \mathcal{F} . Consider any set $X \in \mathcal{F}_q$ for some $q \neq j$. Then there is an i such that $\text{ID}(q)$ differs from $\text{ID}(j)$ in the i 'th bit. If the i 'th bit of $\text{ID}(q)$ is 1 then the i 'th bit of $\text{ID}(j)$ is 0 and hence $a_i \in S'$. Since the i 'th bit of $\text{ID}(q)$ is 1, $a_i \in X$.

In the other direction, suppose there is a colored hitting set S' of size $l + k$ of \mathcal{F} . For every $i \leq l$, exactly one out of the vertices a_i and b_i is in S' . Let p be the number between 0 and $2^l - 1$ such that for every i the i 'th bit of p is 1 if and only if $b_i \in S'$. Observe that the sets in \mathcal{F} originating from the family \mathcal{F}_j such that $\text{ID}(j) = p$ do not contain any of the elements of $S' \cap C$. Thus $S'' = S' \cap U$ is a colored hitting set for \mathcal{F}_j containing at most one element from each color class. S'' can thus be extended to a colored hitting set S of \mathcal{F}_j with $|S| = k$, concluding the proof. \square

Theorem 5.1. *SMALL UNIVERSE HITTING SET parameterized by solution size k and universe size $|U| = d$ does not have a polynomial kernel unless $PH = \Sigma_p^3$. The DOMINATING SET problem parameterized by the solution size k and the size c of a minimum vertex cover of the input graph does not have a polynomial kernel.*

Proof. The first part of the theorem follows immediately from Lemmata 5.1 and 5.2. To show that the DOMINATING SET problem parameterized by the solution size k and the size c of a minimum vertex cover of the input graph does not have a polynomial kernel we give a polynomial parameter transformation from COL-SUHS to DOMINATING SET. On input (\mathcal{F}, U, d, k) to COL-SUHS we make an instance (G, k) to DOMINATING SET as follows. We start by letting G be the bipartite element-set incidence graph of (\mathcal{F}, U) . Now we add one vertex v_i for every color class i and make v_i adjacent to every vertex in G corresponding to an element of U with color i . This concludes the construction.

First, observe that U is a vertex cover of size d of G . Second, observe that if there is a colored hitting set S with $|S| \leq k$ of (\mathcal{F}, U, d, k) then S is a dominating set of G . Finally, if S is a dominating set of G then for every i there is a vertex in $S \cap N[v_i]$. Hence, $S \cap U$ is a hitting set for \mathcal{F} containing at most one element of each color. Thus this set can be extended to a colorful hitting set of \mathcal{F} , concluding the proof. \square

Theorem 5.1 has some interesting consequences. For instance, it implies that the HITTING SET problem parameterized by solution size k and the maximum size d of any set in \mathcal{F} does not have a kernel of size $\text{poly}(k, d)$ unless $PH = \Sigma_p^3$. The second part of Theorem 5.1 implies that the DOMINATING SET problem in graphs excluding a fixed graph H as a minor parameterized by $(k, |H|)$ does not have a kernel of size $\text{poly}(k, |H|)$ unless $PH = \Sigma_p^3$. This follows from the well-known fact that every graph with a vertex cover of size c excludes the complete graph K_{c+2} as a minor. Similarly, since every graph with a vertex cover of size c is c -degenerate it follows that the DOMINATING SET problem in d -degenerate graphs does not have a kernel of size $\text{poly}(k, d)$ unless $PH = \Sigma_p^3$.

Theorem 5.2. *Unless $PH = \Sigma_p^3$ the problems HITTING SET parameterized by solution size k and the maximum size d of any set in \mathcal{F} , DOMINATING SET IN H -MINOR FREE GRAPHS parameterized by $(k, |H|)$, and DOMINATING SET parameterized by solution size k and degeneracy d of the input graph do not have a polynomial kernel.*

6 Numeric Problem: Small Subset Sum

In the SUBSET SUM problem we are given a set S of n integers and a target t and asked whether there is a subset S' of S that adds up to exactly t . In the most common parameterization of this problem one is also given an integer k and asked whether there is a subset S' of S of size at most k that adds up to t . This parameterization, however, is $W[1]$ -hard. We consider a stronger parameterization where in addition to k an extra parameter d is provided and each of the integers in S is required to have size at most 2^d . This version, SMALL SUBSET SUM, is trivially fixed parameter tractable by dynamic programming. We believe that SMALL SUBSET SUM is the most restrictive plausible parameterization of the SUBSET SUM problem. We show that even this version does not admit a polynomial kernel, by giving a polynomial parameter transformation from the COLORED RED-BLUE DOMINATING SET (COL-RBDS) problem.

