The Complexity of Symmetry-Breaking Formulas

Eugene M. Luks*
Department of Computer Science
University of Oregon
Eugene, OR 97403
luks@cs.uoregon.edu

Amitabha Roy*
Department of Computer Science
Boston College
Chestnut Hill, MA 02167
aroy@cs.bc.edu

Keywords: symmetry-breaking, lex-leader formulas, symmetry in search.

Abstract

Symmetry-breaking formulas for a constraint-satisfaction problem are satisfied by exactly one member (e.g., the lexicographic leader) from each set of "symmetrical points" in the search space. Thus, the incorporation of such formulas can accelerate the search for a solution without sacrificing satisfiability. We study the computational complexity of generating lex-leader formulas. We show, even for abelian symmetry groups, that the number of essential clauses in the "natural" lex-leader formula could be exponential. Furthermore, we show the intractability (NP-hardness) of finding any expression of lex-leadership without reordering the variables, even for elementary abelian groups with orbits of size 3. Nevertheless, using techniques of computational group theory, we describe a reordering relative to which we construct small lex-leader formulas for abelian groups.

1 Introduction

The exploitation of symmetries to facilitate search is a standard tool in combinatorics (see, e.g., [17, 5, 12]) and has, more recently, been applied with some success to constraint-satisfaction problems [4, 7, 3, 1]. In particular, in [7], Crawford and Ginsberg along with the present authors proposed a novel use

*The authors are pleased to acknowledge partial support via NSF grant CCR9820945

of symmetries that coordinates easily with any existing search technology. Introducing the concept of *symmetry-breaking formulas*, they proposed the use of symmetries to restrict the underlying problem. These additional constraints are satisfied by exactly one member ("the lexicographic leader") of each set of symmetric points[1] in the search space. Thus, instead of having to reformulate each advance in search technology, this method can be used as a preprocessor to any constraint solver. The technique was extended and successfully applied to practical planning problems by Joslin and Roy [14].

The symmetry-breaking approach operates as follows. Let T be the input constraint-satisfaction problem and let G be a group of permutations of the variables known to preserve models of T. One constructs a set S of constraints that are satisfied only by the lexical leader in each G-orbit in the search space. In considering the augmented problem $T \wedge S$, the search algorithm will automatically restrict to those lex-leaders, thus pruning the search.

The central theme of this paper is the efficiency of constructing a suitable lex-leader S.

In previous investigations [7], a "natural" lex-leader formula $\Lambda_{\mathrm{nat}}(G)$ (see Section 3 herein) was proposed, namely, a truth assignment X on the variables is a lex-leader if and only if

$$\forall g \in G : X \geq g(X)$$

(where $g(X)$ represents the effect on the truth assignment of the permutation of the variables; a formal definition appears in Section 3). The efficacy of this formula was successfully demonstrated on selected problems involving small groups. Although $\Lambda_{\mathrm{nat}}(G)$ required enumeration of clauses for all group elements, it was pointed out that one expects considerable redundancy amongst these and so the formula might be brought down to manageable size by "pruning" redundant clauses (see examples in Section 3). However, we now describe symmetry groups for which $\Lambda_{\mathrm{nat}}(G)$ cannot be pruned below exponential size (see Section 6.1). Indeed, this can happen even within a rudimentary class of abelian permutation groups operating on the set Ω of variables: groups in the class have orbits of size ≤ 2 and are, thereby, identifiable with vector spaces over the 2-element field.

Our approach now is to avoid group enumeration by substituting a formula that expresses "$g \in G$". This in itself is not difficult. However, we then need to reformulate what seems an instance of nonsatisfiability

$$\neg\exists g : [(g \in G) \wedge (X < g(X))]$$

as an instance of satisfiability. We do this first for the above "vector spaces", exploiting the fact that nonsolvability of a linear system can be expressed as solvability of a "dual" linear system. As a result, we write (in Section 7.4) a lex-leader formula $\Lambda(G)$ of size $O(n^3)$ for such groups ($n = |\Omega|$).

