Preprocessing of Min Ones Problems: A Dichotomy

Stefan Kratsch and Magnus Wahlström

Max-Planck-Institut für Informatik, Saarbrücken, Germany
{skratsch,wahl}@mpi-inf.mpg.de

Abstract. Min Ones Constraint Satisfaction Problems, i.e., the task of finding a satisfying assignment with at most k true variables (Min Ones SAT(Γ)), can express a number of interesting and natural problems. We study the preprocessing properties of this class of problems with respect to k, using the notion of kernelization to capture the viability of preprocessing. We give a dichotomy of Min Ones SAT(Γ) problems into admitting or not admitting a kernelization with size guarantee polynomial in k, based on the constraint language Γ. We introduce a property of boolean relations called *mergeability* that can be easily checked for any Γ. When all relations in Γ are mergeable, then we show a polynomial kernelization for Min Ones SAT(Γ). Otherwise, any Γ containing a non-mergeable relation and such that Min Ones SAT(Γ) is NP-complete permits us to prove that Min Ones SAT(Γ) does not admit a polynomial kernelization unless NP $\subseteq$ co-NP/poly, by a reduction from a particular parameterization of Exact Hitting Set.

1 Introduction

Preprocessing and data reduction are ubiquitous, especially in the context of combinatorially hard problems. This contrasts the well-known fact that there can be no polynomial-time algorithm that provably shrinks every instance of an NP-hard problem, unless P = NP; we cannot expect every instance to become smaller. An immediate answer to this apparent disconnect between theory and practice is to consider preprocessing to be a heuristic, not open to theoretical performance guarantees, but this is not necessary. A rigorous study is possible, but one has to accept the existence of instances that cannot be reduced any further. Such instances can be thought of as already having small size compared to their inherent combinatorial complexity. In the light of this we should measure the success of preprocessing related to the difficulty of the instances.

This new goal can be formulated nicely using notions from parameterized complexity. An instance of a parameterized problem, say (I, k), features an additional parameter, intended to capture the super-polynomial part of the combinatorial complexity. Thus we can define polynomial-time preprocessing in a way that cannot be simply ruled out under the assumption that P $\neq$ NP: a so-called *polynomial kernelization* is a polynomial-time mapping $K : (I, k) \mapsto (I', k')$ such that (I, k) and (I', k') are equivalent and k' as well as the size of I' are bounded by a polynomial in the parameter k. While a kernelization with some performance guarantee can be obtained by various techniques, achieving the strongest size bounds or at least breaking the polynomial barrier is of high interest. Consider for example the improvements for FEEDBACK VERTEX SET, from the first polynomial kernel [6], to cubic [3], and now quadratic [21]; the existence of a linear kernel is still an open problem.

Recently a seminal paper by Bodlaender, Downey, Fellows, and Hermelin [4] provided a framework to rule out polynomial kernelizations, based on hypotheses in classical complexity. Using results by Fortnow and Santhanam [12], they showed that so-called *compositional* parameterized problems admit no polynomial kernelizations unless NP $\subseteq$ co-NP/poly; by Yap [22], this would imply that the polynomial hierarchy collapses. The existence of such lower bounds has sparked high activity in the field.

Constraint satisfaction problems. CSPs are a fundamental and general problem setting, encompassing a wide range of natural problems, e.g., satisfiability, graph modification, and covering problems. A CSP instance is a formula given as a conjunction of restrictions, called *constraints*, on the feasible assignments to a set of variables. The complexity of a CSP problem depends on the type of constraints that the problem allows. For instance, CLIQUE can be considered as a maximization problem (Max Ones; see below) allowing only constraints of the type $(\neg x \vee \neg y)$. The types of constraints allowed in a problem are captured by a *constraint language* Γ (see Section 2 for definitions).

There are several results that characterize problem properties depending on the constraint language. In 1978, Schaefer [20] classified the problem SAT(Γ) of deciding whether any satisfying assignment exists for a formula as being either in P or NP-complete; as there are no intermediate cases, this is referred to as a *dichotomy.* Khanna et al. [14] classified CSPs according to their approximability, for the problems Min/Max Ones (i.e., finding a satisfying assignment of optimal weight) and Min/Max SAT (i.e., optimizing the number of satisfied or unsatisfied constraints). On the parameterized complexity side, Marx [18] classified the problem of finding a solution with *exactly* k true variables as being either FPT or W[1]-complete. In this paper we are concerned with the problem family Min Ones SAT(Γ):

Input: A formula $\mathcal{F}$ over a finite constraint language Γ; an integer k.
Parameter: k.
Task: Decide whether there is a satisfying assignment for $\mathcal{F}$ with at most k true variables.

We study the kernelization properties of Min Ones SAT(Γ), parameterized by the maximum number of true variables, and classify these problems into admitting or not admitting a polynomial kernelization. Note that Min Ones SAT(Γ) is in FPT for every finite Γ, by a simple branching algorithm [18]. We also point out that Max SAT(Γ), as a subset of Max SNP (cf. [14]), admits polynomial kernelizations for any constraint language Γ [15].

