Kernelization of Constraint Satisfaction Problems: A Study through Universal Algebra *

Victor Lagerkvist[†1] and Magnus Wahlström[‡2]

[1]Institut für Algebra, TU Dresden, Dresden, Germany
[2]Department of Computer Science, Royal Holloway, University of London, Great Britain

Abstract

A *kernelization* algorithm for a computational problem is a procedure which compresses an instance into an equivalent instance whose size is bounded with respect to a complexity parameter. For the Boolean satisfiability problem (SAT), and the constraint satisfaction problem (CSP), there exist many results concerning upper and lower bounds for kernelizability of specific problems, but it is safe to say that we lack general methods to determine whether a given SAT problem admits a kernel of a particular size. This could be contrasted to the currently flourishing research program of determining the classical complexity of finite-domain CSP problems, where almost all non-trivial tractable classes have been identified with the help of algebraic properties. In this paper, we take an algebraic approach to the problem of characterizing the kernelization limits of NP-hard SAT and CSP problems, parameterized by the number of variables. Our main focus is on problems admitting linear kernels, as has, somewhat surprisingly, previously been shown to exist. We show that a CSP problem has a kernel with $O(n)$ constraints if it can be embedded (via a domain extension) into a CSP problem which is preserved by a Maltsev operation. We also study extensions of this towards SAT and CSP problems with kernels with $O(n^c)$ constraints, $c > 1$, based on embeddings into CSP problems preserved by a k-edge operation, $k \leq c + 1$. These results follow via a variant of the celebrated few subpowers algorithm. In the complementary direction, we give indication that the Maltsev condition might be a complete characterization of SAT problems with linear kernels, by showing that an algebraic condition that is shared by all problems with a Maltsev embedding is also necessary for the existence of a linear kernel unless NP $\subseteq$ co-NP/poly.

1 Introduction

Kernelization is a preprocessing technique based on reducing an instance of a computationally hard problem in polynomial time to an equivalent instance, a *kernel*, whose size is bounded by a function f with respect to a given complexity parameter. The function f is referred to as the *size* of the kernel, and if the size is polynomially bounded we say that the problem admits a *polynomial kernel*. A classical example is Vertex Cover, which admits a kernel with $2k$ vertices, where k denotes the size of the cover [29]. Polynomial kernels are of great interest in parameterized complexity, as well as carrying practical significance in speeding up subsequent computations (e.g., the winning contribution in the 2016 PACE challenge for Feedback Vertex Set used a novel kernelization step as a key component (see `https://pacechallenge.wordpress.com/`).

*This is an extended preprint of *Kernelization of Constraint Satisfaction Problems: A Study through Universal Algebra*, appearing in Proceedings of the 23rd International Conference on Principles and Practice of Constraint Programming (CP 2017)

[†]victor.lagerqvist@tu-dresden.de
[‡]magnus.wahlstrom@rhul.ac.uk

When the complexity parameter is a size parameter, e.g., the number of variables n, then such a size reduction is also referred to as *sparsification* (although a sparsification is not always required to run in polynomial time). A prominent example is the famous *sparsification lemma* that underpins research into the Exponential Time Hypothesis [17], which shows that for every k there is a subexponential-time reduction from k-SAT on n variables to k-SAT on $O(n)$ clauses, and hence $\tilde{O}(n)$ bits in size. However, the super-polynomial running time is essential to this result. Dell and van Melkebeek [12] showed that k-SAT cannot be kernelized even down to size $O(n^{k-\varepsilon})$, and Vertex Cover cannot be kernelized to size $O(n^{2-\varepsilon})$, for any $\varepsilon > 0$ unless the polynomial hierarchy collapses (in the sequel, we will make this assumption implicitly). These results suggest that in general, polynomial-time sparsification can give no non-trivial size guarantees. (Note that a kernel of size $O(n^k)$ for k-SAT is trivial.) The first result to the contrary was by Bart Jansen (unpublished until recently [18]), who observed that 1-in-k-SAT admits a kernel with at most n constraints using Gaussian elimination. More surprisingly, Jansen and Pieterse [19] showed that the Not-All-Equal k-SAT problem admits a kernel with $O(n^{k-1})$ constraints, improving on the trivial bound by a factor of n and settling an implicit open problem. In later research, they improved and generalized the method, and also showed that the bound of $O(n^{k-1})$ is tight [18]. These improved upper bounds are all based on rephrasing the SAT problem as a problem of low-degree polynomials, and exploiting linear dependence to eliminate superfluous constraints. Still, it is fair to say that we currently lack the tools for making a general analysis of the kernelizability of a generic SAT problem.

In this paper we take a step in this direction, by studying the kernelizability of the *constraint satisfaction problem* over a constraint language Γ (CSP(Γ)), parameterized by the number of variables n, which can be viewed as the problem of determining whether a set of constraints over Γ is satisfiable. Some notable examples of problems of this kind are k-colouring, k-SAT, 1-in-k-SAT, and not-all-equal-k-SAT. We will occasionally put a particular emphasis on the Boolean CSP problem and therefore denote this problem by SAT(Γ). Note that CSP(Γ) has a trivial polynomial kernel for any finite language Γ (produced by simply discarding duplicate constraints), but the question remains for which languages Γ we can improve upon this. Concretely, our question in this paper is for which languages Γ the problem CSP(Γ) admits a kernel of $O(n^c)$ constraints, for some $c \geq 1$, with a particular focus on linear kernels ($c = 1$).

The algebraic approach in parameterized and fine-grained complexity. For any language Γ, the classical complexity of CSP(Γ) (i.e., whether CSP(Γ) is in P) is determined by the existence of certain algebraic invariants of Γ known as *polymorphisms* [20]. This gave rise to the *algebraic approach* to characterizing the complexity of CSP(Γ) by studying algebraic properties. It has been conjectured that for every Γ, CSP(Γ) is either in P or NP-complete, and that the tractability of a CSP problem can be characterized by a finite list of polymorphisms [8]. Recently, several independent results appeared, claiming to settle this conjecture in the positive [6, 30, 34].

However, for purposes of parameterized and fine-grained complexity questions, looking at polymorphisms alone is too coarse. More technically, the polymorphisms of Γ characterize the expressive power of Γ up to *primitive positive definitions*, i.e., up to the use of conjunctions, equality constraints, and existential quantification, whereas for many questions a liberal use of existentially quantified local variables is not allowed. In such cases, one may look at the expressive power under *quantifier-free* primitive positive definitions (qfpp-definitions), allowing only conjunctions and equality constraints. This expressive power is characterized by more fine-grained algebraic invariants called *partial polymorphisms*. For example, there are numerous dichotomy results for the complexity of *parameterized* SAT(Γ) and CSP(Γ) problems, both for so-called FPT algorithms and for kernelization [23, 24, 25, 28], and in each of the cases listed, a dichotomy is given which is equivalent to requiring a finite list of partial polymorphisms of Γ. Similarly, Jonsson et al. [22] showed that the exact running times of NP-hard SAT(Γ) and CSP(Γ) problems in terms of the number of variables n are characterized by the partial polymorphisms of Γ.

Unfortunately, studying properties of SAT(Γ) and CSP(Γ) for questions phrased in terms of the size parameter n is again more complicated than for more permissive parameters k. For example, it is known that for every finite set P of strictly partial polymorphisms, the number of relations invariant under P is double-exponential in terms of the arity n (hence they cannot all be described in a polynomial number of bits) [26, Lemma 35]. It can similarly be shown that the existence of a polynomial kernel cannot be characterized by such a finite set P. Instead, such a characterization must be given in another way (for example, Lagerkvist et al. [27] provide a way to finitely characterize all partial polymorphisms of a finite Boolean language Γ).

Our results. In Section 3 we generalize and extend the results of Jansen and Pieterse [18] in the case of linear kernels to a general recipe for NP-hard SAT and CSP problems in terms of the existence of a *Maltsev embedding*, i.e., an embedding of a language Γ into a tractable language Γ' on a larger domain with a *Maltsev polymorphism*. We show that for any language Γ with a Maltsev embedding into a finite domain, CSP(Γ) has a kernel with $O(n)$ constraints. More generally, we in Section 4, turn to the problem of finding kernels with $O(n^c)$ constraints ($c > 1$) where we utilize k*-edge* embeddings, and a technique which encompasses the recent results from Jansen and Pieterse, concerning SAT problems representable as low-degree polynomials over a finite field [18]. Attempting an algebraic characterization, we in Section 5 also show an infinite family of *universal* partial operations which are partial polymorphisms of every language Γ with a Maltsev embedding, and show that these operations guarantee the existence of a Maltsev embedding for Γ, albeit into a language with an infinite domain.

Turning to lower bounds against linear kernels, we show that the smallest of these universal partial operations is also necessary, in the sense that for any Boolean language Γ which is not invariant under this operation, SAT(Γ) admits no kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$. We conjecture that this can be completed into a tight characterization – i.e., that at least for Boolean languages Γ, SAT(Γ) admits a linear kernel if and only if it is invariant under all universal partial Maltsev operations.

2 Preliminaries

In this section we introduce the constraint satisfaction problem, kernelization, and the algebraic machinery that will be used throughout the paper.

2.1 Operations and Relations

An n-ary function $f : D^n \to D$ over a domain D is typically referred to as a *operation* on D, although we will sometimes use the terms function and operation interchangeably. We let $\mathrm{ar}(f) = n$ denote the arity of f. Similarly, if $R \subseteq D^n$ is an n-ary relation over D we let $\mathrm{ar}(R) = n$. If $t \in D^n$ is a tuple we let $t[i]$ denote the ith element in t and we let $\mathrm{pr}_{i_1,\ldots,i_{n'}}(t) = (t[i_1], \ldots, t[i_{n'}])$, $n' \leq n$, denote the *projection* of t on (not necessarily distinct) coordinates $i_1, \ldots, i_{n'} \in \{1, \ldots, n\}$. Similarly, if R is an n-ary relation we let $\mathrm{pr}_{i_1,\ldots,i_{n'}}(R) = \{\mathrm{pr}_{i_1,\ldots,i_{n'}}(t) \mid t \in R\}$. We will often represent relations by logical formulas, and if ψ is a first-order formula with free variables $x_1, \ldots, x_k$ we by $R(x_1, \ldots, x_k) \equiv \psi(x_1, \ldots, x_k)$ denote the relation $R = \{(f(x_1), \ldots, f(x_k)) \mid f \text{ is a satisfying assignment to } \psi\}$.

2.2 The Constraint Satisfaction Problem

A set of relations Γ is referred to as a *constraint language*. The *constraint satisfaction problem* over a constraint language Γ over D (CSP(Γ)) is the computational decision problem defined as follows.