[1] A search space "point" is a string that records truth assignments to the boolean variables

An additional obstacle arises in extending $\Lambda(G)$ to general abelian groups. Via a sharpening of a result of [2] and [7], we show that even if the orbits have size as large as 3, *testing* lex-leadership of points (i.e., strings) is coNP-complete (Section 6.2). Hence, it is unlikely that the property can be captured in a polynomial-size formula. However, it seems that the problem is sensitive to the ordering of the variables. We go on to show that, for any abelian G, one can find an ordering of Ω with respect to which we construct a suitable $\Lambda(G)$ of size $O\left(n^3\mu(\log n)\right)$ (here $\mu(\log n)$ is a lower-order term capturing the cost of arithmetic on numbers with $O(\log n)$ digits).

The construction of succinct $\Lambda(G)$ makes essential use of the duality of subgroups of abelian groups (equivalent to the isomorphism of the group with its character group) (Section 7.1). This device is needed to express membership in a subgroup which is defined initially by its dual subgroup (Section 7.5). It is also critical in generalizing the result for $\mathbb{Z}_2$ and converting the assertion of nonsolvability of a linear system over some $\mathbb{Z}_{p^e}$ (where p is prime) to the assertion of solvability of a linear system (Section 7.2); this is again a necessary translation en route to an interpretation as satisfiability of a boolean formula.

In Section 8, we comment on the extendability of the results to nonabelian groups.

2 Definitions and Notations

For a group G, we write $H \leq G$ to indicate that H is a *subgroup* of G. The group consisting of all permutations of a set Ω, called *the symmetric group on* Ω, is denoted by $\mathrm{Sym}(\Omega)$; a *permutation group* is a subgroup of $\mathrm{Sym}(\Omega)$ for a specified Ω.

We say that G *acts on* Ω if there is a homomorphism $\phi : G \to \mathrm{Sym}(\Omega)$. Suppose G acts on Ω. For $\omega \in \Omega$ and $g \in G$, the image of ω under $\phi(g)$ is denoted by ω^g. The *orbit of* ω *under* G is $\omega^G = \{\omega^g \mid g \in G\}$. The restriction of G on the orbit Δ, denoted by G^Δ, is called an *orbit constituent* of G. The group G is said to be *transitive* on Ω if Ω is itself an orbit of G. The *point stabilizer* of ω is the subgroup $G_\omega = \{g \in G \mid \omega^g = \omega\}$. The *point-wise stabilizer* of $\Delta \subset \Omega$ is $G_{(\Delta)} = \cap_{\delta\in\Delta} G_\delta$. A group G acts *regularly* on Ω if $G_\omega = 1$ for all $\omega \in \Omega$.

We have particular need to deal with permutation groups G for which every orbit has size ≤ 2. Note, in particular, such G is necessarily an *elementary abelian 2-group*, that is, a direct product of cyclic groups of order 2. These groups are characterizable as well as the additive groups of vector spaces over the 2-element field.

We write $G = \langle S \rangle$ to indicate that set $S \subset G$ generates the group G. For computation, permutation groups are input (and output) via generators. Thus, subgroups of $\mathrm{Sym}(\Omega)$ have succinct descriptions since they have generating sets of size $O(|\Omega|)$ [8]; in particular, we may assume that a group is specified in space that is polynomial in $|\Omega|$. We refer to any standard text (e.g., [13]) for basic facts about groups. For permutation groups, we refer to [8]. See [19] for background on polynomial-time computation in permutation groups.