Related work. In the literature there exists an impressive list of problems that admit polynomial kernels (in fact often linear or quadratic); giving stronger and stronger kernels has become its own field of interest. We name only a few results for problems that also have a notion of arity: a kernelization achieving $O(k^{d-1})$ universe size for Hitting Set with set size at most d [1], a kernelization to $O(k^{d-1})$ vertices for packing k vertex disjoint copies of a d-vertex graph [19], and kernelizations to $O(k^d)$ respectively $O(k^{d+1})$ base set size for any problem from MIN $F^+\Pi_1$ or MAX NP with at most d variables per clause [15].

Let us also mention a few lower bound results that are based on the framework of Bodlaender et al. [4]. First of all, Bodlaender et al. [5] provided kernel-preserving reductions, which can be used to extend the applicability of the lower bounds; a similar reduction concept was given by Harnik and Naor [13].

Using this, Dom et al. [9] gave polynomial lower bounds for a number of problems, among them Steiner Tree and Connected Vertex Cover. Furthermore they considered problems that have a $k^{f(d)}$ kernel, where k is the solution size and d is a secondary parameter (e.g., maximum set size), and showed that there is no kernel with size polynomial in $k+d$; for, e.g., Hitting Set, Set Cover, and Unique Coverage. More recently, Dell and van Melkebeek [8] provided polynomial local bounds for a list of problems, implying that d-Hitting Set, for $d \geq 2$, cannot have a kernel with *total* size $O(k^{d-\epsilon})$ for any $\epsilon > 0$ unless the polynomial hierarchy collapses. (Note that this is a bound on total size, i.e., the space required to write down the instance, while the previously cited upper bounds are on the number of vertices.) In [16] the present authors show that a certain Min Ones SAT problem does not admit a polynomial kernel and employ this bound to show that there are $\mathcal{H}$-free edge deletion respectively edge editing problems that do not admit a polynomial kernel.

Our work. We give a complete classification of Min Ones SAT(Γ) problems with respect to polynomial kernelizability. We introduce a property of boolean relations called *mergeability* (see Section 2) and, apart from the hardness dichotomy of Theorem 2, we distinguish constraint languages Γ by being *mergeable* or containing at least one relation that is not mergeable. We prove the following result.

Theorem 1. *For any finite constraint language Γ, Min Ones SAT(Γ) admits a polynomial kernelization if it is in* P *or if all relations in Γ are mergeable. Otherwise it does not admit a polynomial kernelization unless* NP $\subseteq$ *co*-NP*/poly.*

When all relations in Γ are mergeable we are able to provide a new polynomial kernelization based on *sunflowers* and *non-zero-closed cores* (Section 3). We say that constraints form a sunflower when they have the same variables in some positions, the *core*, and the variable sets of the other positions, the *petals*, are disjoint (an analogue of sunflowers from extremal set theory). By an adaption of the Erdős-Rado Sunflower Lemma [11] such sunflowers can be easily found in sets containing more than $O(k^d)$ constraints. If we could then replace these sunflowers by smaller, equivalent constraints (e.g., of lower arity), we would be done. Unlike in the d-Hitting Set case (where this is trivial), this is not always possible in general. However, we show that for mergeable constraints forming a sunflower there is an equivalent replacement that increases the number of *zero-closed* positions; a position of a constraint is *zero-closed* if changing the corresponding variable to zero cannot turn the constraint false. We show how to find sunflowers such that this replacement leads to a simplification of the instance, resulting in a polynomial-time kernelization with $O(k^{d+1})$ variables.

If at least one relation in Γ is not mergeable and Min Ones SAT(Γ) is NP-complete by [14] then we show, using Schaefer-style implementations, that this allows us to implement an n-ary *selection formula* using $O(\log n)$ true local variables (Section 4). A selection formula is simply any kind of formula that is false when no variables are true, and true whenever exactly one variable is true, with arbitrary behavior otherwise; particular examples are disjunctions and parity checks. We show that log-cost selection formulas essentially suffice to allow a kernelization-preserving reduction from Exact Hitting Set (parameterized by the number of sets), and that this problem does not admit a polynomial kernelization unless NP $\subseteq$ co-NP/poly (in a proof following Dom et al. [9]).

2 CSPs and Mergeability

A *constraint* $R(x_1, \ldots, x_d)$ is the application of a relation R to a tuple of variables; a *constraint language* Γ is a set of relations. A *formula over* Γ is a conjunction of constraints using relations $R \in \Gamma$. We consider only finite constraint languages, and only boolean variables. Finally, having fixed a constraint language Γ, a CSP instance is a formula $\mathcal{F}$ over Γ. The problem Min Ones SAT(Γ) was defined in the previous section. As a technical point, we allow repeated variables in our constraints (e.g., $R(x, x, y)$ is allowed).