Theorem 6.1. *SMALL SUBSET SUM parameterized by (d, k) does not admit a kernel polynomial in (d, k) unless $PH = \Sigma_p^3$.*

Proof. We give a polynomial parameter transformation from the COLORED RED-BLUE DOMINATING SET (COL-RBDS) problem to SMALL SUBSET SUM. Given an instance $(G = (T \cup N, E), k, d)$ to COL-RBDS, such that $|T| = d$ and N has been colored with colors from $\{1, \dots, k\}$, we build an instance (S, t, k', d') to SMALL SUBSET SUM. For an integer $x \in S$ we treat x both as a number and as a string—the encoding of x in the number system with base $k(k+1)$.

We let t be a length- $(d+2k)$ string of digits representing the number $1 + k(k+1)/2$ and $k' = k(d+1)$. Now, order the elements of T in some order, say $T = t_1, t_2, \dots, t_d$. For a vertex $v \in N$ we define the string $Z(v)$ to be a string on d digits, where the i 'th digit is set to 1 if $\{v, t_i\} \in E$ and 0 otherwise. For an integer i between 1 and k we define the string $B(i)$ to be a length- k string with zeroes everywhere, except in the i 'th digit, which is $1 + k(k+1)/2$. For every vertex $x \in N$ we add a string to S : Let i be the color of x . We add the string $B(i)Z(x)B(k+1-i)$ to S and we will say that this string (or the number it represents) corresponds to x . Even though the numbers in the SMALL SUBSET SUM are uncolored, we color the string corresponding to x with the same color as x in order to ease the discussion. Finally we add a set of uncolored numbers to S : For every i between 1 and d and every j between 1 and k the string $u_{i,j}$ is a string of length $2k+d$ with zeroes everywhere except for the $k+i$ 'th digit, which is set to j . This concludes the construction. One should notice that every number is bounded by $(k(k+1))^{2k+d}$ and hence we can set $d' = 3(2k+d) \log k$.

Next, we prove that there is a set $N' \subseteq N$ containing one vertex of each color such that every vertex of T has a neighbor in N' if and only if there is a set $S' \subseteq S$ of at most k' numbers that add up to t .

Suppose that there is a set $N' \subseteq N$ containing one vertex of each color such that every vertex of T has a neighbor in N' . We pick $S' \subseteq S$ as follows. For every vertex $v \in N'$ we add the string corresponding to v to S' . Furthermore, for every $i \leq d$, let x_i be the number of neighbors the vertex $t_i \in T$ has in N' . We add the set $\{u_{i,j} : 1 \leq j \leq k \wedge j \neq x_i - 1\}$ to S' . Since we picked one number of each color, when we add up the numbers in S' there are no carries. Since every vertex $t_i \in T$ sees at least one and at most k vertices in N' , the $k+i$ 'th digit of the sum of all numbers in S' is exactly $1 + k(k+1)/2$.

In the other direction, suppose there is a set $S' \subseteq S$ with at most k' numbers that add up to t . If S contains no numbers colored 1, then the last digit of $\sum_{x \in S'} x$ must be zero, a contradiction. If S contains at least two numbers colored 1 then $\sum_{x \in S'} x > t$ again yielding a contradiction. Hence S' contains exactly one number colored 1. Let s_1 be the number in S' colored 1. Now, if S' contains no numbers colored 2 then the second last digit of $\sum_{x \in S'} x$ is zero and if S' contains at least 2 numbers colored 2 then $-s_1 + \sum_{x \in S'} x > t - s_1$, again contradicting that $t = \sum_{x \in S'} x$. Repeating the argument for the remaining colors yields that S' contains exactly one number of

each color. For every $i \leq k$ we let s_i be the number in S' colored i . For every i let v_i be the vertex in N corresponding to s_i . We prove that for every i the vertex $t_i \in T$ has a neighbor in N' . To achieve this we argue that there is a number s_j such that the $k + i$ 'th digit of s_j is 1. Suppose for contradiction that this is not the case. Notice that since there is at most one number of each color in S' there are no carries when we add up the numbers of S' , even if S' contains all uncolored numbers. Hence, even if S' contains all uncolored numbers whose $k + i$ 'th digit is non-zero the $k + i$ 'th digit of $\sum_{x \in S'} x$ must be strictly less than $1 + k(k + 1)/2$, yielding the desired contradiction, thereby completing the proof. \square