INSTANCE: A set V of variables and a set C of constraint applications $R(v_1, \ldots, v_k)$ where $R \in \Gamma$, $\mathrm{ar}(R) = k$, and $v_1, \ldots, v_k \in V$.
QUESTION: Is there a function $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for each $R(v_1, \ldots, v_k)$ in C?

In the particular case when Γ is Boolean we denote CSP(Γ) by SAT(Γ), and we let BR denote the set of all Boolean relations. As an example, first consider the ternary relation $R_{1/3} = \{(0,0,1),(0,1,0),(1,0,0)\}$. It is then readily seen that SAT($\{R_{1/3}\}$) can be viewed as an alternative formulation of the 1-in-3-SAT problem restricted to instances consisting only of positive literals. More generally, if we let $R_{1/k} = \{(x_1, \ldots, x_k) \in \{0,1\}^k \mid x_1 + \ldots + x_k = 1\}$, then SAT($\{R_{1/k}\}$) is a natural formulation of 1-in-k-SAT without negation.

2.3 Kernelization

A *parameterized problem* is a subset of $\Sigma^* \times \mathbb{N}$ where Σ is a finite alphabet. Hence, each instance is associated with a natural number, called the *parameter*.

Definition 1. *A* kernelization algorithm*, or a* kernel*, for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time algorithm which, given an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, computes $(x', k') \in \Sigma^* \times \mathbb{N}$ such that (1) $(x, k) \in L$ if and only if $(x', k') \in L$ and (2) $|x'| + k' \leq f(k)$ for some function f.*

The function f in the above definition is sometimes called the *size* of the kernel. In this paper, we are mainly interested in the case where the parameter denotes the number of variables in a given instance.

2.4 Polymorphisms and Partial Polymorphisms

In this section we define the link between constraint languages and algebras that was promised in Section 1. If f is an n-ary operation and $t_1, \ldots, t_n$ a sequence of k-ary tuples we can in a natural way obtain a k-ary tuple by applying f componentwise, i.e., $f(t_1, \ldots, t_n) = (f(t_1[1], \ldots, t_n[1]), \ldots, f(t_1[k], \ldots, t_n[k]))$.

Definition 2. *An n-ary operation f is a* polymorphism *of a k-ary relation R if $f(t_1, \ldots, t_n) \in R$ for each sequence of tuples $t_1, \ldots, t_n \in R$.*

If f is a polymorphism of R we also say that R is *invariant* under f, or that f *preserves* R, and for a constraint language Γ we let $\mathrm{Pol}(\Gamma)$ denote the set of operations preserving every relation in Γ. Similarly, if F is a set of functions, we let $\mathrm{Inv(F)}$ denote the set of all relations invariant under F. Sets of functions of the form $\mathrm{Pol}(\Gamma)$ are referred to as *clones*. It is well known that $\mathrm{Pol}(\Gamma)$ (1) for each $n \geq 1$ and each $1 \leq i \leq n$ contains the *projection function* $\pi_i^n(x_1, \ldots, x_i, \ldots, x_n) = x_i$, and (2) if $f, g_1, \ldots, g_m \in \mathrm{Pol}(\Gamma)$, where $\mathrm{ar}(f) = m$ and all g_i have the same arity n, then the *composition* $f \circ g_1, \ldots, g_m(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ is included in $\mathrm{Pol}(\Gamma)$. Similarly, sets of the form $\mathrm{Inv(F)}$ are referred to as *relational clones*, or *co-clones*, and are sets of relations closed under *primitive positive definitions* (pp-definitions), which are logical formulas consisting of existential quantification, conjunction, and equality constraints. In symbols, we say that a k-ary relation R has a pp-definition over a constraint language Γ over a domain D if $R(x_1, \ldots, x_k) \equiv \exists y_1, \ldots, y_{k'} \,.\, R_1(\mathbf{x_1}) \wedge \ldots \wedge R_m(\mathbf{x_m})$, where each $R_i \in \Gamma \cup \{\mathrm{Eq}\}$, $\mathrm{Eq} = \{(\mathrm{x}, \mathrm{x}) \mid \mathrm{x} \in \mathrm{D}\}$ and each $\mathbf{x_i}$ is an $\mathrm{ar}(R_i)$-ary tuple of variables over $x_1, \ldots, x_k$, $y_1, \ldots, y_{k'}$. Clones and co-clones are related via the following *Galois connection*.

Theorem 3 ([3, 4, 14])**.** *Let Γ and Γ' be two constraint languages. Then $\Gamma \subseteq \mathrm{Inv}(\mathrm{Pol}(\Gamma'))$ if and only if $\mathrm{Pol}(\Gamma') \subseteq \mathrm{Pol}(\Gamma)$.*

As a shorthand, we let $[F] = \mathrm{Pol}(\mathrm{Inv(F)})$ denote the smallest clone containing F and $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{Pol}(\Gamma))$ the smallest co-clone containing Γ. Using Theorem 3 Jeavons et al. proved that if Γ and Γ' are two finite constraint languages and $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Gamma')$, then CSP($\Gamma'$) is polynomial-time many-one reducible to CSP(Γ) [20]. As remarked in Section 1, while this theorem is useful for establishing complexity dichotomies for CSP and related problems [1, 10], it offers little information on whether a problem admits a kernel of a particular size. Hence, in order to have any hope of studying kernelizability of SAT problems, we need algebras more fine-grained than polymorphisms. In our case these algebras will consist of *partial operations* instead of total operations. An n-ary partial operation over a set D of values is a map of the form $f : X \to D$, where $X \subseteq D^n$ is called the *domain* of f. As in the case of total operations we let $\mathrm{ar}(f) = n$, and furthermore let $\mathrm{domain}(f) = X$. If f and g are n-ary partial operations such that $\mathrm{domain}(g) \subseteq \mathrm{domain}(f)$ and $f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n)$ for each $(x_1, \ldots, x_n) \in \mathrm{domain}(g)$, then g is said to be a *subfunction* of f.

Definition 4. *An n-ary partial operation f is a* partial polymorphism *of a k-ary relation R if, for every sequence $t_1, \ldots, t_n \in R$, either $f(t_1, \ldots, t_n) \in R$ or there exists $i \in \{1, \ldots, k\}$ such that $(t_1[i], \ldots, t_n[i]) \notin \mathrm{domain}(f)$.*

Again, this notion easily generalizes to constraint languages, and if we let $\mathrm{pPol}(\Gamma)$ denote the set of partial polymorphisms of the constraint language Γ, we obtain a *strong partial clone*. It is known that strong partial clones are sets of partial operations which are (1) closed under composition of partial operations and (2) containing all partial projection functions [31]. More formally, the first condition means that if $f, g_1, \ldots, g_m$ are included in the strong partial clone, where f is m-ary and every g_i is n-ary, then the function $f \circ g_1, \ldots, g_m(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ is also included in the strong partial clone, and this function will be defined for $(x_1, \ldots, x_n) \in D^n$ if and only if $(x_1, \ldots, x_n) \in \bigcap_{i=1}^{m} \mathrm{domain}(g_i)$ and $(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)) \in \mathrm{domain}(f)$. The

second condition, containing all partial projection functions, is known to be equivalent to closure under taking subfunctions; a property which in the literature is sometimes called *strong*.

If F is a set of partial functions we let $\mathrm{Inv(F)}$ denote the set of all relations invariant under F, but this time $\mathrm{Inv(F)}$ is in general not closed under pp-definitions, but under *quantifier-free primitive positive definitions* (qfpp-definitions). As the terminology suggests, a relation R has a qfpp-definition over Γ if R is definable via a pp-formula which does not make use of existential quantification. Such formulas are sometimes simply called *conjunctive formulas*. We have the following Galois connection.

Theorem 5 ([14, 31]). *Let Γ and Γ' be two constraint languages. Then $\Gamma \subseteq \mathrm{Inv}(\mathrm{pPol}(\Gamma'))$ if and only if $\mathrm{pPol}(\Gamma') \subseteq \mathrm{pPol}(\Gamma)$.*

As a shorthand we let $\mathrm{Inv}(\mathrm{pPol}(\Gamma)) = \langle\Gamma\rangle_{\exists}$. Jonsson et al. [22] showed the following useful theorem.

Theorem 6. *Let Γ and Γ' be two finite constraint languages. If $\mathrm{pPol}(\Gamma) \subseteq \mathrm{pPol}(\Gamma')$ then there exists a polynomial-time many-one reduction from SAT(Γ') to SAT(Γ) which maps an instance (V, C) of SAT(Γ') to an instance (V', C') of SAT(Γ) where $|V'| \leq |V|$ and $|C'| \leq c|C|$, where c depends only on Γ and Γ'.*

2.5 Maltsev Operations, Signatures and Compact Representations

A *Maltsev operation* over $D \supseteq \{0, 1\}$ is a ternary operation ϕ which for all $x, y \in D$ satisfies the two identities $\phi(x, x, y) = y$ and $\phi(x, y, y) = x$. Before we can explain the powerful, structural properties of relations invariant under Maltsev operations, we need a few technical definitions from Bulatov and Dalmau [7]. Let t, t' be two n-ary tuples over D. We say that (t, t') *witnesses* a tuple $(i, a, b) \in \{1, \ldots, n\} \times D^2$ if $\mathrm{pr}_{1,\ldots,i-1}(t) = \mathrm{pr}_{1,\ldots,i-1}(t')$, $t[i] = a$, and $t'[i] = b$. The *signature* of an n-ary relation R over D is then defined as

$$\mathrm{Sig}(R) = \{(i, a, b) \in \{1, \ldots, n\} \times D^2 \mid \exists t, t' \in R \text{ such that } (t, t') \text{ witnesses } (i, a, b)\},$$

and we say that $R' \subseteq R$ is a *representation* of R if $\mathrm{Sig}(R) = \mathrm{Sig}(R')$. If R' is a representation of R it is said to be *compact* if $|R'| \leq 2|\mathrm{Sig}(R)|$, and it is known that every relation invariant under a Maltsev operation admits a compact representation. Furthermore, we have the following theorem from Bulatov and Dalmau, where we let $\langle R\rangle_f$ denote the smallest superset of R preserved under the operation f.

Theorem 7 ([7]). *Let ϕ be a Maltsev operation over a finite domain, $R \in \mathrm{Inv}(\{\phi\})$ a relation, and R' a representation of R. Then $\langle R'\rangle_\phi = R$.*

Hence, relations invariant under Maltsev operations are reconstructible from their representations.

3 Maltsev Embeddings and Kernels of Linear Size

In this section we give general upper bounds for kernelization of NP-hard SAT problems based on algebraic conditions. We begin in Section 3.1 by outlining the polynomial-time algorithm for Maltsev constraints, and in Section 3.2 this algorithm is modified to construct linear-sized kernels for certain SAT(Γ) problems.