The NP-hardness of Min Ones SAT(Γ) was characterized in [14]. For the statement of the theorem, we need to define a few basic relation types. A relation R is *0-valid* if $(0, \ldots, 0) \in R$; *Horn* if it can be expressed as a conjunction of clauses containing at most one positive literal each; and *width-2 affine* if it can be expressed as a conjunction of constraints of the form $x = y$, $x \neq y$, and $x = b$ for $b \in \{0, 1\}$. A constraint language Γ is 0-valid (Horn, width-2 affine) if every $R \in \Gamma$ is. Then, the results of [14] are as follows.

Theorem 2 ([14]). *Let Γ be a finite set of relations over the boolean domain. If Γ is zero-valid, Horn, or width-2 affine,then Min Ones SAT(Γ) is in* P *(even with non-negative weights on the variables); otherwise it is* NP*-complete.*

We now define mergeability and give some basic results about it. First, we need some notation: For two tuples $\alpha = (\alpha_1, \ldots, \alpha_r)$, $\beta = (\beta_1, \ldots, \beta_r)$, let $\alpha \wedge \beta = (\alpha_1 \wedge \beta_1, \ldots, \alpha_r \wedge \beta_r)$, and likewise for $\alpha \vee \beta$, and write $\alpha \leq \beta$ if $\alpha_i \leq \beta_i$ for every $1 \leq i \leq r$ (where 0 and 1 are used for false and true values, respectively). Let Γ be a finite set of relations over the boolean domain. We say that Γ *implements* a relation R if R is the set of satisfying assignments for a formula over Γ, i.e., $R(x_1, \ldots, x_r) \equiv \bigwedge_i R_i(x_{i1}, \ldots, x_{it})$ where each $R_i \in \Gamma$ (we do not automatically allow the equality relation unless $= \in \Gamma$). A constraint is

IHSB- (Implicative Hitting Set Bounded-) if it can be implemented by assignments, implications, and negative clauses (i.e., disjunctions of negated variables). Constraint types can also be characterized by closure properties. A constraint R is IHSB- if and only if it is closed under an operation $\alpha \wedge (\beta \vee \gamma)$ for tuples α, β, γ in R, i.e., $\alpha \wedge (\beta \vee \gamma) \in R$ for any choice of $\alpha, \beta, \gamma \in R$. See [7] for more on this.

Definition 1. *Let R be a relation on the boolean domain. Given four (not necessarily distinct) tuples $\alpha, \beta, \gamma, \delta \in R$, we say that the* merge operation *applies if $\alpha \wedge \delta \leq \beta \leq \alpha$ and $\beta \wedge \gamma \leq \delta \leq \gamma$. If so, then* applying *the merge operation produces the tuple $\alpha \wedge (\beta \vee \gamma)$. We say that R is* mergeable *if for any four tuples $\alpha, \beta, \gamma, \delta \in R$ for which the merge operation applies, we have $\alpha \wedge (\beta \vee \gamma) \in R$.*

We give some basic results about mergeability: an alternate presentation of the property, which will be important in Section 3 when sunflowers are introduced, and some basic closure properties. Both propositions are easy to check.

Proposition 1. *Let R be a relation of arity r on the boolean domain. Partition the positions of R into two sets, called the* core *and the* petals*; w.l.o.g. assume that positions 1 through c are the core, and the rest the petals. Let (α_C, α_P), where α_C is a c-ary tuple and α_P an $(r-c)$-ary tuple, denote the tuple whose first c positions are given by α_C, and whose subsequent positions are given by α_P. Consider then the following four tuples.*

$$\begin{aligned} \alpha &= (\alpha_C, \alpha_P) \\ \beta &= (\alpha_C, 0) \\ \gamma &= (\gamma_C, \gamma_P) \\ \delta &= (\gamma_C, 0) \end{aligned}$$

If α through δ are in R, then the merge operation applies, giving us

$$(\alpha_C, \alpha_P \wedge \gamma_P) \in R.$$

Furthermore, for any four tuples to which the merge operation applies, there is a partitioning of the positions into core and petals such that the tuples can be written in the above form.

Proposition 2. *Mergeability is preserved by assignment and identification of variables, i.e., if R is mergeable, then so is any relation produced from R by these operations. Further, any relation implementable by mergeable relations is mergeable.*

Lemma 1 ($\star$[1]). *Any zero-valid relation R which is mergeable is also IHSB-, and can therefore be implemented using negative clauses and implications.*

Examples 1. As basic examples, positive and negative clauses, and implications are mergeable. Thus, the same holds for anything implementable by such relations, such as monotone or antimonotone relations (i.e., $\alpha \in R$ implies $\beta \in R$ for all $\beta \geq \alpha$ resp. $\beta \leq \alpha$). A further positive example is the 3-ary ODD relation ($x \oplus y \oplus z = 1$), but not the 3-ary EVEN or 4-ary ODD relations.

