Originally published in *Proc. 36th ICALP*, volume 5555 of *LNCS*. Springer. To appear.

Incompressibility through Colors and IDs

Michael Dom[1], Daniel Lokshtanov[2], and Saket Saurabh[2]

[1] Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany. michael.dom@uni-jena.de

[2] Department of Informatics, University of Bergen, 5020 Bergen, Norway. {daniello|saket.saurabh}@ii.uib.no

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 has been developed by Bodlaender et al. [4] and Fortnow and Santhanam [9]. 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:

—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 size $2k$.

—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 [2]. 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 [13]. We show that their algorithm is essentially optimal since the instances cannot be compressed further.

—Hitting Set and Set Cover 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.

All results are under the assumption that the polynomial hierarchy does not collapse to the third level. The existence of polynomial kernels for several of the problems mentioned above were open problems explicitly stated in the literature [2, 3, 11, 12, 14]. Many of our results also rule out the existence of compression algorithms, a notion similar to kernelization defined by Harnik and Naor [13], 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 the quality of preprocessing rules. Parameterized complexity provides a natural mathematical framework to give performance guarantees of preprocessing rules: A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm, called a *kernelization*, that reduces the input instance to an instance with size bounded by a polynomial $p(k)$ in the parameter k, while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. (See [6, 8, 15] for further introductions.) 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. [4] and Fortnow and Santhanam [9] have developed a framework for showing that a problem does not admit a polynomial kernel unless the polynomial hierarchy collapses to the third level ($PH = \Sigma_p^3$), which is deemed unlikely.

Bodlaender et al. [4] 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. [7] showing that the Directed Max Leaf Out-Branching problem does not have a polynomial kernel, and a result by Bodlaender et al. [5] 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. [5]. In order to show our results, we apply both methods. First, we present in Section 3 a "cookbook" approach for showing kernelization lower bounds by using composition algorithms together with polynomial parameter transformations. In the subsequent sections, 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, showing that a variety of basic problems do not have a polynomial kernel. All our results summarized below 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: In Section 4, 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 [3]. 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 [11, 12], and the negative answer is surprising because the closely related Vertex Cover problem admits a kernel of size $2k$. Finally, Bounded Rank Disjoint Sets and Unique Coverage do not admit a polynomial kernel. The latter result resolves an open problem of Moser et al. [14].
Domination and Transversals: In Section 5, 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 [2], who obtain a $k^{poly(h)}$ kernel for Dominating Set in H-Minor Free Graphs and ask whether kernels of smaller size exist. Another implication of the results in Sections 4 and 5 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].
Numeric Problems: Harnik and Naor obtain a *compression algorithm* for the Sparse Subset Sum problem [13]. 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. In Section 6, we show in contrast that if the input instance is a huge set of relatively small numbers, the instance cannot be reduced.

Harnik and Naor [13] define *compression*, a notion with applications in cryptography and similar to kernelization in spirit. It is implicit from the discussion in [9] that for a large class of problems the notions of kernelization and compression are equivalent. Due to this, our kernelization lower bounds imply that several of the problems we considered do not admit compression to a language in NP. These problems are Connected Vertex Cover, Capacitated Vertex Cover, Steiner Tree, Unique Coverage, and Small Subset Sum.

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 solvability in time $f(k) \cdot p(|x|)$ for any instance (x, k), where f is an arbitrary function of k and p is a polynomial.

Definition 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. [Composition [4]] *A composition algorithm for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, k), \ldots, (x_t, k))$, 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 $(y, k') \in \Sigma^* \times \mathbb{N}^+$ with (a) $(y, k') \in L \iff (x_i, k) \in L$ 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.*

We utilize a recent result of Bodlaender et al. [4] and Fortnow and Santhanam [9] concerning the non-existence of polynomial kernels. To this end, 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 1 ([4, 9]). *Let L be a compositional parameterized problem whose unparameterized version $\widetilde{L}$ is* NP*-complete. Then, unless* PH=Σ_p^3*, there is no polynomial kernel for L.*

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

Definition 3 ([5]). *Let P and Q be parameterized problems. We say that P is polynomial parameter reducible to Q, written $P \leq_{Ptp} Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ 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 1 ([5]). *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.*

A notion similar to polynomial parameter transformation was independently used by Fernau et al. [7] albeit without being explicitly defined.

We close with some definitions from graph theory. For a vertex v in a graph 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. The subgraph of G induced by a vertex set V' is denoted with $G[V']$. A vertex v *dominates* a vertex u if $u \in N_G(v)$.