3.1 The Simple Algorithm for Maltsev Constraints

At this stage the connection between Maltsev operations, compact representations and tractability of Maltsev constraints might not be immediate to the reader. We therefore give a brief description of the simple algorithm for Maltsev constraints from Bulatov and Dalmau [7], which will henceforth simply be referred to as the *Maltsev algorithm*. In a nutshell, the algorithm operates as follows, where ϕ is a Maltsev operation over a finite set D.

1. Let $(V, \{C_1, \ldots, C_m\})$ be an instance of CSP($\mathrm{Inv}(\{\phi\})$), and S_0 a compact representation of $D^{|V|}$.
2. For each $i \in \{1, \ldots, m\}$ compute a compact representation S_i of the solution space of the instance $(V, \{C_1, \ldots, C_i\})$ using S_{i-1}.
3. Answer yes if $S_m \neq \emptyset$ and no otherwise.

The second step is accomplished by removing the tuples from $\langle S_{i-1} \rangle_\phi$ that are not compatible with the constraint C_i. While the basic idea behind the Maltsev algorithm is not complicated, the intricate details of the involved subprocedures are outside the scope of this paper, and we refer the reader to Bulatov and Dalmau [7]. We note that although the algorithm applies to infinite languages, it is assumed that the relations in the input are specified by explicit lists of tuples, i.e., the running time includes a factor proportional to $\max |R|$ over relations R used in the input.

Example 8. *Let $G = (D, \cdot)$ be a group over a finite set D, i.e., $\cdot$ is a binary, associative operator, D is closed under $\cdot$ and contains an identity element 1_G, and each element $x \in D$ has an inverse element $x^{-1} \in D$ such that $x \cdot x^{-1} = 1_G$. The ternary operation $s(x, y, z) = x \cdot y^{-1} \cdot z$ is referred to as the* coset generating operation *of G, and is Maltsev since $s(x, y, y) = x \cdot y^{-1} \cdot y = x$ and $s(x, x, y) = x \cdot x^{-1} \cdot y = y$. The problem CSP$(\mathrm{Inv}(\{\mathrm{s}\}))$ is known to be tractable via the algorithm from Feder and Vardi [13], but since s is a Maltsev operation CSP$(\mathrm{Inv}(\{\mathrm{s}\}))$ can equivalently well be solved via the Maltsev algorithm.*

Another early class of tractable CSP problems was discovered via the observation that if R is preserved by a certain Maltsev operation, it can be viewed as the solution space of a system of linear equations.

Example 9. *An* Abelian *group $G = (D, +)$ is a group where $+$ is commutative. Similar to Example 8 we can consider the coset generating operation $s(x, y, z) = x - y + z$, where $-y$ denotes the inverse of the element y. If $|D|$ is prime it is known that $R \in \mathrm{Inv}(\{\mathrm{s}\})$ if and only if R is the solution space of a system of linear equations modulo $|D|$ [21]. Hence, the problem CSP$(\mathrm{Inv}(\{\mathrm{s}\}))$ can efficiently be solved with Gaussian elimination, but can also be solved via the Maltsev algorithm.*

3.2 Upper Bounds Based on Maltsev Embeddings

In this section we use a variation of the Maltsev algorithm to obtain kernels of SAT problems. First, observe that Γ is never preserved by a Maltsev operation when SAT(Γ) is NP-hard [32]. However, it is sometimes possible to find a related constraint language $\hat{\Gamma}$ which is preserved by a Maltsev operation. This will allow us to use the advantageous properties of relations invariant under Maltsev operations in order to compute a kernel for the original SAT(Γ) problem. We thus begin by making the following definition.

Definition 10. *A constraint language Γ over a domain D admits an* embedding *over the constraint language $\hat{\Gamma}$ over $D' \supseteq D$ if there exists a bijective function $h : \Gamma \to \hat{\Gamma}$ such that $\mathrm{ar}(h(R)) = \mathrm{ar}(R)$ and $h(R) \cap D^{\mathrm{ar}(R)} = R$ for every $R \in \Gamma$.*

If $\hat{\Gamma}$ is preserved by a Maltsev operation then we say that Γ admits a *Maltsev embedding*. In general, we do not exclude the possibility that the domain D' is infinite. In this section, however, we will only be concerned with finite domains, and therefore do not explicitly state this assumption. If the bijection h is efficiently computable and there exists a polynomial p such that $h(R)$ can be computed in $O(p(|R|))$ time for each $R \in \Gamma$, then we say that Γ admits a *polynomially bounded* embedding. In particular, an embedding over a finite domain of any finite Γ is polynomially bounded.

Example 11. *Recall from Section 2.2 that $R_{1/3}$ consists of the three tuples $(0, 0, 1), (0, 1, 0)$, and $(1, 0, 0)$. We claim that $R_{1/3}$ has a Maltsev embedding over $\{0, 1, 2\}$. Let $\hat{R}_{1/3} = \{(x, y, z) \in \{0, 1, 2\}^3 \mid x + y + z = 1 \pmod 3\}$. By definition, $\hat{R}_{1/3} \cap \{0, 1\}^3 = R_{1/3}$, so all that remains to prove is that $\hat{R}_{1/3}$ is preserved by a Maltsev operation. But recall from Example 9 that a relation R is the solution space of a system of linear equations over D, where $|D|$ is prime, if and only if R is preserved by the operation $x - y + z$ over D. Hence, $\hat{R}_{1/3}$ is indeed a Maltev embedding of $R_{1/3}$. More generally, one can also prove that $R_{1/k}$ has a Maltsev embedding to a finite domain D where $|D| \geq k$ and $|D|$ is prime.*

Given an instance $I = (\{x_1, \ldots, x_n\}, C)$ of CSP(Γ) we let $\Psi_I = \{(g(x_1), \ldots, g(x_n)) \mid g \text{ satisfies } I\}$. If ϕ is a Maltsev operation and $I = (V, \{C_1, \ldots, C_m\})$ an instance of CSP$(\mathrm{Inv}(\{\phi\}))$ we let $\mathrm{Seq}(I) = (S_0, S_1 \ldots, S_m)$ denote the compact representations of the relations $\Psi_{(V,\emptyset)}, \Psi_{(V,\{C_1\})}, \ldots, \Psi_{(V,\{C_1,\ldots,C_m\})}$ computed by the Maltsev algorithm. We remark that the ordering chosen in the sequence $\mathrm{Seq}I$ does not influence the upper bound in the forthcoming kernelization algorithm.

Definition 12. *Let ϕ be a Maltsev operation, p a polynomial and let $\Delta \subseteq \mathrm{Inv}(\{\phi\})$. We say that Δ and CSP(Δ) have* chain length *p if $|\{\langle S_i \rangle_\phi \mid i \in \{0, 1, \ldots, |C|\}\}| \leq p(|V|)$ for each instance $I = (V, C)$ of CSP(Δ), where $\mathrm{Seq}(I) = (S_0, S_1, \ldots, S_{|C|})$.*

We now have everything in place to define our kernelization algorithm.

Theorem 13. *Let Γ be a Boolean constraint language which admits a polynomially bounded Maltsev embedding $\hat{\Gamma}$ with chain length p. Then SAT(Γ) has a kernel with $O(p(|V|))$ constraints.*

Proof. Let $\phi \in \mathrm{Pol}(\hat{\Gamma})$ denote the Maltsev operation witnessing the embedding $\hat{\Gamma}$. Given an instance $I = (V, C)$ of SAT(Γ) we can obtain an instance $I' = (V, C')$ of CSP($\hat{\Gamma}$) by replacing each constraint $R_i(\mathbf{x_i})$ in C by $\hat{R}_i(\mathbf{x_i})$. We arbitrarily order the constraints as $C' = (C_1, \ldots, C_m)$ where $m = |C'|$. We then iteratively compute the corresponding sequence $\mathrm{Seq}(I') = (S_0, S_1, \ldots, S_{|C'|})$. This can be done in polynomial time with respect to the size of I via the same procedure as the Maltsev algorithm. For each $i \in \{1, \ldots, m\}$ we then do the following.

1. Let the ith constraint be $C_i = \hat{R}_i(x_{i_1}, \ldots, x_{i_r})$ with $\mathrm{ar}(R_i) = r$.
2. For each $t \in S_{i-1}$ determine whether $\mathrm{pr}_{i_1,\ldots,i_r}(t) \in \hat{R}_i$.
3. If yes, then remove the constraint C_i, otherwise keep it.

This can be done in polynomial time with respect to the size of the instance I', since (1) $|S_{i-1}|$ is bounded by a polynomial in $|V|$ and (2) the test $\mathrm{pr}_{i_1,\ldots,i_r}(t) \in \hat{R}_i$ can naively be checked in linear time with respect to $|\hat{R}_i|$. We claim that the procedure outlined above will correctly detect whether the constraint C_i is redundant or not with respect to $\langle S_{i-1}\rangle_\phi$, i.e., whether $\langle S_{i-1}\rangle_\phi = \langle S_i\rangle_\phi$. First, observe that if there exists $t \in S_{i-1}$ such that $\mathrm{pr}_{i_1,\ldots,i_r}(t) \notin \hat{R}_i$, then the constraint is clearly not redundant. Hence, assume that $\mathrm{pr}_{i_1,\ldots,i_r}(t) \in \hat{R}_i$ for every $t \in S_{i-1}$. Then $S_{i-1} \subseteq \langle S_i\rangle_\phi$, hence also $\langle S_{i-1}\rangle_\phi \subseteq \langle S_i\rangle_\phi$. On the other hand, $\langle S_i\rangle_\phi \subseteq \langle S_{i-1}\rangle_\phi$ holds trivially. Therefore, equality must hold.

Let $I'' = (V, C'')$ denote the resulting instance. Since CSP($\mathrm{Inv}(\{\phi\})$) has chain length p it follows that (1) the sequence $\langle S_0\rangle_\phi, \langle S_1\rangle_\phi, \ldots, \langle S_{|C'|}\rangle_\phi$ contains at most $p(|V|)$ distinct elements, hence $|C''| \leq p(|V|)$, and (2) $\Psi_{I'} = \Psi_{I''}$. Clearly, it also holds that $\Psi_I = (\Psi_{I'} \cap \{0,1\}^{|V|}) = (\Psi_{I''} \cap \{0,1\}^{|V|})$. Hence, we can safely transform I'' to an instance I^* of SAT(Γ) by replacing each constraint $\hat{R}_i(\mathbf{x_i})$ with $R_i(\mathbf{x_i})$. Then I^* is an instance of SAT(Γ) with at most $p(|V|)$ constraints, such that $\Psi_I = \Psi_{I^*}$. In particular, I^* has a solution if and only if I has a solution. □

Clearly, the above algorithm also works for finite-domain CSP. As with the Maltsev algorithm, the procedure runs in polynomial time with respect to the total size of the instance. For languages with bounded arity this simply means time polynomial in n, but it is worth noting that if Γ is infinite but somehow concisely encoded, then we cannot necessarily check whether an n-ary constraint is redundant in time polynomial in n. All that remains to be proven now is that there actually exist Maltsev embeddings with bounded chain length.