7 Kernelization and Compression Algorithms

Harnik and Naor [21] define *compression*, a notion with applications in cryptography and similar to kernelization in spirit. In this section we show that for a large class of problems, the notions of compressibility and polynomial kernels are in fact equivalent. This argument is implicit in [17] where it is observed that the definitions of compression of Harnik and Naor [21] and Fortnow and Santhanam are equivalent. We choose to include the argument for the sake of completeness. We follow the definitions given by Fortnow and Santhanam [17] in the realm of compression. A *parametric problem* is a subset of $\{x\#1^n \mid x \in \{0, 1\}^*, n \in \mathbb{N}\}$. The difference between parametric problems and parameterized problems is in the interpretation of the parameter. In a parametric problem the parameter is the size of the “witness string encoding the solution”, while in a parameterized problem it is typically the cardinality of a solution set. For an example let us consider the problem CLIQUE. A parametric problem corresponding to CLIQUE is $\{G\#1^{k \log n} \mid G \text{ has a clique of size } k\}$ (where n is the size of G), while the parameterized problem corresponding to CLIQUE is $\{(G, k) \mid G \text{ has a clique of size } k\}$. The unparameterized version of the latter is $\{G\#1^k \mid G \text{ has a clique of size } k\}$.

Definition 7.1. [Compression [17]] Let L be a parametric problem and $A \subseteq \{0, 1\}^*$. L is said to be compressible within A if there is a polynomial $p(\cdot)$, and a polynomial time computable function f such that for each $x \in \{0, 1\}^*$ and $n \in \mathbb{N}$, $|f(x\#1^n)| \leq p(n)$ and $x\#1^n \in L$ if and only if $f(x\#1^n) \in A$. L is compressible if there is some A for which L is compressible within A .

We study *subset problems* where the objective is to find in a given set a k -sized subset with specific properties such that if a subset is proposed, one can check in polynomial time whether this subset indeed is a solution to the problem considered. For such problems, the length of a witness string of an instance $(|I|, k)$ is $k \cdot \log |I|$. Thus, for a subset problem, the corresponding parametric problem is $L = \{I\#1^{k \cdot \log(|I|)} \mid I \text{ has a solution of size } k\}$, while the corresponding parameterized problem is $L^* = \{(I, k) \mid I \text{ has a solution of size } k\}$ and the unparameterized version of the latter is $\widetilde{L}^* = \{I\#1^k \mid I \text{ has a solution of size } k\}$.

Theorem 7.1. *Let L be the parametric problem corresponding to a subset problem, and let L^* be the corresponding parameterized problem. If the unparameterized version of L^* is NP-complete and there is an algorithm that decides whether an instance (I, k) belongs to L^* in time $2^{k^c} \cdot |I|^{O(1)}$ for a fixed constant c , then L has a compression to a language A in NP if and only if L^* admits a polynomial kernel.*

Proof. We first show that if L^* has a polynomial kernel then L has a compression to a language A in NP. Given an instance $I\#1^t$, we run the kernelization algorithm for L^* on the instance (I, k) with $k = t / \log |I|$. This returns an instance of I' of L^* such that $|I'| \leq k^c$ for some fixed constant c . Hence, $|I'| \leq k^c \leq t^c$ yielding a compression for L in L itself, i.e. self-compression.

In the other direction, given an instance (I, k) we check whether $|I| > 2^{k^c}$. If $|I| > 2^{k^c}$ we solve the problem in time $2^{k^c} |I|^{O(1)} \leq |I|^{O(1)}$, which is polynomial in $|I|$. If, on the other

hand, $|I| \leq 2^{k^c}$, then we apply compression on $I \# 1^{k \cdot \log |I|}$ resulting in an instance I' to the NP problem A such that $|I'| \leq \text{poly}(k, \log |I|)$ which is polynomial in k since $\log |I| \leq k^c$. Now since A is in NP and the unparameterized version \widetilde{L}^* of L^* is NP-complete, there is a polynomial-time many-one reduction from A to \widetilde{L}^* . This yields an instance $I' \# 1^{k^*}$ to \widetilde{L}^* such that $|I'|$ and k^* are bounded by a polynomial in k and $(I, k) \in L^*$ if and only if $(I', k^*) \in L^*$. \square

Since many of the problems considered in this paper are subset problems, combining Theorem 7.1 with our kernelization lower bounds yields the following corollary.