Suppose $\Omega = \{1, 2, \ldots, n\}$ and $G \leq \mathrm{Sym}(\Omega)$ (this will be the usual situation herein). For $0 \leq i \leq n$, let Ω_i denote $\{1, 2, \ldots, i\}$ and $G_i = G_{(\Omega_i)}$. Let 2^Ω denote the set of functions from Ω to $\{0, 1\}$ (equivalently, 2^Ω is the set of all n-bit strings). Then G acts on 2^Ω via $X \mapsto {}^gX$ for $g \in G$, $X \in 2^\Omega$ where $({}^gX)(i) = X(i^g)$.[2] (The orbits of the action of G on 2^Ω will be of particular concern to us.) For any $X \in 2^\Omega$ and $0 \leq i \leq n$, let X_i be the restriction of X to Ω_i (considering X as a string, X_i is an i-tuple consisting of the first i coordinates). There is a natural lexicographic (dictionary) order on 2^Ω: $X < Y$ if $X \neq Y$ and $X(i) < Y(i)$ for the least i such that $X_i \neq Y_i$. The *lex-leader* in an orbit is the lexically *largest* string.

A *propositional variable* can take on two values, true or false (we write 0 for false, 1 for true). Let L be a set of propositional variables. *Literals* are variables in L or negations of variables in L. A *clause* is a disjunction of *distinct* literals in L. A *theory* is a conjunction of clauses. A *truth assignment* for a set of variables L is a function $X : L \to \{0, 1\}$. In the usual way, X extends by the semantics of propositional logic to a function on the set of theories over L and by abuse of notation, we will continue to denote the extended function by X. A truth assignment X of L is called a *model* of a theory T if $X(T) = 1$.

The *propositional satisfiability* problem or SAT is the following decision problem: given a theory, decide whether it has a model. This is a canonical example of an NP-complete problem [10].

Let T be a theory. A sub-collection $\mathcal{S}$ of clauses of T is said to be a *pruning* of T if the conjunction $\bigwedge_{s \in \mathcal{S}} s$, is logically equivalent to T. A particular clause of T is said to be *non-prunable* if it belongs to all prunings of T. A clause A (or a collection of clauses) is said to *prune* a clause B in T if A (or the conjunction of the collection of clauses) logically implies B. We remark that non-prunability is a very stringent requirement on a clause: if *any* subcollection of clauses of T is logically equivalent to T, it *must* include the non-prunable clauses.

3 The "Natural" Lex-Leader Formula

We formalize the notion of lex-leader formulas in the context of a permutation group acting on the set of variables and develop the "natural" formula of this type.

Let $\Omega = \{1, 2, \ldots, n\}$ be an ordered set, equipped with a total order ρ, and let $G \leq \mathrm{Sym}(\Omega)$. Our goal is to write a formula $\Lambda(G, \rho)$ in propositional logic that is true of only one member of each G-orbit in 2^Ω; we may consider that member to be *canonical*. In this paper, we choose the canonical member to be the lexical leader in the orbit, i.e., a function X such that for all $Y \neq X$ in the same orbit, $Y < X$. Thus, a *lex-leader formula* for G is a boolean formula $\Lambda(G, \rho)$ defined over n variables, whose models are lex-leaders in their orbits. If the ordering ρ of Ω is clear from the context (e.g., when an ordering is explicitly

[2] It is natural to write this as a "left action", e.g., we have ${}^{g_1 g_2}X = {}^{g_1}({}^{g_2}A)$, whereas expressing the image of X under g_1 by X^{g_1} would lead to the awkward relation $X^{g_1 g_2} = (X^{g_2})^{g_1}$.

defined or when it is the natural integer total order in {1,2, ... n}) we drop it from the notation and refer to a lex-leader formula as $\Lambda(G)$.

In subsequent sections, we will allow $\Lambda(G, \rho)$ to be defined over a larger set of variables and require that the projection of its models in a fixed set of n coordinates (e.g., the variables of a theory T where G acts as symmetries of T, see Section 4) are lex-leaders in their G-orbits. However, the first formula that comes to mind involves only the given variables.