Definition 14. *Let f be an n-ary operation over D. A binary relation $R \in \mathrm{Inv}(\{\mathrm{f}\})$ is said to be a* congruence *of f if it is an equivalence relation over D.*

Before we prove Theorem 17, we need two subsidiary lemmas.

Lemma 15. *Let ϕ be a Maltsev operation over D and I an instance of CSP($\mathrm{Inv}(\{\phi\})$). Then $\mathrm{Sig}(S_{i-1}) \supseteq \mathrm{Sig}(S_i)$ for each S_{i-1} in $\mathrm{Seq}(I)$.*

Proof. Let $I = (V, C)$, $(j, a, b) \in \mathrm{Sig}(S_i)$, where $j \in \{1, \ldots, |V|\}$ and $a, b \in D$. Then there exists $t, t' \in S_i$ such that (t, t') witnesses (j, a, b), i.e., $\mathrm{pr}_{1,\ldots,j-1}(t) = \mathrm{pr}_{1,\ldots,j-1}(t')$, and $t[j] = a$, $t'[j] = b$. Since $\langle S_{i-1}\rangle_\phi \supseteq \langle S_i\rangle_\phi \supseteq S_i$, it follows that $t, t' \in \langle S_{i-1}\rangle_\phi$, and hence also that $(j, a, b) \in \mathrm{Sig}(\langle S_{i-1}\rangle_\phi)$. But since S_{i-1} is a representation of $\langle S_{i-1}\rangle_\phi$, $\mathrm{Sig}(S_{i-1}) = \mathrm{Sig}(\langle S_{i-1}\rangle_\phi)$, from which we infer that $(j, a, b) \in \mathrm{Sig}(S_{i-1})$. □

Lemma 16. *Let ϕ be a Maltsev operation over a finite domain D, and $R \in \mathrm{Inv}(\{\phi\})$. For every $i \in \{1, \ldots, \mathrm{ar}(R)\}$, the tuples (i, a, b) in $\mathrm{Sig}(R)$ define an equivalence relation on $\mathrm{pr}_i(R) \subseteq D$.*

Proof. Define the relation $a \sim b$ if and only if $(i, a, b) \in \mathrm{Sig}(R)$. Note that $(i, a, a) \in \mathrm{Sig}(R)$ if and only if $a \in \mathrm{pr}_i(R)$, and that $(i, a, b) \notin \mathrm{Sig}(R)$ for any b if $a \notin \mathrm{pr}_i(R)$. Also note that $\sim$ is symmetric by its definition. It remains to show transitivity. Let $(i, a, b) \in \mathrm{Sig}(R)$ be witnessed by (t_a, t_b) and $(i, a, c) \in$

$\mathrm{Sig}(R)$ be witnessed by (t'_a, t'_c). We claim that $t_c := \phi(t_a, t'_a, t'_c) \in R$ is a tuple such that (t_b, t_c) witnesses $(i, b, c) \in \mathrm{Sig}(R)$. Indeed, for every $j < i$ we have $\phi(t_a[j], t'_a[j], t'_c[j]) = \phi(t_a[j], t'_a[j], t'_a[j]) = t_a[j]$, whereas $\phi(t_a[i], t'_a[i], t'_c[i]) = (a, a, c) = c$. Since $t_a[j] = t_b[j]$ for every $j < i$, it follows that (t_b, t_c) witnesses $(i, b, c) \in \mathrm{Sig}(R)$. Hence $\sim$ is an equivalence relation on $\mathrm{pr}_i(R)$. □

Theorem 17. *Let ϕ be a Maltsev operation over a finite domain D. Then CSP($\mathrm{Inv}(\{\phi\})$) has chain length $O(|D||V|)$.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathrm{Inv}(\{\phi\})$), with $|V| = n$ and $|C| = m$, and let $\mathrm{Seq}(I) = (S_0, S_1, \ldots, S_m)$ be the sequence of compact representations computed by the Maltsev algorithm. By Lemma 15, $\mathrm{Sig}(S_{i+1}) \subseteq \mathrm{Sig}(S_i)$ for every $i < m$, and by Lemma 16, the sets $(j, a, b) \in \mathrm{Sig}(S_i)$ induce an equivalence relation on $\mathrm{pr}_j(\langle S_i \rangle_\phi)$ for every $i \leq m$, $j \leq n$. (Lemma 16 applies here since $\mathrm{Sig}(S_i) = \mathrm{Sig}(\langle S_i \rangle_\phi)$ for every S_i in $\mathrm{Seq}(I)$, and $\langle S_i \rangle_\phi \in \mathrm{Inv}(\{\phi\})$.) We also note that if $\mathrm{Sig}(S_{i+1}) = \mathrm{Sig}(S_i)$, then $\langle S_i \rangle_\phi = \langle S_{i+1} \rangle_\phi$ since S_{i+1} is a compact representation of $\langle S_i \rangle_\phi$. Hence, we need to bound the number of times that $\mathrm{Sig}(S_{i+1}) \subset \mathrm{Sig}(S_i)$ can hold. Now note that whenever $\mathrm{Sig}(S_{i+1}) \subset \mathrm{Sig}(S_i)$, then either $\mathrm{pr}_j(\langle S_i \rangle_\phi) \subset \mathrm{pr}_j(\langle S_{i+1} \rangle_\phi)$ for some j, or the equivalence relation induced by tuples $(j, a, b) \in \mathrm{Sig}(S_{i+1})$ is a refinement of that induced by tuples $(j, a, b) \in \mathrm{Sig}(S_i)$ for some j. Both of these events can only occur $|D| - 1$ times for every position j (unless $S_m = \emptyset$). Hence the chain length is bounded by $2|V||D|$. □

This bound can be slightly improved for a particular class of Maltsev operations. Recall from Example 8 that $s(x, y, z) = x \cdot y^{-1} \cdot z$ is the coset generating operation of a group $G = (D, \cdot)$.

Lemma 18. *Let $G = (D, \cdot)$ be a finite group and let s be its coset generating operation. Then CSP($\mathrm{Inv}(\{\mathrm{s}\})$) has chain length $O(|V| \log |D|)$.*

Proof. Let $I = (V, C)$ be an instance of CSP($\mathrm{Inv}(\{\mathrm{s}\})$), where $|V| = n$ and $|C| = m$. Let $\mathrm{Seq}(I) = (S_0, S_1, \ldots, S_m)$ be the corresponding sequence. First observe that S_0 is a compact representation of D^n and that $(D^n, \cdot)$ is nothing else than the nth direct power of G. It is well-known that R is a coset of a subgroup of $(D^n, \cdot)$ if and only if s preserves R [11]. In particular, this implies that S_1 is a compact representation of a subgroup of $(D^n, \cdot)$, and more generally that each S_i is a compact representation of a subgroup of $\langle S_{i-1} \rangle_s$. An application of Lagrange's theorem reveals that $|\langle S_i \rangle_s|$ divides $|\langle S_{i-1} \rangle_s|$, which implies that the sequence $\langle S_0 \rangle_s, \langle S_1 \rangle_s, \ldots, \langle S_m \rangle_s$ contains at most $n \log_2 |D| + 1$ distinct elements. □

Note that if the domain $|D|$ is prime in Lemma 18 then the proof can be strengthened to obtain the bound $O(|V|)$. As an application of this result, let us briefly return to Example 11, where we demonstrated that $R_{1/k}$ had a Maltsev embedding over the coset generating operation of an Abelian group $(D, +)$ where $|D|$ is prime. Combining Theorem 13 and Lemma 18 we therefore conclude that SAT($\{R_{1/k}\}$) has a kernel with $O(|V|)$ constraints. More generally, we may interpret the results in this section as follows. If Γ admits a Maltsev embedding over the coset generating operation of an Abelian group $(D, +)$, where $|D|$ is prime, then we obtain kernels with $O(|V|)$ constraints, closely mirroring the results from Jansen and Pieterse [18]. This is in turn a special case of constraint languages admitting Maltsev embeddings over coset generating operations over arbitrary groups, where we obtain kernels with $O(|V| \log |D|)$ constraints. It is not hard to find examples of groups whose coset generating operations cannot be represented by the aforementioned Abelian groups. One such example is the group A_n of all even permutations over $\{1, \ldots, n\}$ for $n \geq 3$. Last, in the most general case, where we obtain kernels with $O(|V||D|)$ constraints, we have embeddings over arbitrary Maltsev operations. Furthermore, it is known that a Maltsev operation ϕ over D is the coset generating operation of a group $(D, \cdot)$ if and only if $\phi(\phi(x, y, z), z, u) = \phi(x, y, u)$, $\phi(u, z, \phi(z, y, x)) = \phi(u, y, x)$ for all $x, y, z, u \in D$ [11]. Hence, any Maltsev operation which do not satisfy any of these two identities cannot be viewed as a coset generating operation of some group.

4 Kernels of Polynomial Size

Section 3.2 gives a description of SAT problems admitting kernels with $O(n)$ constraints. In this section we study two generalizations which provide kernels with $O(n^c)$ constraints for $c > 1$.

4.1 Moving Beyond Maltsev: k-Edge Embeddings

It is known that Maltsev operations are particular examples of a more general class of operations called *k-edge operations*. Following Idziak et al. [2] we define a k-edge operation e as a $(k+1)$-ary operation satisfying $e(x,x,y,y,y,\ldots,y,y) = e(x,y,x,y,y,\ldots,y,y) = y$ and for each $i \in \{4,\ldots,k+1\}$, $e(y,\ldots,y,x,y,\ldots,y) = y$, where x occurs in position i. Note that a Maltsev operation is nothing else than a 2-edge operation with the first and second arguments permuted. A *k-edge embedding* is then defined analogously to the concept of a Maltsev embedding, with the distinction that the embedding $\hat{\Gamma}$ must be preserved by a k-edge operation for some $k \geq 2$. It is known that k-edge operations satisfy many of the advantageous properties of Maltsev operations, and the basic definitions concerning signatures and representations are similar. Before the proof of Theorem 21 we need the following theorem from Idziak et al. [2].

Theorem 19. *[2] If e is a k-edge operation over D then $[\{e\}]$ also contains a binary operation d and a ternary operation p satisfying*

$$p(x,y,y) = x, p(x,x,y) = d(x,y), d(x,d(x,y)) = d(x,y),$$

and a k-ary operation s, satisfying $s(x,y,y,y,\ldots,y,y) = d(y,x)$ and for each $i \in \{2,\ldots,k\}$, $s(y,y,\ldots,y,x,y,\ldots,y) = y$, where x appears in position i.