Corollary 7.1. *CONNECTED VERTEX COVER, CAPACITATED VERTEX COVER, STEINER TREE, UNIQUE COVERAGE, and SMALL SUBSET SUM do not have a compression to any language in NP.*

8 Conclusion, Discussions and Further Work

We have shown that several important parameterized problems do not admit polynomial kernels unless the polynomial hierarchy collapses to the third level. In order to do this we gave a cookbook for how to show composition algorithms, applied the approach to a few basic problems, and used these problems as starting points for polynomial parameter transformations. We believe that both the cookbook for showing composition algorithms and the problems we have shown not to admit polynomial kernels will be useful tools for showing kernelization lower bounds for many other problems as well.

The approach of bounding the number of instances in the input by 2^{k^c} (steps 5 and 6a in our “cookbook”) was already applied by Bodlaender et al. [7] for obtaining a composition algorithm for a string problem called DISJOINT FACTORS; implicitly, Bodlaender et al. also assigned IDs to the problem instances: Their composition algorithm constructs one single output string by concatenating all input strings—the position of each input string in the output string can be interpreted as the ID of the string. However, we use the IDs in a more systematic way.

Results in this paper have already been used to obtain kernelization lower bound in a few papers. These include recent results by Kratsch and Wahlström [23], Cygan et al. [10] and Misra et al. [24] who used UNIQUE COVERAGE, STEINER TREE and CONNECTED VERTEX COVER and CONNECTED VERTEX COVER respectively as a starting point to obtain polynomial parameter transformations. In another development Dell and van Melkebeek [11] have obtained a strengthening of a result in [17] and using that they are able to show concrete lower bounds on problems that do admit polynomial kernels. In particular they have shown that HITTING SET does not admit a kernel of size $O(k^{d-1})$ when parameterized by the solution size k and maximum set size d .

References

- [1] F. N. Abu-Khzam. Kernelization algorithms for d -hitting set problems. In *Proc. 10th WADS*, volume 4618 of *LNCS*, pages 434–445. Springer, 2007.
- [2] J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004.
- [3] N. Alon and S. Gutner. Kernels for the dominating set problem on graphs with an excluded minor. Technical Report TR08-066, Electronic Colloquium on Computational Complexity (ECCC), 2008.
- [4] N. Betzler. Steiner tree problems in the analysis of biological networks. Diploma thesis, Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany, 2006.

- [5] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. 39th STOC*, pages 67–74. ACM Press, 2007.
- [6] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
- [7] H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *ESA*, volume 5757 of *LNCS*, pages 635–646, 2009.
- [8] J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM J. Comput.*, 37(4):1077–1106, 2007.
- [9] J. Chen, I. A. Kanj, and W. Jia. Vertex Cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.
- [10] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. Wojtaszczyk. Kernelization hardness of connectivity problems in d -degenerate graphs. In *WG*, volume 6410 of *LNCS*, pages 147–158, 2010.
- [11] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260, 2010.
- [12] M. Dom, D. Lokshtanov, S. Saurabh, and Y. Villanger. Capacitated domination and covering: A parameterized perspective. In *Proc. 3rd IWPEC*, volume 5018 of *LNCS*, pages 78–90. Springer, 2008.
- [13] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
- [14] H. Fernau, F. V. Fomin, D. Lokshtanov, D. Raible, S. Saurabh, and Y. Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. In *Proc. 26th STACS*, volume 3 of *LIPICs*, pages 421–432, 2009.
- [15] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [16] F. V. Fomin, D. Kratsch, and G. J. Woeginger. Exact (exponential) algorithms for the dominating set problem. In *Proc. 30th WG*, volume 3353 of *LNCS*, pages 245–256. Springer, 2004.
- [17] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proc. 40th STOC*, pages 133–142. ACM Press, 2008.
- [18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
- [19] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
- [20] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of Vertex Cover variants. *Theory Comput. Syst.*, 41(3):501–520, 2007.
- [21] D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. In *Proc. 47th FOCS*, pages 719–728. IEEE, 2007.
- [22] J. Kratochvíl and M. Krivánek. On the computational complexity of codes in graphs. In *Proc. 13th MFCS*, volume 324 of *LNCS*, pages 396–404. Springer, 1988.