3 A Systematic Approach to Prove Kernelization Lower Bounds

In this section we describe a "cookbook" for showing kernelization and compressibility lower bounds. To show that a problem does not admit a polynomial size kernel we go through the following steps.
1. Define a suitable colored version of the problem. This is in order to get more control over how solutions to problem instances can look.

2. 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.
3. Give a polynomial parameter transformation from the colored to the uncolored version. This will imply that kernelization lower bounds for the colored version directly transfer to the original problem.
4. Show that the colored version parameterized by k is solvable in time $2^{k^c} \cdot n^{O(1)}$ for a fixed constant c.
5. 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 [5].
(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 $poly(k)$ sized set.
(c) Use the coding power provided by colors and IDs to complete the composition algorithm.

4 Connectivity and Covering Problems

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 $cap : V \to \mathbb{N}^+$ 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 $cap(v)$ 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 [10] and the proof of Theorem 2); 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, therefore, NP-complete [10]. 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 $\{1, \ldots, k\}$, that is, we are additionally given a function $col : N \to \{1, \ldots, k\}$, and N' is required to contain exactly one vertex of each color. We will now follow the framework from Section 3.

Lemma 1. [⋆][3] (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.*

Lemma 2. COL-RBDS *parameterized by* $(|T|, k)$ *is compositional.*

Proof. For a sequence $(G_1 = (T_1 \cup N_1, E_1), k, col_1), \ldots, (G_t = (T_t \cup N_t, E_t), k, col_t)$ of COL-RBDS instances with $|T_1| = |T_2| = \ldots = |T_t| = p$, we show how to construct a COL-RBDS instance $(G = (T \cup N, E), k, col)$ as described in Definition 2.

For $i \in \{1, \ldots, t\}$, let $T_i := \{u_1^i, \ldots, u_p^i\}$ and $N_i := \{v_1^i, \ldots, v_{q_i}^i\}$. We start with adding p vertices $u_1, \ldots, u_p$ 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 $N_1, \ldots, N_t$, preserving the colors of the vertices. That is, we set $N = \bigcup_{i \in \{1,\ldots,t\}} N_i$, and $col(v_j^i) = col_i(v_j^i)$. Now, we add the edge set $\bigcup_{i \in \{1,\ldots,t\}} \{\{u_{j_1}, v_{j_2}^i\} \mid \{u_{j_1}^i, v_{j_2}^i\} \in E_i\}$ to G (see Figure 1). The graph G and the coloring col constructed so far have the following property: If at least one of $(G_1, k, col_1), \ldots, (G_t, k, col_t)$ is a *yes*-instance, then (G, k, col) is also a *yes*-instance. However, (G, k, col) may even be a *yes*-instance in the case where all instances $(G_1, k, col_1), \ldots, (G_t, k, col_t)$ 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 $\mathrm{ID}(G_i)$, which consists of a size-$(p + k)$ subset of $\{1, \ldots, 2(p + k)\}$. Since $\binom{2(p+k)}{p+k} \geq 2^{p+k}$ and since we can assume that the input sequence does not contain more than 2^{p+k} instances, it is always possible to assign unique identifiers to all instances of the input sequence. For each color pair $(a, b) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ with $a \neq b$, we add a vertex set $W_{(a,b)} = \{w_1^{(a,b)}, \ldots, w_{2(p+k)}^{(a,b)}\}$ to T,

[3] Proofs of results labelled with [⋆] have been omitted, whole or in part.

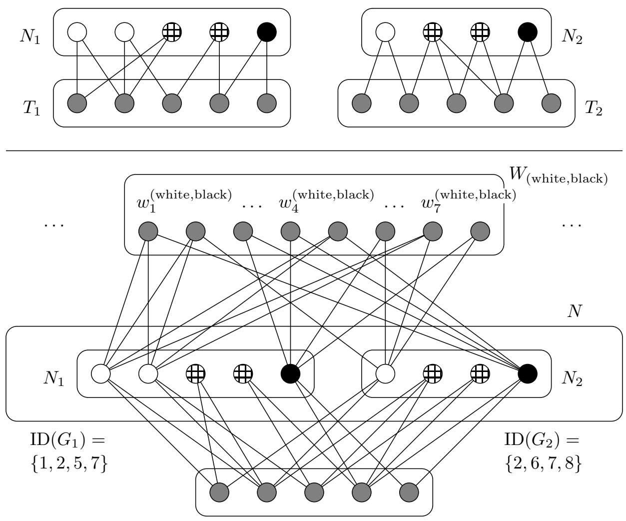

Fig. 1. Example for the composition algorithm for Col-RBDS. The upper part of the figure shows an input sequence consisting of two instances with $k = 3$ (there are three colors: white, checkered, and black). The lower part of the figure shows the output of the composition algorithm. For the sake of clarity, only the vertex set $W_{(\text{white,black})}$ is displayed, whereas five other vertex sets $W_{(a,b)}$ with $a, b \in \{\text{white, checkered, black}\}$ are omitted. Since $k = 3$ and $p = 5$, each ID should consist of eight numbers, and $W_{(\text{white,black})}$ should contain 16 vertices. For the sake of clarity, the displayed IDs consist of only four numbers each, and $W_{(\text{white,black})}$ contains only eight vertices.