By the definition of lexicographical order, for any $X, Y \in 2^\Omega$, the assertion $X \geq Y$ is captured in the boolean formula

$$\bigwedge_{1 \leq i \leq n} (X_{i-1} = Y_{i-1} \ \rightarrow \ X(i) \geq Y(i))$$

With this convention, 11 is the lex-leader in the set $\{00, 01, 10, 11\}$. Observe that $X(i) \geq Y(i)$ is just a mnemonic for the boolean expression $Y(i) \rightarrow X(i)$.

We wish to assert that $X \geq {}^gX$, for all $g \in G$. With this in mind, we let $C(g, i)$ denote the formula

$$({}^gX)_{i-1} \ = \ X_{i-1} \ \rightarrow \ X(i) \geq ({}^gX)(i)$$

(the X will be understood in our use of $C(g, i)$). Note that $C(g, i)$ expands to

$$[(X(1) = X(1^g)] \wedge [X(2) = X(2^g)] \wedge \cdots \wedge [X(i{-}1) = X((i{-}1)^g)] \ \rightarrow \ X(i) \geq X(i^g)$$

Thus, we construct the "natural" lex-leader formula, $\Lambda_{\text{nat}}(G)$, where

$$\Lambda_{\text{nat}}(G, \rho) \ = \ \bigwedge_{g \in G} \bigwedge_{i=1}^{n} C(g, i) \tag{1}$$

As before, if the order ρ is clear from the context, we drop it from the notation and refer to the natural lex-leader formula $\Lambda_{\text{nat}}(G)$.

Equation (1) could have duplicate clauses. For example, consider $G = \text{Sym}(\{1, 2, 3\})$. Then $C((1\ 2), 1) = C((1\ 2\ 3), 1) = (X(1) \geq X(2))$ which means that the clause $X(1) \geq X(2)$ appears twice in Equation (1). Notice that the group elements (1 2) and (1 2 3) both belong to the same right coset of G_1.

The above intuition allows us to eliminate duplicate clauses as follows: For each i, we include the clause $C(g, i)$ for just one g in each coset of G mod G_i. This approach can still leave us with $\sum_{i=0}^{n-1} |G/G_{i+1}|$ clauses (which could be of exponential size in general groups). So the question remains: can we prune $\Lambda_{\text{nat}}(G)$ further? In some cases, we can: for example, the clause

$$C((1, 3), 1) = (X(1) \geq X(3))$$

logically implies the clause

$$C((1, 2, 3), 2) = \{(X(1) = X(2)) \rightarrow X(2) \geq X(3)\}$$

so that, in the presence of the former, the latter can be dropped.

Here are some more substantial examples of pruning.

Example (Symmetric Group)

Let $G = \mathrm{Sym}(\Omega)$ where $\Omega = \{1, 2, \ldots, n\}$. Observe that the lex-leaders of 2^Ω under the action of G are those assignments where all 1's appear before all 0's, i.e., these are assignments X such that $X(i) \geq X(i+1)$ for all $1 \leq i \leq n-1$. Thus a lex-leader formula for G is

$$\bigwedge_{1 \leq i \leq n-1} (X(i) \geq X(i+1)) \tag{2}$$

It is easy to see that one can prune $\Lambda_{\mathrm{nat}}(G)$ to Formula (2). Since the formula in Equation (1) involves a conjunction over every group element, $\Lambda_{\mathrm{nat}}(G)$ starts out with at least $n!$ clauses. First observe that $C(g, i)$ is trivial if $i^g \leq i$ (in fact, this remains true regardless of the group). So we need only consider clauses $C(g, i)$ where $i^g > i$. Any such nontrivial clause $C(g, i)$ is pruned by a clause of the form $C(h, i)$ where h is the transposition $(i\ i^g)$. This removes all clauses but those of the form $X(i) \geq X(j)$ for $i < j$. This means that there are $O(n^2)$ clauses in $\Lambda_{\mathrm{nat}}(G)$ after pruning. But we can further prune even further by replacing any 3 clauses of the form $(X(i) \geq X(j)) \wedge (X(j) \geq X(k)) \wedge (X(i) \geq X(k))$ by $(X(i) \geq X(j)) \wedge (X(j) \geq X(k))$. This prunes $\Lambda_{\mathrm{nat}}(G)$ to Formula (2).