- [23] S. Kratsch and M. Wahlström. Preprocessing of min ones problems: A dichotomy. In *ICALP*, volume 6198 of *LNCS*, pages 653–665, 2010.
- [24] N. Misra, G. Philip, V. Raman, S. Saurabh, and S. Sikdar. FPT algorithms for connected feedback vertex set. In *WALCOM*, volume 5942 of *LNCS*, pages 269–280, 2010.
- [25] D. Mölle, S. Richter, and P. Rossmanith. Enumerate and expand: Improved algorithms for connected vertex cover and tree cover. *Theory Comput. Syst.*, 43(2):234–253, 2008.
- [26] H. Moser, V. Raman, and S. Sikdar. The parameterized complexity of the unique coverage problem. In *Proc. 18th ISAAC*, volume 4835 of *LNCS*, pages 621–631. Springer, 2007.
- [27] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- [28] S. Thomassé. A quadratic kernel for feedback vertex set. In *Proc. 20th SODA*, pages 115–119. ACM/SIAM, 2009.

Problem Compendium

BIPARTITE REGULAR PERFECT CODE

- Input:** A bipartite graph $G = (T \cup N, E)$ with every vertex in N having the same degree and a positive integer k .
- Question:** Is there a vertex subset $N' \subseteq N$ of size at most k such that every vertex in T has exactly one neighbor in N' ?
- Parameter:** $|T|, k$.
- Algorithm:** $O(2^{(k \cdot |T|)} \cdot (|T| + |N|)^{O(1)})$.
- Best Kernel:** $O(2^{|T|})$; no $O((|T| + k)^{O(1)})$ kernel.
- Colored Version:** N is colored with colors from $\{1, \dots, k\}$ and N' is required to contain one vertex of each color.

BOUNDED RANK DISJOINT SETS

- Input:** A set family \mathcal{F} over a universe U with every set $S \in \mathcal{F}$ having size at most d , and a positive integer k .
- Question:** Is there a subfamily \mathcal{F}' of \mathcal{F} of size at most k such that every pair of sets S_1, S_2 in \mathcal{F}' we have $S_1 \cap S_2 = \emptyset$?
- Parameter:** k, d .
- Algorithm:** $2^{O(kd)} (|\mathcal{F}| + |U|)^{O(1)}$ [13].
- Best Kernel:** $2^{O(kd)}$ [13]; no $O((k + d)^{O(1)})$ kernel.

CAPACITATED VERTEX COVER (CAPVC)

- Input:** A graph $G = (V, E)$ and a capacity function $cap : V \rightarrow \mathbb{N}^+$ and a positive integer k .
- Question:** Is there a vertex subset $C \subseteq V$ of size at most k and a function $f : E \rightarrow C$ that maps every edge to one of its endpoints and so that for all $v \in C$, $|f^{-1}(v)| \leq cap(v)$?
- Parameter:** k .
- Algorithm:** $O(2^{k \log k} \cdot |V|^{O(1)})$ [12].
- Best Kernel:** $O(4^k \cdot k^2)$ [20]; no $O(k^{O(1)})$ kernel.

CONNECTED VERTEX COVER (CONVC)

- Input:** A graph $G = (V, E)$ and a positive integer k .
- Question:** Is there a vertex subset $C \subseteq V$ of size at most k such that $G[C]$ is connected and every edge of G has at least one endpoint in C ?
- Parameter:** k .
- Algorithm:** $O(2.7606^k \cdot |V|^{O(1)})$ [25].
- Best Kernel:** $O(2^k + k + 2k^2)$; no $O(k^{O(1)})$ kernel.