[1] Proof deferred to the full version [17].

3 Kernelization

In this section we show that Min Ones SAT(Γ) admits a polynomial kernelization if all relations in Γ are mergeable. For the purpose of describing our kernelization we first define a sunflower of tuples, similarly to the original sunflower definition for sets. Then we give an adaption of Erdős and Rado's Sunflower Lemma [11] to efficiently find such structures in the given set of constraints. We point out that a similar though more restricted definition for sunflowers of tuples was given by Marx [18]; accordingly the bounds of our sunflower lemma are considerably smaller.

Definition 2. *Let $\mathcal{U}$ be a finite set, let $d \in \mathbb{N}$, and let $\mathcal{H} \subseteq \mathcal{U}^d$. A* sunflower *(*of tuples*)* with cardinality t and core $C \subseteq \{1, \ldots, d\}$ *in $\mathcal{U}$ is a subset consisting of t tuples that have the same element at all positions in C and, in the remaining positions, no element occurs in more than one tuple. The set of remaining positions $P = \{1, \ldots, d\} \setminus C$ is called the* petals.

As an example, $(x_1, \ldots, x_c, y_{11}, \ldots, y_{1p})$, $\ldots$, $(x_1, \ldots, x_c, y_{t1}, \ldots, y_{tp})$ is a sunflower of cardinality t with core $C = \{1, \ldots, c\}$, if all y_{ij} and $y_{i'j'}$ are distinct when $i \neq i'$. Note that, differing from Marx [18] variables in the petal positions may also occur in the core. Observe that every feasible assignment of weight less than t must assign 0 to all variables $y_{i1}, \ldots, y_{ip}$ for some $i \in \{1, \ldots, t\}$. Thus such an assignment must satisfy $R(x_1, \ldots, x_c, 0, \ldots, 0)$ when there are constraints $R(x_1, \ldots, x_c, y_{11}, \ldots, y_{1p})$, $\ldots$, $R(x_1, \ldots, x_c, y_{t1}, \ldots, y_{tp})$.

Lemma 2 ($\star$). *Let $\mathcal{U}$ be a finite set, let $d \in \mathbb{N}$, and let $\mathcal{H} \subseteq \mathcal{U}^d$. If the size of $\mathcal{H}$ is greater than $k^d(d!)^2$, then a sunflower of cardinality $k+1$ in $\mathcal{H}$ can be efficiently found.*

We require some definitions revolving around sunflowers and zero-closed positions to address the issue of simplifying constraints that form a sunflower.

Definition 3. *Let R be an r-ary relation. The relation R is* zero-closed on position i*, if for every tuple $(t_1, \ldots, t_{i-1}, t_i, t_{i+1}, \ldots, t_r) \in R$ we have that R contains also $(t_1, \ldots, t_{i-1}, 0, t_{i+1}, \ldots, t_r)$. A relation is* non-zero-closed *if it has no zero-closed positions. We define functions Δ, Π, and ∇:*

- *$\Delta_P(R)$ is defined to be the* zero-closure of R on positions $P \subseteq \{1, \ldots, r\}$*, i.e., the smallest superset of R that is zero-closed on all positions $i \in P$.*
- *$\Pi(R)$ denotes the* non-zero-closed core*, i.e., the projection of R onto all positions that are not zero-closed or, equivalently, the relation on the non-zero-closed positions obtained by forcing $x_i = 0$ for all zero-closed positions.*
- *$\nabla_C(R)$ denotes the* sunflower restriction of R with core C*: w.l.o.g. the relation given by $[\nabla_C(R)](x_1, \ldots, x_r) = R(x_1, \ldots, x_r) \wedge R(x_1, \ldots, x_c, 0, \ldots, 0)$ for $C = \{1, \ldots, c\}$.*

Variables occurring only in zero-closed positions are easy to handle since setting them to zero does not restrict the assignment to any other variable. To measure the number of non-zero-closed positions in our formula, i.e., occurrences of variables that prevent us from such a replacement, we introduce $\mathcal{Z}(\mathcal{F}, R)$.

Definition 4. *Let $\mathcal{F}$ be a formula and let R be a relation. We define $\mathcal{Z}(\mathcal{F}, R)$ as the set of all tuples $(x_1, \ldots, x_t)$ where $[\Pi(R)](x_1, \ldots, x_t)$ is the non-zero-closed core of an R-constraint in $\mathcal{F}$.*

We prove that mergeable relations admit an implementation of their sunflower restrictions using the respective zero-closures on the petal positions. By choosing carefully the sunflowers that we replace such that the petals contain non-zero-closed positions this will lower $|\mathcal{Z}(\mathcal{F}, R)|$.