(see Figure 1), and we add to E the edge set

$$\bigcup_{i\in\{1,\ldots,t\},j_1\in\{1,\ldots,q_i\}} \Big\{\{v^i_{j_1}, w^{(a,b)}_{j_2}\} \mid a = col(v^i_{j_1}) \wedge b \neq a \wedge j_2 \in \mathrm{ID}(G_i)\Big\} \cup$$
$$\bigcup_{i\in\{1,\ldots,t\},j_1\in\{1,\ldots,q_i\}} \Big\{\{v^i_{j_1}, w^{(a,b)}_{j_2}\} \mid b = col(v^i_{j_1}) \wedge a \neq b \wedge j_2 \notin \mathrm{ID}(G_i)\Big\} .$$

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 $|T| = p + k(k-1) \cdot 2(p+k)$. 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, note that for every color pair $(a, b) \in \{1, \ldots, k\} \times \{1, \ldots, k\}$ 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 $col(v) = b$.

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

Originally published in *Proc. 36th ICALP*, volume 5555 of *LNCS*. Springer. To appear.

vertices originating from different instances of the input sequence. To this end, note that each two vertices in N' must have different colors: Assume, for the sake of a contradiction, that N' contains a vertex $v_{j_1}^{i_1}$ with $col(v_{j_1}^{i_1}) = a$ originating from the instance (G_{i_1}, k, col_{i_1}) and a vertex $v_{j_2}^{i_2}$ with $col(v_{j_2}^{i_2}) = b$ originating from a different instance (G_{i_2}, k, col_{i_2}). Due to the construction of the IDs, we have $\mathrm{ID}(G_{i_1}) \setminus \mathrm{ID}(G_{i_2}) \neq \emptyset$ and $\mathrm{ID}(G_{i_2}) \setminus \mathrm{ID}(G_{i_1}) \neq \emptyset$. No vertex $w_j^{(a,b)}$ with $j \in \mathrm{ID}(G_{i_2}) \setminus \mathrm{ID}(G_{i_1})$ and no vertex $w_j^{(b,a)}$ with $j \in \mathrm{ID}(G_{i_1}) \setminus \mathrm{ID}(G_{i_2})$ is adjacent to one of $v_{j_1}^{i_1}$ and $v_{j_2}^{i_2}$. Therefore, N' does not dominate all vertices from T—a contradiction. □

Theorem 2. *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* $PH = \Sigma_p^3$*.*

Proof. For RBDS the statement of the theorem follows directly by Theorem 1 together with Lemmata 1 and 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 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 $k'' = |T'| + k = |T| + 1 + k$ iff G' has a Steiner tree containing k vertices from N iff 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 $(G''' = (V''', E'''), cap, k''')$ 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 $k''' = 4 \cdot |T| + k$ iff all vertices from T can be dominated in G by k vertices from N.

The results for Small Universe Set Cover and Set Cover follow from the equivalence of Set Cover and RBDS. □

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

We show that the Unique Coverage problem does not have a polynomial kernel unless PH=Σ_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 $\mathcal{F}$. We start by defining the colorful reduced version Colored Reduced Unique Coverage (Col-Red-UC) of the Unique Coverage problem. In this version the sets of $\mathcal{F}$ are colored with colors from $\{1,\ldots,k\}$ 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 3. [$\star$] (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 *parameterized by k is solvable in time $O(k^{2k^2})$.*