If e is a k-edge operation over D and d the operation in Theorem 19 then $(a,b) \in D^2$ is a *minority pair* if $d(a,b) = b$. Given an n-ary relation $R \in \mathrm{Inv}(\{\mathrm{e}\})$ and $t, t' \in R$ we then say that the index $(i,a,b) \in \{1,\ldots,n\} \times D^2$ *witnesses* (t,t') if (a,b) is a minority pair and $\mathrm{pr}_{1,\ldots,i-1}(t) = \mathrm{pr}_{1,\ldots,i-1}(t')$ and $t[i] = a$, $t'[i] = b$. We let $\mathrm{Sig}_e(R)$ denote the set of all indexes witnessing tuples of the relation $R \in \mathrm{Inv}(\{\mathrm{e}\})$. Last, $R' \subseteq R$ is a *representation* of R if (1) $\mathrm{Sig}_e(R) = \mathrm{Sig}_e(R')$ and (2) for every $i_1,\ldots,i_{k'} \in \{1,\ldots,n\}$, $k' < k$, $\mathrm{pr}_{i_1,\ldots,i_{k'}}(R) = \mathrm{pr}_{i_1,\ldots,i_{k'}}(R')$. Similar to the Maltsev case we have the following useful property of representations of relations invariant under k-edge operations.

Theorem 20. *[2] Let e be a k-edge operation over a finite domain, $R \in \mathrm{Inv}(\{\mathrm{e}\})$ a relation, and R' a representation of R. Then $\langle R' \rangle_e = R$.*

Moreover, each n-ary relation invariant under a k-edge operation has a compact representation of size $O(n^{k-1})$. By this stage it should not come as a surprise to the reader that Maltsev algorithm outlined in Section 3.1 can be modified to solve CSP($\mathrm{Inv}(\{\mathrm{e}\})$) in polynomial time. We will refer to this algorithm as the *few subpowers* algorithm [16]. We then obtain analogous to the Maltsev case from Section 3.2.

Theorem 21. *Let Γ be a Boolean constraint language which admits a polynomially bounded k-edge embedding $\hat{\Gamma}$ over a finite domain D. Then SAT(Γ) has a kernel with $O(|D|^{k-1}|V|^{k-1})$ constraints.*

Proof. We only provide a proof sketch since the details are very similar to the Maltsev case. Assume $k \geq 3$, since otherwise the bound follows from Theorem 13 and 17. Given an instance $I = (V, \{C_1,\ldots,C_m\})$, iteratively compute compact representations $S_0, S_1, \ldots, S_m$ of the solution space of $(V,\emptyset)$, $(V,\{C_1\})$, …, $(V,\{C_1,\ldots,C_m\})$. This can be done in polynomial time using the procedures from the few subpowers algorithm [16]. We then remove the constraint C_i if and only if $\langle S_i \rangle_e = \langle S_{i-1} \rangle_e$. All that remains to be proven is therefore that $\{\langle S_0 \rangle_e, \langle S_1 \rangle_e, \ldots, \langle S_m \rangle_e\}$ is bounded by $O(|D|^{k-1}|V|^{k-1})$. For each S_i define

$$\mathrm{Proj}(S_i) = \{(I,J) \mid I \in \{1,\ldots,|V|\}^i, J \in D^i, i < k, \mathrm{pr}_I(S_i) = J\}.$$

If $\langle S_i \rangle_e \supset \langle S_{i-1} \rangle_e$ then either $\mathrm{Sig}_e(S_i) \supset \mathrm{Sig}_e(S_{i-1})$ or $\mathrm{Proj}(S_i) \supset \mathrm{Proj}(S_{i-1})$. This gives the bound $1 + |\mathrm{Sig}(D^n)| + |\mathrm{Proj}(D^n)| = O(|D|^{k-1}|V|^{k-1})$. □

4.2 Degree-c Extensions

We now consider an alternative technique for obtaining kernels with $O(n^c)$ constraints, $c > 1$, which is useful for classes of languages that do not admit Maltsev or k-edge embeddings. This will generalise the results on kernelization for constraints defined via non-linear polynomials over finite fields [18].

Definition 22. *We make the following definitions.*

1. *Let $t \in \{0,1\}^r$ be a tuple of arity r and $S_1, \ldots, S_l$ an enumeration of all subsets of $\{1, \ldots, r\}$ of size at most c. A tuple $\check{t} \in \{0,1\}^l$ is a* degree-c *extension of t if $\check{t}[i] = \prod_{j \in S_i} t[j]$, $i \in \{1, \ldots, l\}$.*

2. *A* degree-c extension of Γ *is a language $\check{\Gamma}$ with a bijection h between relations $R \in \Gamma$ and relations $\check{R} \in \check{\Gamma}$ such that for every $R \in \Gamma$ and for every tuple $t \in \{0,1\}^{\mathrm{ar}(R)}$, $t \in R$ if and only if $\check{t} \in \check{R}$ where $\check{t}$ is a degree-c extension of t.*

Let $I = (V, C)$ be a SAT(Γ) instance for a Boolean constraint language Γ. Let the *degree-c extension* of V be the set $V^{(c)}$ consisting of all subsets of V of size at most c, and from any assignment $g : V \to \{0,1\}$ we define an assignment $g' : V^{(c)} \to \{0,1\}$ as $g'(S) := \prod_{v \in S} g(v)$ for every set $S \in V^{(c)}$. Degree-c extensions, Maltsev embeddings and k-edge embeddings are related by the following lemma.

Theorem 23. *Let Γ be a finite Boolean language and $\check{\Gamma}$ a degree-c extension of Γ. If $\check{\Gamma}$ admits a Maltsev embedding, then SAT(Γ) admits a kernel of $O(n^c)$ constraints; if $\check{\Gamma}$ admits a k-edge embedding, then SAT(Γ) admits a kernel of $O(n^{kc})$ constraints.*

Proof. Since Γ is finite and fixed, we skirt all issues about how to compute the extension and the embedding. Let $I = (V, C)$, $|V| = n$, be an instance of SAT(Γ), and let $V^{(c)}$ be the degree-c extension of V. For each constraint $R(x_1, \ldots, x_m)$, $m = \mathrm{ar}(R)$, let $X_1, \ldots, X_l \in V^{(c)}$ denote the subsets of $\{x_1, \ldots, x_m\}$ of size at most c, and replace $R(x_1, \ldots, x_m)$ by the constraint $\check{R}(X_1, \ldots, X_l)$. Let I' be the instance of SAT($\check{\Gamma}$) resulting from repeating this for every constraint in the instance. Observe that if g is a satisfying assignment to I then $g'(X) = \prod_{x \in X} g(x)$, $X \in V^{(c)}$, is a satisfying assignment to I'. We now apply the kernelization for languages with Maltsev embeddings, respectively k-edge embeddings, to I', and let $I'' = (V, C')$ where $C' \subseteq C$ is the set of constraints kept by the kernelization. Note that the contents of the relation Ψ_I defined by I correspond directly to the relation $\{\check{t} \cap \Psi_{I'} \mid t \in \{0,1\}^n\}$. Since the kernelizations we use preserve the entire solution space, this kernelization procedure is sound, and the desired bound for the number of constraints in the output follows. □

We observe that this captures the class of SAT problems which can be written as roots of low-degree polynomials from Jansen and Pieterse [18].

Theorem 24. *Let Γ be a Boolean language such that every relation $R \in \Gamma$ can be defined as the set of solutions in $\{0,1\}$ to a polynomial of degree at most d, over some fixed finite field F. Then Γ admits a degree-d extension with a Maltsev embedding.*

Proof. We give a short sketch of the most important ideas. Let $G_1 = (D, \cdot)$ and $G_2 = (D, +)$ be the two Abelian groups representing the field F. For $R \in \Gamma$, let p_R be the polynomial defining R. Then p_R can be written as a sum of monomials over G_1 of degree at most d, and each of these monomials corresponds to a member of $V^{(d)}$. Hence, the extension $\check{R}$ of R can be written as a linear sum over G_2, and similar to Example 9 it is now clear that the coset generating operation of G_2 will preserve the resulting Maltsev embedding, and the result follows from Theorem 23. □

5 Universal Partial Maltsev Operations and Lower Bounds

We have seen that Maltsev embeddings and, more generally, k-edge embeddings, provide an algebraic criterion for determining that a SAT(Γ) problem admits a kernel of a fixed size. In this section we demonstrate that our approach can also be used to give lower bounds for the kernelization complexity of SAT(Γ). More specifically, we will use the fact that if a satisfiability problem SAT(Γ) admits a Maltsev embedding, then this can be witnessed by certain canonical partial operations preserving Γ. We begin in Section 5.1 by studying properties of these canonical partial operations, and in Section 5.2 prove that the absence of these operations can be used to prove lower bounds on kernelizability.

5.1 Universal Partial Maltsev Operations

Let $f : D^k \to D$ be a k-ary operation over $D \supseteq \{0,1\}$. We can then in a natural way associate a partial Boolean operation $f_{|\mathbb{B}}$ with f by restricting f to the Boolean arguments which also result in a Boolean value. In other words $\mathrm{domain}(f_{|\mathbb{B}}) = \{(x_1,\ldots,x_k) \in \{0,1\}^k \mid f(x_1,\ldots,x_k) \in \{0,1\}\}$, and $f_{|\mathbb{B}}(x_1,\ldots,x_k) = f(x_1,\ldots,x_k)$ for every $(x_1,\ldots,x_k) \in \mathrm{domain}(f_{|\mathbb{B}})$. For Maltsev embeddings we have the following straightforward lemma.

Lemma 25. *Let Γ be a Boolean constraint language admitting a Maltsev embedding $\hat{\Gamma}$. Then $f_{|\mathbb{B}} \in \mathrm{pPol}(\Gamma)$ for every $f \in \mathrm{Pol}(\hat{\Gamma})$.*

Proof. Assume, with the aim of reaching a contradiction, that $f_{|\mathbb{B}}(t_1,\ldots,t_n) \notin R$ for some $R \in \Gamma$ and some n-ary $f \in \mathrm{Pol}(\hat{\Gamma})$. By construction, $f_{|\mathbb{B}}(t_1,\ldots,t_n) = t$ is a Boolean tuple. But since $\hat{R} \cap \{0,1\}^{\mathrm{ar}(R)} = R$, this implies (1) that $t \notin \hat{R}$ and (2) that $f_{|\mathbb{B}}(t_1,\ldots,t_n) = f(t_1,\ldots,t_n) = t \notin \hat{R}$. Hence, f does not preserve $\hat{R}$ or $\hat{\Gamma}$, and we conclude that $f_{|\mathbb{B}} \in \mathrm{pPol}(\Gamma)$. □

A Boolean partial operation f is a *universal partial Maltsev operation* if $f \in \mathrm{pPol}(\Gamma)$ for every Boolean Γ admitting a Maltsev embedding.