Lemma 3 ($\star$). *Let R be a mergeable relation and let $C \cup P$ be a partition of its positions. Then $\Delta_P(\nabla_C(R))$ is mergeable and there is an implementation of $\nabla_C(R)$ using $\Delta_P(\nabla_C(R))$ and implications.*

In the following theorem we establish a preliminary kernel in the form of a formula $\mathcal{F}'$ over some larger constraint language Γ', such that $\mathcal{Z}(\mathcal{F}', R)$ is small for every non-zero-valid relation $R \in \Gamma'$. The language Γ' admits us to apply Lemma 3, as $\Delta_P(\nabla_C(R))$ is not necessarily contained in Γ.

Theorem 3 ($\star$). *Let Γ be a mergeable constraint language with maximum arity d. Let $\mathcal{F}$ be a formula over Γ and let k be an integer. In polynomial time one can compute a formula $\mathcal{F}'$ over a mergeable constraint language $\Gamma' \supseteq \Gamma$ with maximum arity d, such that every assignment of weight at most k satisfies $\mathcal{F}$ if and only if it satisfies $\mathcal{F}'$ and, furthermore, $|\mathcal{Z}(\mathcal{F}', R)| \in O(k^d)$ for every non-zero-valid relation that occurs in $\mathcal{F}'$.*

Proof (sketch). We construct $\mathcal{F}'$, starting from $\mathcal{F}' = \mathcal{F}$. While $|\mathcal{Z}(\mathcal{F}', R)| > k^d(d!)^2$ for any non-zero-valid relation R in $\mathcal{F}'$, search for a sunflower of cardinality $k+1$ in $\mathcal{Z}(\mathcal{F}', R)$, according to Lemma 2. Let C denote the core of the sunflower. Replace each R-constraint whose non-zero-closed core matches a tuple of the sunflower by its sunflower restriction with core C using an implementation according to Lemma 3. Repeating this step until $|\mathcal{Z}(\mathcal{F}', R)| \leq k^d(d!)^2$ for all non-zero-valid relations R in $\mathcal{F}'$ completes the construction.

To show correctness we consider a single replacement. We denote the tuples of the sunflower by $(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip})$, with $i \in \{1, \ldots, k+1\}$, i.e., w.l.o.g. with core $C = \{1, \ldots, c\}$ and petals $P = \{c+1, \ldots, c+p\}$. Let ϕ be any satisfying assignment of weight at most k and consider any tuple $(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip})$ of the sunflower. Let $R(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip}, z_1, \ldots, z_t)$ be a constraint whose non-zero-closed core matches the tuple, w.l.o.g. the last positions of R are zero-closed, let Z be those positions. Since the z_i are in zero-closed positions, ϕ must satisfy $R(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip}, 0, \ldots, 0)$. Observe that, by maximum weight k, the assignment ϕ assigns 0 to all variables $y_{i1}, \ldots, y_{ip}$ for an $i \in \{1, \ldots, k+1\}$. Thus ϕ satisfies $R(x_1, \ldots, x_c, 0, \ldots, 0)$.

Hence for any constraint $R(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip}, z_1, \ldots, z_t)$, the assignment satisfies $\nabla_C(R(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip}, z_1, \ldots, z_t))$ too. This permits us to replace each R-constraint, whose non-zero-closed core matches a tuple of the sunflower, by an implementation of its sunflower restriction with core C, according to Lemma 3. This uses $\Delta_{P \cup Z}(\nabla_C(R(x_1, \ldots, x_c, y_{i1}, \ldots, y_{ip}, z_1, \ldots, z_t)))$ and implications. $\square$

Now we are able to establish polynomial kernelizations for Min Ones SAT(Γ) when Γ is mergeable. For a given instance $(\mathcal{F}, k)$, we first generate an equivalent formula $\mathcal{F}'$ according to Theorem 3. However, $\mathcal{F}'$ will not replace $\mathcal{F}$, rather, it allows us to remove variables from $\mathcal{F}$ based on conclusions drawn from $\mathcal{F}'$.

Theorem 4 ($\star$). *For any mergeable constraint language Γ, Min Ones SAT(Γ) admits a polynomial kernelization.*

Proof (sketch). Let $(\mathcal{F}, k)$ be an instance of Min Ones SAT(Γ) and let d be the maximum arity of relations in Γ. According to Theorem 3, we generate a formula $\mathcal{F}'$, such that assignments of weight at most k are satisfying for $\mathcal{F}$ if and only if they are satisfying for $\mathcal{F}'$. Moreover, for each non-zero-valid relation R, we have that $|\mathcal{Z}(\mathcal{F}', R)| \in O(k^d)$. We allow the constant 0 to be used for replacing variables; a simple construction without using $(x = 0)$ is given in the full proof.