Lemma 4. [$\star$] *The* Col-Red-UC *problem is compositional.*

Proof. Given a sequence of Col-Red-UC instances $\mathcal{I}_1 = (U, \mathcal{F}_1, k), \ldots, \mathcal{I}_t = (U, \mathcal{F}_t, k)$, 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 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 for every instance, this is done in two steps. In the first step every instance i gets a unique small id $\mathrm{ID}'(\mathcal{I}_i)$ which is a subset of size $k^3/2$ of the set $\{1,\ldots,k^3\}$. The identifier of instance i is the set $\mathrm{ID}(\mathcal{I}_i)$ which is defined to be $\mathrm{ID}(\mathcal{I}_i) = \{x \in \mathbb{N} : \lfloor x/k^3 \rfloor \in \mathrm{ID}'(\mathcal{I}_i)\}$. In other words, $\mathrm{ID}(\mathcal{I}_i) = \{k^3 \cdot j + j' \mid j \in \mathrm{ID}'(\mathcal{I}_i) \wedge j' \in \{0,\ldots,k^3-1\}\}$. Notice that the identifier of every instance is now a subset of size $k^6/2$ of the set $\{1,\ldots,k^6\}$ 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 $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \ldots \cup \mathcal{F}_t$. 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 $U_{i,j} = \{u_{i,j}^1, \ldots, u_{i,j}^{k^6}\}$ 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 $\{u^x_{i,j} : x \in \mathrm{ID}(\mathcal{I}_p)\}$. Also, to every set S with color j in $\mathcal{F}_p$ we add the set $\{u^x_{i,j} : x \notin \mathrm{ID}(\mathcal{I}_p)\}$. Finally we set $k' = k(k-1)k^6 + k$. This concludes the construction. The correctness proof is omitted. □

Theorem 3. *The* Unique Coverage *problem parameterized by k does not admit a polynomial kernel unless $PH = \Sigma^3_p$.*

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. To show that this problem does not admit a poly(k,d) kernel, we define a variation of the Perfect Code problem on graphs: 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'.

Theorem 4. [$\star$] Bipartite Regular Perfect Code *parameterized by* $(|T|, k)$ *and* Bounded Rank Disjoint Sets *parameterized by* (d, k) *do not have a polynomial kernel unless $PH = \Sigma^3_p$.*

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 $PH = \Sigma^3_p$. 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 $\{1, \ldots, k\}$ 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. [$\star$] (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 $O(2^d \cdot n^{O(1)})$.*

Lemma 6. *The problem* Col-SUHS *is compositional.*

Proof. Given a sequence $(\mathcal{F}_1, U, d, k), \ldots, (\mathcal{F}_t, U, d, k)$ of Col-SUHS instances where $|U| \leq d$, we construct an instance $(\mathcal{F}, U', d', k')$ of Col-SUHS as described in Definition 2. Due to the time-$O(2^d \cdot n^{O(1)})$ algorithm from Lemma 5, 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, such that the total number of instances is 2^l. 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 $(\mathcal{F}_1, U, d, k), \ldots, (\mathcal{F}_t, U, d, k)$ 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 $C = \{a_1, b_1, \ldots, a_l, b_l\}$ 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 $\mathrm{ID}(j)$ 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 $\mathrm{ID}(j)$. 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 $|S| \leq k$, one can construct a colored hitting set S' for $\mathcal{F}$ of size $k+l$ as follows. First, add S to S' and then consider the identification number $\mathrm{ID}(j)$ of instance j. For every i between 1 and l consider the i'th bit of $\mathrm{ID}(j)$. If this bit is set to 1 add b_j to S' else 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. Moreover, one can easily prove that S' hits all other sets of $\mathcal{F}$.

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 $\mathrm{ID}(j) = p$ 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 $|S| = k$. □

Theorem 5. [$\star$] 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.*

Theorem 5 has some interesting consequences. For instance, the second part of Theorem 5 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 $poly(k, |H|)$ unless $PH = \Sigma_p^3$.

Theorem 6. *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 the problem one is also given an integer k, and S' may contain not more than k numbers. This parameterization is $W[1]$-hard. We consider a stronger parameterization where in addition to k a parameter d is provided and the integers in S must have size at most 2^d. This version, Small Subset Sum, is trivially fixed parameter tractable by dynamic programming.

Theorem 7. [$\star$] Small Subset Sum *parameterized by* (d, k) *does not admit a kernel polynomial in* (d, k) *unless* $PH = \Sigma_p^3$.

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. 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.
3. N. Betzler. Steiner tree problems in the analysis of biological networks. Diploma thesis, Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany, 2006.
4. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. In *Proc. 35th ICALP*, volume 5125 of *LNCS*, pages 563–574. Springer, 2008.
5. H. L. Bodlaender, S. Thomassé, and A. Yeo. Analysis of data reduction: Transformations give evidence for non-existence of polynomial kernels. Technical Report UU-CS-2008-030, Institute of Information and Computing Sciences, Utrecht University, Netherlands, 2008.
6. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
7. 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*, To appear.
8. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
9. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proc. 40th STOC*, pages 133–142. ACM Press, 2008.
10. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
11. J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.
12. J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of Vertex Cover variants. *Theory Comput. Syst.*, 41(3):501–520, 2007.
13. D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. In *Proc. 47th FOCS*, pages 719–728. IEEE, 2007.
14. 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.
15. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.