Definition 26. *Let the infinite domain D_∞ be recursively defined to contain 0, 1, and ternary tuples of the form (x,y,z) where $x,y,z \in D_\infty$. The ternary Maltsev operation u over D_∞ is defined as $u(x,x,y) = y, u(x,y,y) = x$, and $u(x,y,z) = (x,y,z)$ otherwise.*

In the following theorem we show that if an operation q is included in the clone generated by the operation u, then the partial Maltsev operation $q_{|\mathbb{B}}$ is universal. Before the presenting the proof we need some additional notation. It is well-known that if $[F]$ is a clone over a domain D then $f \in [F]$ if and only if f is definable as a term operation over the algebra (D,F) [15]. Given a term $T(x_1,\ldots,x_n)$ over an algebra (D,F) defining a function $g \in [F]$ and $b_1,\ldots,b_n \in D$, we let $\mathrm{Val}(T(b_1,\ldots,b_n)) = g(b_1,\ldots,b_n)$.

Theorem 27. *Let $q \in [\{u\}]$. Then $q_{|\mathbb{B}}$ is a universal partial Maltsev operation.*

Proof. Let Γ be a Boolean constraint language which admits a Maltsev embedding $\hat{\Gamma}$. We will prove that $q_{|\mathbb{B}} \in \mathrm{pPol}(\Gamma)$. Let p be the Maltsev operation witnessing the embedding $\hat{\Gamma}$, let n denote the arity of q, and let $q(x_1,\ldots,x_n) = T^u(x_1,\ldots,x_n)$ where T^u is the term over u defining q. Now, first consider the operation $q' \in [\{p\}]$ obtained by replacing each occurence of u with p in the term $T^u(x_1,\ldots,x_n)$. Let $T^p(x_1,\ldots,x_n)$ denote this term over p, and for each term $T^u_i(\mathbf{x_i})$ occurring as a subterm in $T^u(x_1,\ldots,x_n)$ we let $T^p_i(\mathbf{x_i})$ denote the corresponding term over p.

Now observe that the partial operation $q'_{|\mathbb{B}}$ is included in $\mathrm{pPol}(\Gamma)$ via Lemma 25. We claim that $q_{|\mathbb{B}}$ can be obtained as a subfunction of $q'_{|\mathbb{B}}$, which implies that $q_{|\mathbb{B}} \in \mathrm{pPol}(\Gamma)$, since a strong partial clone is always closed under taking subfunctions. By definition, we have that $(b_1,\ldots,b_n) \in \mathrm{domain}(q_{|\mathbb{B}})$ if and only if $b_1,\ldots,b_n \in \{0,1\}$ and $q(b_1,\ldots,b_n) \in \{0,1\}$.

We will prove that for each sequence of Boolean arguments $b_1,\ldots,b_n$, if $q(b_1,\ldots,b_n) = b \in \{0,1\}$ then $q'(b_1,\ldots,b_n) = b$. First, let us illustrate the intuition behind this by an example. Assume that $n = 7$ and that $T^u(x_1,x_2,x_3,x_4,x_5,x_6,x_7) = u(u(x_1,x_2,x_3),u(x_4,x_5,x_6),x_7)$. In this case we will e.g. have that $\mathrm{Val}(T^u(0,1,0,0,1,0,1)) = 1$ since $u(u(0,1,0),u(0,1,0),1) = u((0,1,0),(0,1,0),1) = 1$, due to the fact that u always respect the Maltsev identities. But since p is also a Maltsev operation it must also be the case that $\mathrm{Val}(T^p(0,1,0,0,1,0,1)) = 1$, even if $u(0,1,0)$ and $p(0,1,0)$ might differ.

We will now prove the general case by a case inspection of the term T^u. First, assume that T^u contains a term of the form $u(x_{i_1},x_{i_2},x_{i_3})$. If $b_{i_1},b_{i_2},b_{i_3} \in \{0,1\}$ then $u(b_{i_1},b_{i_2},b_{i_3}) \in \{0,1\}$ if and only if $b_{i_1} = b_{i_2}$ or $b_{i_2} = b_{i_3}$. But this implies that $p(b_{i_1},b_{i_2},b_{i_3}) = u(b_{i_1},b_{i_2},b_{i_3})$ since p is Maltsev. Second, assume that T^u contains a term of the form $u(T^u_1(\mathbf{x_1}),T^u_2(\mathbf{x_2}),T^u_3(\mathbf{x_3}))$ where $\mathbf{x_1},\mathbf{x_2}$ and $\mathbf{x_3}$ are tuples of variables over $x_1,\ldots,x_n$. Let $\mathbf{b_1}$, $\mathbf{b_2}$ and $\mathbf{b_3}$ be Boolean tuples matching the length of $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{x_3}$, and assume that $\mathrm{Val}(T^u_1(\mathbf{b_1})) = \mathrm{Val}(T^p_1(\mathbf{b_1}))$, $\mathrm{Val}(T^u_2(\mathbf{b_2})) = \mathrm{Val}(T^p_2(\mathbf{b_2}))$ and $\mathrm{Val}(T^u_3(\mathbf{b_3})) = \mathrm{Val}(T^p_3(\mathbf{b_3}))$. Similarly to the first case we have that

$$u(\mathrm{Val}(T^u_1(\mathbf{b_1})),\mathrm{Val}(T^u_2(\mathbf{b_2})),\mathrm{Val}(T^u_3(\mathbf{b_3}))) \in \{0,1\}$$

if and only if $\mathrm{Val}(T_1^u(\mathbf{b_1})) = \mathrm{Val}(T_2^u(\mathbf{b_2}))$ or $\mathrm{Val}(T_2^u(\mathbf{b_2})) = \mathrm{Val}(T_3^u(\mathbf{b_3}))$, and since p is Maltsev this implies that

$$p(\mathrm{Val}(T_1^p(\mathbf{b_1})), \mathrm{Val}(T_2^p(\mathbf{b_2})), \mathrm{Val}(T_3^p(\mathbf{b_3}))) = u(\mathrm{Val}(T_1^u(\mathbf{b_1})), \mathrm{Val}(T_2^u(\mathbf{b_2})), \mathrm{Val}(T_3^u(\mathbf{b_3}))).$$

Hence, for each $(b_1, \ldots, b_n) \in \mathrm{domain}(q_{|\mathbb{B}})$ we have that $(b_1, \ldots, b_n) \in \mathrm{domain}(q'_{|\mathbb{B}})$ and that $q_{|\mathbb{B}}(b_1, \ldots, b_n) = q'_{|\mathbb{B}}(b_1, \ldots, b_n)$. This implies that $q_{|\mathbb{B}}$ is a subfunction of $q'_{|\mathbb{B}}$, that $q_{|\mathbb{B}} \in \mathrm{pPol}(\Gamma)$, and, finally, that $q_{|\mathbb{B}}$ is a universal partial Maltsev operation. □

Using Theorem 27 we can now prove that every Boolean language Γ invariant under the universal partial Maltsev operations admits a Maltsev embedding over D_∞.

Theorem 28. *Let Γ be a Boolean constraint language. Then $\mathrm{pPol}(\Gamma)$ contains all universal partial Maltsev operations if and only if Γ has a Maltsev embedding $\hat{\Gamma}$ over D_∞.*

Proof. For the first direction, let u be the Maltsev operation from Definition 26 over the infinite domain D_∞. For each relation $R \in \Gamma$ we let $\hat{R} = \langle R \rangle_u$. Let $\hat{\Gamma}$ denote the resulting constraint language over D_∞. By definition, $u \in \mathrm{Pol}(\hat{\Gamma})$, and everything that remains to be proven is that $\hat{R} \cap \{0,1\}^{\mathrm{ar}(R)} = R$ for each $\hat{R} \in \hat{\Gamma}$. Hence, assume that there exists at least one tuple $t \in (\hat{R} \cap \{0,1\}^{\mathrm{ar}(R)}) \setminus R$. This implies that there exists a term T over u such that $\mathrm{Val}(T(t_1[i], \ldots, t_m[i])) = t[i]$ for each $i \in \{1, \ldots, \mathrm{ar}(R)\}$, where $R = \{t_1, \ldots, t_m\}$. Let q denote the function corresponding to the term T and observe that $q \in [\{u\}]$. According to Theorem 27 this implies that $q_{|\mathbb{B}}$ is a universal partial Maltsev operation and, furthermore, that $q_{|\mathbb{B}}(t_1[i], \ldots, t_m[i])$ is defined for each $i \in \{1, \ldots, \mathrm{ar}(R)\}$, since $q(t_1[i], \ldots, t_m[i]) \in \{0,1\}$. Hence, $q_{|\mathbb{B}}(t_1, \ldots, t_m) = t \notin R$, which contradicts the assumption that Γ was invariant under all universal partial Maltsev operations.

The second direction is trivial since if Γ has a Maltsev embedding over D_∞ then Γ by definition is preserved by every universal partial Maltsev operation. □

It is worth remarking that Theorem 27 and Theorem 28 implies that every universal partial Maltsev operation can be described via Theorem 27.

5.2 Lower Bounds

Define the *first partial Maltsev operation* ϕ_1 as $\phi_1(x, y, y) = x$ and $\phi_1(x, x, y) = y$ for all $x, y \in \{0,1\}$, and observe that $\mathrm{domain}(\phi_1) = \{(0,0,0), (1,1,1), (0,0,1), (1,1,0), (1,0,0), (0,1,1)\}$. Via Theorem 27 it follows that ϕ_1 is equivalent to $u_{|\mathbb{B}}$, and is therefore a universal partial Maltsev operation. In this section we will prove that $\phi_1 \in \mathrm{pPol}(\Gamma)$ is in fact a necessary condition for the existence of a linear-sized kernel for SAT(Γ), modulo a standard complexity theoretical assumption. A pivotal part of this proof is that if $\phi_1 \notin \mathrm{pPol}(\Gamma)$, then Γ can qfpp-define a relation Φ_1, which can be used as a gadget in a reduction from the Vertex Cover problem. This relation is defined as $\Phi_1(x_1, x_2, x_3, x_4, x_5, x_6) \equiv (x_1 \vee x_4) \wedge (x_1 \neq x_3) \wedge (x_2 \neq x_4) \wedge (x_5 = 0) \wedge (x_6 = 1)$. Note that the values enumerated by the arguments of Φ_1 is in a one-to-one correspondance with $\mathrm{domain}(\phi_1)$. However, as made clear in the following lemma, there is an even stronger relationship between ϕ_1 and Φ_1.