First, according to Lemma 1, we replace each zero-valid constraint of $\mathcal{F}'$ by an implementation through negative clauses and implications. Variables that occur only in zero-closed positions in $\mathcal{F}'$ are replaced by 0, without affecting the possible assignments for the other variables. By equivalence of $\mathcal{F}$ and $\mathcal{F}'$ with respect to assignments of weight at most k, the same is true for $\mathcal{F}$.

Let X be the set of variables that occur in a non-zero-closed position of some non-zero-valid constraint of $\mathcal{F}'$. If a variable $x \in X$ implies at least k other variables, i.e., they have to take value 1 if $x = 1$ by implication constraints in $\mathcal{F}'$, then there is no satisfying assignment of weight at most k for $\mathcal{F}'$ that assigns 1 to x. By equivalence of $\mathcal{F}$ and $\mathcal{F}'$ with respect to such assignments, we replace all such variables by 0. Finally we replace all variables $y \in V(\mathcal{F}') \setminus X$, that are not implied by a variable from X in $\mathcal{F}'$, by the constant 0 in $\mathcal{F}$ and $\mathcal{F}'$. Note that such variables occur only in zero-closed positions and in implications.

Now we prove a bound of $O(k^{d+1})$ on the number of variables in $\mathcal{F}$. First, we observe that all remaining variables of $\mathcal{F}$ must occur in a non-zero-closed position of some constraint of $\mathcal{F}'$. We begin by bounding the number of variables that occur in a non-zero-closed position of some non-zero-valid R-constraint, i.e., the remaining variables of the set X. Observe that such a variable must occur in the corresponding tuple of $\mathcal{Z}(\mathcal{F}', R)$. Since there is only a constant number of relations of arity at most d and since $\mathcal{Z}(\mathcal{F}', R) \in O(k^d)$, this limits the number of such variables by $O(k^d)$. For all other variables, their non-zero-closed occurrences must be in implications, since negative clauses are zero-closed on all positions. Thus, these variables must be implied by a variable of X. Since each variable implies at most $k-1$ other variables, we get an overall bound of $O(k^{d+1})$. Finally, the total size of $\mathcal{F}$ is polynomial for a fix d, since the number of variables is polynomial and the arity of the constraints is bounded. $\square$

4 Kernel Lower Bounds

We will now complete the dichotomy by showing that if Min Ones SAT(Γ) is NP-complete and some $R \in \Gamma$ is not mergeable, then the problem admits no polynomial kernelization unless NP $\subseteq$ co-NP/poly. The central concept of our lower bound construction is the following definition.

Definition 5. *A* selection formula *of arity n is a formula on variable sets X and Y, with $|Y| = n$ and $|X| = n^{O(1)}$, such that there is no solution where $Y = 0$, but for any $y \in Y$ there is a solution where $y = 1$ and $y' = 0$ for any $y' \neq y$, $y' \in Y$; refer to such a solution as* selecting y. *Let the* selection cost *for $y \in Y$ be the minimum number of true variables in X among assignments selecting y. A* log-cost selection formula *is a selection formula where there is a number $w_n = O(\log n)$ such that all selection costs are exactly w_n, and where every solution has at least w_n true variables among X.*

We will show that any Γ as described can be used to construct log-cost selection formulas, and then derive a lower bound from this. The next lemma describes our constructions.

Lemma 4. *The following types of relations can implement log-cost selection formulas of any arity.*
1. A 3-ary relation R_3 with $\{(0,0,0),(1,1,0),(1,0,1)\} \subseteq R_3$ and $(1,0,0) \notin R_3$, together with relations $(x = 1)$ and $(x = 0)$.
2. A 5-ary relation R_5 with $\{(1,0,1,1,0),(1,0,0,0,0),(0,1,1,0,1),(0,1,0,0,0)\} \subseteq R_5$ and $(1,0,1,0,0),(0,1,1,0,0) \notin R_5$, together with relations $(x \neq y)$, $(x = 1)$, and $(x = 0)$.

Proof. Let $Y = \{y_1, \ldots, y_n\}$ be the variables over which a log-cost selection formula is requested. We will create "branching trees" over variables $x_{i,j}$ for $0 \leq i \leq \log_2 n$, $1 \leq j \leq 2^i$, as variants of the composition trees used in [16]. Assume that $n = 2^h$ for some integer h; otherwise pad Y with variables forced to be false, as assumed to be possible in both constructions.

The first construction is immediate. Create the variables $x_{i,j}$ and add a constraint $(x_{0,1} = 1)$. Further, for all i, j with $0 \leq i < h$ and $1 \leq j \leq 2^i$, add a constraint $R_3(x_{i,j}, x_{i+1,2j-1}, x_{i+1,2j})$. Finally, replace variables $x_{h,j}$ by y_j. It can be easily checked that the requirements on R_3 imply that the result is a correct log-cost selection formula with $w_n = h = \log_2 n$.