DOMINATING SET IN GRAPHS WITH SMALL VERTEX COVER

Input: A graph $G = (V, E)$ with a vertex cover of size t and an integer k .
Question: Is there a vertex set $V' \subseteq V$ of size at most k such that every vertex in $V \setminus V'$ has a neighbor in V' ?
Parameter: k, t .
Algorithm: $2^t \cdot |V|^{O(1)}$.
Best Kernel: $O(2^t)$; no $O((k + t)^{O(1)})$ kernel.
Remark: Implies that DOMINATING SET IN H -MINOR FREE GRAPHS parameterized by solution size k and $|H|$ does not have a $O((k + |H|)^{O(1)})$ kernel, and that DOMINATING SET parameterized by k and degeneracy d of the input graph has no $O((k + d)^{O(1)})$ kernel.

RED-BLUE DOMINATING SET (RBDS)

Input: A graph $G = (T \cup N, E)$ where both T and N are independent sets in G , and an integer k .
Question: Is there a vertex subset $N' \subseteq N$ of size at most k such that each vertex in T has at least one neighbor in N' ?
Parameter: $|T|, k$.
Algorithm: $O(2^{|T|} \cdot |T \cup N|^{O(1)})$ [16, Lemma 2].
Best Kernel: $O(2^{|T|} + |T|)$; no $O(k^{O(1)})$ kernel.
Colored Version: N is colored with colors from $\{1, \dots, k\}$ and N' is required to contain one vertex of each color.
Remark: Equivalent to SMALL UNIVERSE SET COVER.

SMALL SUBSET SUM

Input: An integer k , a set S of integers of size at most 2^k and an integer t .
Question: Is there a subset $S' \subseteq S$ with $|S'| \leq k$ such that $\sum_{y \in S'} y = t$?
Parameter: k .
Algorithm: $O(2^k \cdot |S|^{O(1)})$.
Best Kernel: $O(2^k)$; no $O(k^{O(1)})$ kernel.

SMALL UNIVERSE HITTING SET

Input: A set family \mathcal{F} over a universe U with $|U| = d$, and a positive integer k .
Question: Is there a subset $H \subseteq U$ of size at most k such that for every set $S \in \mathcal{F}$, $H \cap S \neq \emptyset$?
Parameter: k, d .
Algorithm: $O(2^d \cdot (|\mathcal{F}| + |U|)^{O(1)})$.
Best Kernel: $O(2^d)$; no $O((k + d)^{O(1)})$ kernel.
Colored Version: U is colored with colors from $\{1, \dots, k\}$ and H is required to contain one vertex of each color.
Remark: Implies that HITTING SET parameterized by solution size k and maximum set size d does not have a $O((k + d)^{O(1)})$ kernel.

SMALL UNIVERSE SET COVER

Input: A set family \mathcal{F} over a universe U with $|U| = d$ and a positive integer k .
Question: Is there a subfamily \mathcal{F}' of \mathcal{F} of size at most k such that $\cup_{S \in \mathcal{F}'} S = U$?
Parameter: k, d .
Algorithm: $O(2^{|T|} \cdot (|T \cup N|)^{O(1)})$ [16, Lemma 2].
Best Kernel: $O(2^{|T|} + |T|)$; no $O(k^{O(1)})$ kernel.
Remark: Equivalent to RED-BLUE DOMINATING SET. Implies that SET COVER parameterized by solution size k and maximum set size d does not have a $O((k + d)^{O(1)})$ kernel.

STEINER TREE

Input: A graph $G = (T \cup N, E)$ and an integer k .
Question: Is there a vertex subset $N' \subseteq N$ of size at most k such that $G[T \cup N']$ is connected?
Parameter: $|T|, k$.
Algorithm: $O(2^{|T|} \cdot |T \cup N|^{O(1)})$ [5].
Best Kernel: $O(2^{|T|} + |T|)$; no $O(k^{O(1)})$ kernel.

UNIQUE COVERAGE

Input: A set family \mathcal{F} over a universe U and a positive integer k .
Question: Is there a subfamily \mathcal{F}' of \mathcal{F} such that at least k elements of U are contained in exactly one set in \mathcal{F}' ?
Parameter: k .
Algorithm: $O(4^{k^2} (|\mathcal{F}| + |U|)^{O(1)})$ [26].
Best Kernel: 4^k [26]; no $O(k^{O(1)})$ kernel.
Colored Version: Each set in \mathcal{F} is colored with a colors from $\{1, \dots, k\}$ and \mathcal{F}' is required to contain exactly one set of each color.