Lemma 29. *If Γ is a Boolean constraint language such that $\langle \Gamma \rangle = BR$ and $\phi_1 \notin \mathrm{pPol}(\Gamma)$ then $\Phi_1 \in \langle \Gamma \rangle_{\not\exists}$.*

Proof. Before the proof we need two central observations. First, the assumption that $\langle \Gamma \rangle = BR$ is well-known to be equivalent to that $\mathrm{Pol}(\Gamma)$ consists only of projections [5]. Second, Φ_1 consists of three tuples which can be ordered as s_1, s_2, s_3 in such a way that there for every $s \in \mathrm{domain}(\phi_1)$ exists $1 \leq i \leq 6$ such that $s = (s_1[i], s_2[i], s_3[i])$. Now, assume that $\langle \Gamma \rangle = BR$, $\phi_1 \notin \mathrm{pPol}(\Gamma)$, but that $\Phi_1 \notin \langle \Gamma \rangle_{\not\exists}$. Then there exists an n-ary partial operation $f \in \mathrm{pPol}(\Gamma)$ such that $f \notin \mathrm{pPol}(\{\Phi_1\})$, and $t_1, \ldots, t_n \in \Phi_1$ such that $f(t_1, \ldots, t_n) \notin \Phi_1$. Now consider the value $k = |\{t_1, \ldots, t_n\}|$, i.e., the number of distinct tuples in the sequence. If $n > k$ then it is known that there exists a closely related partial operation g of arity at most k such that $g \notin \mathrm{pPol}(\{\Phi_1\})$ [27], and we may therefore assume that $n = k \leq |\Phi_1| = 3$. Assume first that $1 \leq n \leq 2$. It is then not difficult to see that there for every $t \in \{0,1\}^n$ exists i such that $(t_1[i], \ldots, t_n[i]) = t$. But then it follows that f is in fact a total

operation which is not a projection, which is impossible since we assumed that $\langle\Gamma\rangle = BR$. Hence, it must be the case that $n = 3$, and that $\{t_1, t_2, t_3\} = \Phi_1$. Assume without loss of generality that $t_1 = s_1$, $t_2 = s_2$, $t_3 = s_3$, and note that this implies that $\mathrm{domain}(f) = \mathrm{domain}(\phi_1)$ (otherwise the arguments of f can be described as a permutation of the arguments of ϕ_1). First, we will show that $f(0,0,0) = 0$ and that $f(1,1,1) = 1$. Indeed, if $f(0,0,0) = 1$ or $f(1,1,1) = 0$, it is possible to define a unary total operation f' as $f'(x) = f(x,x,x)$ which is not a projection since either $f'(0) = 1$ or $f'(1) = 0$. Second, assume there exists $(x,y,z) \in \mathrm{domain}(f)$, distinct from $(0,0,0)$ and $(1,1,1)$, such that $f(x,y,z) \neq \phi_1(x,y,z)$. Without loss of generality assume that $(x,y,z) = (a,a,b)$ for $a, b \in \{0,1\}$, and note that $f(a,a,b) = a$ since $\phi_1(a,a,b) = b$. If also $f(b,b,a) = a$ it is possible to define a binary total operation $f'(x,y) = f(x,x,y)$ which is not a projection, therefore we have that $f(b,b,a) = b$. We next consider the values taken by f on the tuples (b,a,a) and (a,b,b). If $f(b,a,a) = f(a,b,b)$ then we can again define a total, binary operation which is not a projection, therefore it must hold that $f(b,a,a) \neq f(a,b,b)$. However, regardless of whether $f(b,a,a) = b$ or $f(b,a,a) = a$, it is not difficult to verify that f must be a partial projection. This contradicts the assumption that $f \notin \mathrm{pPol}(\{\Phi_1\})$, and we conclude that $\Phi_1 \in \langle\Gamma\rangle_{\not\equiv}$. □

We will now use Lemma 29 to give a reduction from the Vertex Cover problem. It is known that Vertex Cover does not admit a kernel with $O(n^{2-\varepsilon})$ edges for any $\varepsilon > 0$, unless NP $\subseteq$ co-NP/poly [12]. For each n and k let $H_{n,k}$ denote the relation $\{(b_1, \ldots, b_n) \in \{0,1\}^n \mid b_1 + \ldots + b_n = k\}$.

Lemma 30. *Let Γ be a constraint language such that $\langle\Gamma\rangle = BR$. Then Γ can pp-define $H_{n,k}$ with $O(n+k)$ constraints and $O(n+k)$ existentially quantified variables.*

Proof. We first observe that one can recursively design a circuit consisting of fan-in 2 gates which computes the sum of n input gates as follows. At the lowest level, we split the input gates into pairs and compute the sum for each pair, producing an output of 2 bits for each pair. This can clearly be done with $O(1)$ gates. At every level i above that, we join each pair of outputs from the previous level, of i bits each, into a single output of $i+1$ bits which computes their sum. This can be done with $O(i)$ gates by chaining full adders. Finally, at level $\lceil \log_2 n \rceil$, we will have computed the sum. The total number of gates will be

$$\sum_{i=1}^{\lceil \log_2 n \rceil} (\frac{n}{2^i}) \cdot O(i),$$

and it is a straightforward exercise to show that this sums to $O(n)$. Let $z_1, \ldots, z_{\log_2 n}$ denote the output gates of this circuit. By a standard Tseytin transformation we then obtain an equisatisfiable 3-SAT instance with $O(n)$ clauses and $O(n)$ variables [33]. Next, for each $1 \leq i \leq \log_2 n$, add the unary constraint $(z_i = k_i)$, where k_i denotes the ith bit of k written in binary. Each such unary constraint can clearly be pp-defined with $O(1)$ existentially quantified variables over Γ. We then pp-define each 3-SAT clause in order to obtain a pp-definition of R over Γ, which in total only requires $O(n)$ existentially quantified variables. Note that this can be done since we assumed that $\langle\Gamma\rangle = BR$ which implies that Γ can pp-define every Boolean relation. □

Theorem 31. *Let Γ be a finite Boolean constraint language such that $\langle\Gamma\rangle = BR$ and $\phi \notin \mathrm{pPol}(\Gamma)$. Then SAT($\Gamma$) does not have a kernel of size $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless NP $\subseteq$ co-NP/poly.*

Proof. We will give a polynomial-time many-one reduction from Vertex Cover parameterized by the number of vertices to SAT($\Gamma \cup \{\Phi_1\}$), which via Theorem 6 and Lemma 29 has a reduction to SAT(Γ) which does not increase the number of variables. Let (V, E) be the input graph and let k denote the maximum size of the cover. First, introduce two fresh variables x_v and x'_v for each $v \in V$, and one variable y_i for each $1 \leq i \leq k$. Furthermore, introduce two variables x and y. For each edge $\{u, v\} \in E$ introduce a constraint $\Phi_1(x_u, x'_v, x'_u, x_v, x, y)$, and note that this enforces the constraint $(x_u \vee x_v)$. Let $\exists z_1, \ldots, z_m.\phi(x_1, \ldots, x_{|V|}, y_1, \ldots, y_k, z_1, \ldots, z_m)$ denote the pp-definition $H_{|V|+k,k}$ over Γ where $m \in O(k + |V|)$, and consisting of at most $O(k + |V|)$ constraints. Such a pp-definition must exist according to Lemma 30. Drop the existential quantifiers and add the constraints of $\phi(x_1, \ldots, x_{|V|}, y_1, \ldots, y_k, z_1, \ldots, z_m)$. Let (V', C) denote this instance of SAT($\Gamma \cup \{\Phi_1\}$). Assume

first that (V, E) has a vertex cover of size $k' \leq k$. We first assign x the value 0 and y the value 1. For each v in this cover assign x_v the value 1 and x'_v the value 0. For any vertex not included in the cover we use the opposite values. We then set $y_1, \ldots, y_{k-k'}$ to 1, and $y_{k-k'+1}, \ldots, y_k$ to 0. For the other direction, assume that (V', C) is satisfiable. For any x_v variable assigned 1 we then let v be part of the vertex cover. Since $x_1 + \ldots + x_{|V|} + y_1 + \ldots + y_k = k$, the resulting vertex cover is smaller than or equal to k. □

As an example, let $R^k = \{(b_1, \ldots, b_k) \in \{0,1\}^k \mid b_1 + \ldots + b_k \in \{1,2\} \pmod 6\}$ and let $P = \{R^k \mid k \geq 1\}$. The kernelization status of SAT(P) was left open in Jansen and Pieterse [18], and while a precise upper bound seems difficult to obtain, we can at least prove that this problem does not admit a kernel of linear size, unless NP $\subseteq$ co-NP/poly. To see this, observe that $(0,0,1), (0,1,1), (0,1,0) \in R^3$ but $\phi_1((0,0,1), (0,1,1), (0,1,0)) = (0,0,0) \notin R^3$. The result then follows from Theorem 31.

At this stage, it might be tempting to conjecture that $\phi_1 \in \mathrm{pPol}(\Gamma)$ is also a sufficient condition for a Maltsev embedding, and, more ambitiously, that this is also a sufficient condition for a kernel with $O(n)$ constraints. We can immediately rule out the first possibility by exhibiting a relation R and a universal partial Maltsev operation ϕ such that R is invariant under ϕ_1 but not under ϕ. For example, first define the operation ϕ_2 as $q_{|\mathbb{B}}$ where $q(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = u(u(x_1, x_2, x_3), u(x_4, x_5, x_6), u(x_7, x_8, x_9))$. Second, consider the relation R of arity $|\mathrm{domain}(\phi_2)|$ consisting of 9 tuples $t_1, \ldots, t_9$ such that there for each $t \in \mathrm{domain}(\phi_2)$ exists exactly one $1 \leq i \leq |\mathrm{domain}(\phi_2)|$ such that $(t_1[i], \ldots, t_9[i]) = t$. Then, by definition, $\phi_2(t_1, \ldots, t_9) \notin R$ (since ϕ_2 is not a partial projejction), and hence does not preserve R, but one can verify that this relation is preserved by ϕ_1. To rule out the second possibility we have to find a constraint language Γ such that ϕ_1 preserves Γ but SAT(Γ) does not have a kernel with $O(n)$ constraints. In fact, we will prove a stronger result, and show that whenever P is a finite set of partial operations such that $\langle \mathrm{Inv(P)} \rangle = BR$ (and thus, SAT(Inv(P)) is NP-complete), then SAT(Inv(P)) does not admit a polynomial kernel, unless NP $\subseteq$ co-NP/poly. This is in contrast to the existing parameterized dichotomy results for CSP [9, 23, 24, 25, 28], but as noted in the introduction, it is an expected conclusion when the parameter is n; cf. [26, Lemma 35].

Theorem 32. *Let P be a finite set of partial polymorphisms such that $\langle \mathrm{Inv(P)} \rangle = BR$. Then SAT(Inv(P)) does not admit a polynomial kernel unless NP $\subseteq$ co-NP/poly.*

Proof. We will show a reduction from k-SAT on n variables to SAT(Inv(P)) on $O(n^c)$ variables for some absolute constant c that only depends on P. Since k-SAT admits no kernel of size $O(n^{k-\varepsilon})$ for any $\varepsilon > 0$ unless NP $\subseteq$ co-NP/poly [12], and since c is independent of k, the result will follow.