The second construction uses the same principle, but the construction is somewhat more involved. Create variables $x_{i,j}$ and a constraint $(x_{0,1} = 1)$ as before. In addition, introduce for every $0 \leq i \leq h-1$ two variables l_i, r_i and a constraint $(l_i \neq r_i)$. Now the intention is that (l_i, r_i) decides whether the path of true variables from the root to a leaf should take a left or a right turn after level i. Concretely, add for every i, j with $0 \leq i \leq h-1$ and $1 \leq j \leq 2^i$ a constraint $R_5(l_i, r_i, x_{i,j}, x_{i+1,2j-1}, x_{i+1,2j})$. It can again be verified that this creates a log-cost selection formula with $w_n = 2h = 2\log_2 n$. □

We now reach the technical part, where we show that any relation which is not mergeable can be used to construct a relation as in Lemma 4. The constructions are based on the concept of a *witness* that some relation R lacks a certain closure property. For instance, if R is not mergeable, then there are four tuples $\alpha, \beta, \gamma, \delta \in R$ to which the merge operation applies, but such that $\alpha \wedge (\beta \vee \gamma) \notin R$; these four tuples form a witness that R is not mergeable. Using the knowledge that such witnesses exist, we can use the approach of Schaefer [20], identifying variables according to their occurrence in the tuples of the witness, to build relations with the properties we need.

Lemma 5 (⋆). *Let Γ be a set of relations such that Min Ones SAT(Γ) is* NP-*complete and some $R \in \Gamma$ is not mergeable. Under a constraint that at most k variables are true, Γ can be used to force $(x = 0)$ and $(x = 1)$. Furthermore, there is an implementation of $(x = y)$ using R, $(x = 0)$, and $(x = 1)$.*

Lemma 6 (⋆). *Let Min Ones SAT(Γ) be NP-complete, and not mergeable. Then Min Ones SAT(Γ) can express a log-cost selection formula of any arity.*

We now show our result, using the tools of [5]. We have the following definition. Let $\mathcal{Q}$ and $\mathcal{Q}'$ be parameterized problems. A *polynomial time and parameter transformation* from $\mathcal{Q}$ to $\mathcal{Q}'$ is a polynomial-time mapping $H : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N} : (x, k) \mapsto (x', k')$ such that

$$\forall (x, k) \in \Sigma^* \times \mathbb{N} : ((x, k) \in \mathcal{Q} \Leftrightarrow (x', k') \in \mathcal{Q}') \text{ and } k' \leq p(k),$$

for some polynomial p.

We will provide a polynomial time and parameter transformation to Min Ones SAT(Γ) from Exact Hitting Set(m), defined as follows.

Input: A hypergraph $\mathcal{H}$ consisting of m subsets of a universe U of size n.
Parameter: m.
Task: Decide whether there is a set $S \subset U$ such that $|E \cap S| = 1$ for every $E \in \mathcal{H}$.

It was shown in [5] that polynomial time and parameter transformations preserve polynomial kernelizability, thus our lower bound will follow. We first show that Exact Hitting Set(m) admits no polynomial kernelization; the proof follows the outline of Dom et al. [9].

Lemma 7 (⋆). *Exact Hitting Set(m) admits no polynomial kernelization unless* NP $\subseteq$ *co*-NP*/poly.*

We can now show the main result of this section.

Theorem 5. *Let Γ be a constraint language which is not mergeable. Then Min Ones SAT(Γ) is either in* P*, or does not admit a polynomial kernelization unless* NP $\subseteq$ *co*-NP*/poly.*

Proof. By Theorem 2, Min Ones SAT(Γ) is either polynomial-time solvable or NP-complete; assume that it is NP-complete. By Lemma 5 we have both constants and the constraint $(x = y)$, and by Lemma 6 we can implement log-cost selection formulas. It remains only to describe the polynomial time and parameter transformation from Exact Hitting Set(m) to Min Ones SAT(Γ).

Let $\mathcal{H}$ be a hypergraph. If $\mathcal{H}$ contains more than 2^m vertices, then it can be solved in time polynomial in the input length [2]; otherwise, we create a formula $\mathcal{F}$ and fix a weight k so that $(\mathcal{F}, k)$ is positive if and only if $\mathcal{H}$ has an exact hitting set. Create one variable $y_{i,j}$ in $\mathcal{F}$ for every occurrence of a vertex v_i in an edge E_j in $\mathcal{H}$. For each edge $E \in \mathcal{H}$, create a selection formula over the variables representing the occurrences in E. Finally, for all pairs of occurrences of each vertex v_i, add constraints $(y_{i,j} = y_{i,j'})$, and fix $k = m + \sum_{E \in \mathcal{H}} w_{|E|}$, where w_i is the weight of an i-selection formula. We have an upper bound on the value of k of $O(m \log n) = O(m^2)$.

Now solutions with weight exactly k correspond to exact hitting sets of $\mathcal{H}$. Note that k is the minimum possible weight of the selection formulas, which is taken if exactly one occurrence in each edge is picked. By the definition of log-cost selection formulas, any solution where more than one occurrence has been picked (if such a solution is possible at all) will have a total weight which is larger than this, if the weight of the y-variables is counted as well, and thus such a solution to $\mathcal{F}$ of weight at most k is not possible.

As Exact Hitting Set(m) is NP-complete, it follows from [5] that a polynomial kernelization for Min Ones SAT(Γ) would imply the same for Exact Hitting Set(m), giving our result. □

Finally, let us remark that our lower bound still applies under the restriction that constraints contain no repeated variables. Lemma 5 can be adjusted to provide $(x = 1)$ and $(x = y)$ under this restriction, and we can then use standard techniques (see [16] and Theorem 4) to complete the bound. Such a restriction can be useful in showing hardness of other problems, e.g., as in [16].

5 Conclusions

We presented a dichotomy for Min Ones SAT(Γ) for finite sets of relations Γ, characterized by a new property called mergeability. We showed that Min Ones SAT(Γ) admits a polynomial kernelization if the problem is in P or if every relation in Γ is mergeable, while in every other case no polynomial kernelization is possible, unless NP $\subseteq$ co-NP/poly and the polynomial hierarchy collapses.

An immediate question is the correct size bound for the kernelizable cases. In this paper, the total size is bounded only as a side-effect of having $O(k^{d+1})$ variables, while in [8], it was shown that for a number of problems, including d-Hitting Set, the "correct" bound on the total size of a kernel is $O(k^d)$. Closing this gap, or even characterizing the problems which admit e.g. quadratic total size kernels, would be interesting. Similarly, one can ask whether an explicit super-polynomial (e.g. $2^{(\log m)^{O(1)}}$) or exponential lower bound on the kernelizability of our source problem Exact Hitting Set (m) is possible. For this, and related questions, a study of the structure of problems in FPT under the closure of kernelization-preserving reductions may be useful.

Another question is how the results extend to problems on larger domains, e.g., when variables can take t different values, but at most k may be non-zero.

Acknowledgements. The authors are thankful to Gustav Nordh and Dániel Marx for helpful and interesting discussions.

References

1. F. N. Abu-Khzam. Kernelization algorithms for d-hitting set problems. In *WADS*, volume 4619 of *LNCS*, pages 434–445. Springer, 2007.
2. A. Björklund and T. Husfeldt. Exact algorithms for exact satisfiability and number of perfect matchings. *Algorithmica*, 52(2):226–249, 2008.
3. H. L. Bodlaender. A cubic kernel for feedback vertex set. In *STACS*, volume 4393 of *LNCS*, pages 320–331. Springer, 2007.
4. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels (extended abstract). In *ICALP (1)*, volume 5125 of *LNCS*, pages 563–574. Springer, 2008.
5. 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. Springer, 2009.
6. K. Burrage, V. Estivill-Castro, M. R. Fellows, M. A. Langston, S. Mac, and F. A. Rosamond. The undirected feedback vertex set problem has a poly(k) kernel. In *IWPEC*, volume 4169 of *LNCS*, pages 192–202. Springer, 2006.
7. N. Creignou and H. Vollmer. Boolean constraint satisfaction problems: When does Post's lattice help? In *Complexity of Constraints*, volume 5250 of *LNCS*, pages 3–37. Springer, 2008.
8. H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, 2010. To appear.
9. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and ids. In *ICALP (1)*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.
10. R. G. Downey and M. R. Fellows. *Parameterized Complexity (Monographs in Computer Science)*. Springer, November 1998.
11. P. Erdős and R. Rado. Intersection theorems for systems of sets. *J. London Math. Soc.*, 35:85–90, 1960.
12. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142. ACM, 2008.
13. D. Harnik and M. Naor. On the compressibility of NP instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.
14. S. Khanna, M. Sudan, L. Trevisan, and D. P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.
15. S. Kratsch. Polynomial kernelizations for MIN $\mathrm{F}^{+}\Pi_1$ and MAX NP. In *STACS 2009*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 601–612. Schloss Dagstuhl, Germany, 2009.
16. S. Kratsch and M. Wahlström. Two edge modification problems without polynomial kernels. In *IWPEC*, volume 5917 of *LNCS*, pages 264–275. Springer, 2009.
17. S. Kratsch and M.Wahlström. Preprocessing of min ones problems: a dichotomy. *CoRR*, abs/0910.4518, 2009.
18. D. Marx. Parameterized complexity of constraint satisfaction problems. *Computational Complexity*, 14(2):153–183, 2005.
19. H. Moser. A problem kernelization for graph packing. In *SOFSEM*, volume 5404 of *LNCS*, pages 401–412. Springer, 2009.
20. T. J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226, New York, NY, USA, 1978. ACM.
21. S. Thomassé. A quadratic kernel for feedback vertex set. In *SODA*, pages 115–119. SIAM, 2009.
22. C.-K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.