The strategy to show the reduction is similar to that of Lemma 35 of [26]). Let c be the largest arity of a partial polymorphism in P, and let (X, C) be an instance of k-SAT, $|X| = n$. Create a set of padding variables $Y = \{y_{\bar{x},f} \mid \bar{x} \in X^d, f : \{0,1\}^d \to \{0,1\}\}$, one for every $(x_1, \ldots, x_d) \in X^d$ and every d-ary function, $d = c^2$. We will constrain so that for every $y_{\bar{x},f} \in Y$ and every satisfying assignment, we have $y_{\bar{x},f} = f(\bar{x}(1), \ldots, \bar{x}(d))$. The following is a central observation.

Claim 33. *Let $R \subseteq \{0,1\}^r$ be any r-ary Boolean relation, and let $X = \{x_1, \ldots, x_r\}$. Let Y be a set of padding variables for X as above. Define a relation R' by*

$$R'(X, Y) \equiv R(X) \wedge \bigwedge_{y_{\bar{x},f} \in Y} (y_{\bar{x},f} = f(\bar{x}(1), \ldots, \bar{x}(c))).$$

Then $R(X) \equiv \exists Y R'(X, Y)$ and R' is invariant under every non-total partial operation of arity at most c.

Proof. Let ϕ be a non-total partial operation of arity $c' \leq c$, and assume that R' is not invariant under ϕ, i.e., there are tuples $t_1, \ldots, t_{c'} \in R'$ such that $\phi(t_1, \ldots, t_{c'})$ is defined and not contained in R'. We assume that all tuples t_i are distinct, as otherwise the application $\phi(t_1, \ldots, t_c)$ defines an operation ϕ' of arity $|\{t_1, \ldots, t_{c'}\}|$ for which we can repeat the argument below. Let $u_1, \ldots, u_{c'}$ be the projections of the tuples onto X. Note that the tuples $u_1, \ldots, u_{c'}$ are distinct, and that $\phi(u_1, \ldots, u_{c'})$ is defined. Let $I \subseteq [r]$ be a minimal set of "witness positions" for the distinctness of u, i.e., for every pair $i, j \in [c']$, $i \neq j$, there is a position $p \in I$ such that $u_i[p] \neq u_j[p]$. Note that $|I| \leq c^2$. Let $t \in \{0,1\}^{c'}$ be a tuple for which ϕ is undefined. Then there exists a function $f : \{0,1\}^{|I|} \to \{0,1\}$ such that $f(\mathrm{pr}_I u_i) = t[i]$ for each $i \in [c']$, since the projection onto I is distinct for all tuples u_i, and since $|I| \leq d^2$, there exist a

variable $y_{\bar{x},f}$ in Y. This implies that $\phi(t_1, \ldots, t_c)$ is undefined, since in particular ϕ is undefined when applied to the position corresponding to $y_{\bar{x},f}$. Since ϕ was generically chosen, the claim follows. □

We can now wrap up the proof as follows. By Theorem 5, the language $\mathrm{Inv}(\mathrm{P})$ has a quantifier-free pp-definition of any relation R such that R is invariant under any partial operation in P. By the above claim, this in particular includes the padded relation R' for any given relation R. Now, if (X, C) is an instance of k-SAT, with $|X| = n$ as above, and if Y is the set of padding variables, then we can output an instance of SAT($\mathrm{Inv}(\mathrm{P})$) by replacing every k-clause in the input, defining a relation $R(V)$ for some $V \subseteq X$, by the relation $R'(V, Y_V)$ according to the above claim. Note in particular that the padding variables Y_V used in this reduction can be chosen from the set X_V. Finally, since k is constant, the relations $R'(V, Y_V)$ and hence the output can be enumerated in polynomial time. □

6 Concluding Remarks and Future Research

We have studied the kernelization properties of SAT and CSP problems parameterized by the number of variables with tools from universal algebra. We particularly focused on problems with linear kernels, and showed that a CSP problem has a kernel with $O(n)$ constraints if it can be embedded into a CSP problem preserved by a Maltsev operation; thus extending previous results in this direction. On the other hand, we showed that a SAT problem not preserved by a partial Maltsev polymorphism does not admit such a kernel, unless NP $\subseteq$ co-NP/poly. This shows that the algebraic approach is viable for studying such fine-grained kernelizability questions. More generally, we also gave algebraic conditions for the existence of a kernel with $O(n^c)$ constraints, $c > 1$, generalising previous results on kernels for SAT problems defined via low-degree polynomials over a finite field. Our work opens several directions for future research.

A dichotomy theorem for linear kernels? Our results suggest a possible dichotomy theorem for the existence of linear kernels for SAT problems, at least for finite languages. However, two gaps remain towards such a result. On the one hand, we have proven that any language Γ preserved by the universal partial Maltsev operations admits a Maltsev embedding over an infinite domain. However, the kernelization algorithms only work for languages with Maltsev embeddings over finite domains. Does the existence of an infinite-domain Maltsev embedding for a finite language imply the existence of a Maltsev embedding over a finite domain? Alternatively, can the algorithms be adjusted to work also for languages with infinite domains, given that this domain is finitely generated in a simple way? On the other hand, we only have necessity results for the first partial operation ϕ_1 out of an infinite set of conditions for the positive results. Is it true that every universal partial Maltsev operation is a partial polymorphism of every language with a linear kernel, or do there exist SAT problems with linear kernels that do not admit Maltsev embeddings?

Cases of higher degree. Compared to the case of linear kernels, our results on kernels with $O(n^c)$ constraints, $c > 1$, are more partial. Does the combination of degree extensions and k-edge embeddings cover all cases, or are there further SAT problems with non-trivial polynomial kernel bounds to be found?

The Algebraic CSP Dichotomy Conjecture. Several solutions to the CSP dichotomy conjecture have been announced [6, 30, 34]. If correct, these algorithms solve CSP(Γ) in polynomial time whenever Γ is preserved by a *Taylor term*. One can then define the concept of a Taylor embedding, which raises the question of whether the proposed algorithms can be modified to construct polynomial kernels. More generally, when can an operation f such that CSP($\mathrm{Inv}(\{\mathrm{f}\})$) is tractable be used to construct improved kernels? On the one hand, one can prove that *k-edge operations*, which are generalized Maltsev operations, can be used to construct kernels with $O(n^{k-1})$ constraints via a variant of the *few subpowers algorithm*. On the other hand, it is known that relations invariant under *semilattice operations* can be described as generalized Horn formulas, but it is not evident how this property could be useful in a kernelization procedure.

References

[1] L. Barto. Constraint satisfaction problem and universal algebra. *ACM SIGLOG News*, 1(2):14–24, October 2014.

[2] J. Berman, P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Varieties with few subalgebras of powers. *Transactions of the American Mathematical Society*, 362(3):1445–1473, 2010.

[3] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. I. *Cybernetics*, 5:243–252, 1969.

[4] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. II. *Cybernetics*, 5:531–539, 1969.

[5] E. Böhler, H. Schnoor, S. Reith, and H. Vollmer. Bases for Boolean co-clones. *Information Processing Letters*, 96(2):59–66, 2005.

[6] A. Bulatov. A dichotomy theorem for nonuniform CSPs. *CoRR*, abs/1703.03021, 2017.

[7] A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SICOMP*, 36(1):16–27, 2006.

[8] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the complexity of constraints using finite algebras. *SICOMP*, 34(3):720–742, March 2005.

[9] A. Bulatov and D. Marx. Constraint satisfaction parameterized by solution size. *SIAM Journal on Computing*, 43(2):573–616, 2014.

[10] N. Creignou and H. Vollmer. Boolean constraint satisfaction problems: When does Post's lattice help? In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 3–37. Springer Berlin Heidelberg, 2008.

[11] V. Dalmau and P. Jeavons. Learnability of quantified formulas. *TCS*, 306(1–3):485 – 511, 2003.

[12] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

[13] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SICOMP*, 28(1):57–104, 1998.

[14] D. Geiger. Closed systems of functions and predicates. *Pac. J. Math.*, 27(1):95–100, 1968.

[15] M. Goldstern and M. Pinsker. A survey of clones on infinite sets. *Algebra universalis*, 59(3):365–403, 2008.

[16] P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing*, 39(7):3023–3037, June 2010.

[17] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 2001.

[18] B. M. P. Jansen and A. Pieterse. Optimal sparsification for some binary CSPs using low-degree polynomials. In *Proceedings of MFCS 2016*, volume 58, pages 71:1–71:14.

[19] B. M. P. Jansen and A. Pieterse. Sparsification upper and lower bounds for graphs problems and not-all-equal SAT. In *Proceedings of IPEC 2015, Patras, Greece*.

[20] P. Jeavons. On the algebraic structure of combinatorial problems. *TCS*, 200:185–204, 1998.

[21] P. Jeavons, D. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings of CP 1995*, pages 276–291.

[22] P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Strong partial clones and the time complexity of SAT problems. *JCSS*, 84:52 – 78, 2017.

[23] S. Kratsch, D. Marx, and M. Wahlström. Parameterized complexity and kernelizability of max ones and exact ones problems. *TOCT*, 8(1):1, 2016.

[24] S. Kratsch and M. Wahlström. Preprocessing of min ones problems: A dichotomy. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 653–665. Springer, 2010.

[25] A. A. Krokhin and D. Marx. On the hardness of losing weight. *ACM Trans. Algorithms*, 8(2):19, 2012.

[26] V. Lagerkvist and M. Wahlström. The power of primitive positive definitions with polynomially many variables. *JLC*, 2016.

[27] V. Lagerkvist, M. Wahlström, and B. Zanuttini. Bounded bases of strong partial clones. In *Proceedings of the ISMVL 2015*.

[28] D. Marx. Parameterized complexity of constraint satisfaction problems. *Comput. Complexity*, 14(2):153–183, 2005.

[29] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Math. Programming*, 8(1):232–248, 1975.

[30] A. Rafiey, J. Kinne, and T. Feder. Dichotomy for digraph homomorphism problems. *CoRR*, abs/1701.02409, 2017.

[31] B.A. Romov. The algebras of partial functions and their invariants. *Cybernetics*, 17(2):157–167, 1981.

[32] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory Of Computing (STOC-78)*, pages 216–226. ACM Press, 1978.

[33] G. S. Tseitin. *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, chapter On the Complexity of Derivation in Propositional Calculus, pages 466–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.

[34] D. Zhuk. The proof of CSP dichotomy conjecture. *CoRR*, abs/1704.01914, 2017.