Example (Full Vector Space)

Let $G = \langle g_i \mid 1 \leq i \leq n/2 \rangle \leq \mathrm{Sym}(\Omega)$ where $\Omega = \{1, 2, 3, \ldots, n\}$ have orbits $\{2i-1, 2i\}$ where $1 \leq i \leq n/2$, where we assume n is even. Also $(2i-1)^{g_j} = 2i-1$ (which means that $(2i)^{g_j} = 2i$) if $j \neq i$ and $(2j-1)^{g_j} = 2j$ (and $(2j)^{g_j} = 2j-1$). So $G \equiv \mathbb{Z}_2^{n/2}$ where $g \in G \leftrightarrow v_g \in \mathbb{Z}_2^n$ where $v_g(i) = 1$ iff $(2i-1)^g = 2i$. Since $|G| = 2^{n/2}$, $\Lambda_{\mathrm{nat}}(G)$ has exponential size (before pruning). We now show that $\Lambda_{\mathrm{nat}}(G)$ can be pruned to the following formula:

$$\bigwedge_{1 \leq i \leq n/2} (X(2i-1) \geq X(2i)). \tag{3}$$

To see why, consider any $C(g, i)$ for $1 \leq i \leq n$. Observe that $C(g, i)$ is trivial (and can be pruned from $\Lambda_{\mathrm{nat}}(G)$) when i is even. It is also trivial when i is odd and $i^g = i$. So assume i is odd $(= 2j-1)$ and $(2j-1)^g = 2j$. The consequent of $C(g, 2j-1)$ is $X(2j-1) \geq X(2j)$ and so $C(g, 2j-1)$ is pruned by the clause $C(g_j, 2j-1) = (X(2j-1) \geq X(2j))$. Thus clauses of the form $X(2j-1) \geq X(2j)$ for $1 \leq j \leq n/2$, are the only clauses that remain, pruning $\Lambda_{\mathrm{nat}}(G)$ to Formula (3).

Such examples lead one to hope that, even when $\Lambda_{\mathrm{nat}}(G)$ is of exponential size in $|\Omega|$, one could prune it to polynomial size by removing redundant clauses. However, we shall see that this is not the case even for groups with orbits of size 2 (Theorem 5.1).

4 Symmetry-Breaking Formulas

Let T be a theory over an n variable set L. A permutation $g \in \mathrm{Sym}(L)$ is said to be an automorphism (also called a "symmetry") of the theory T if g maps models of T to models and non-models to non-models. The set of all symmetries of a theory is easily seen to form a group: this group is called the "symmetry group" of the theory, denoted by $\mathrm{Aut}(T)$. Our input will be T and a specified *subgroup* G of $\mathrm{Aut}(T)$. The goal of symmetry-breaking is to use the presence of this group to find models of T efficiently.

We remark that this is a slight departure from the methodology of [7] which explicitly computed the group of syntactic symmetries of an input theory T and always used this precise group. A syntactic symmetry is a permutation of the variables that maps the set of clauses to itself.

In this paper, we make no assumptions on how we obtain the input group G. The group G could possibly include symmetries that are not syntactic; for example, G could contain permutations that the user knows are symmetries because of some domain-specific knowledge. On the other hand, syntactic symmetries can reveal hidden structure in the input problem: e.g., in [14], where the authors considered transportation planning problems, structural symmetries involved intricate switching of packages and destinations which were not obvious from a priori knowledge of the problem domain.

Remark: Although not addressed in this paper, the problem of finding syntactic symmetries of T is interesting in its own right. This problem is equivalent to the *graph isomorphism problem* (ISO) [6], whose complexity is a classic open problem in computer science: there are no polynomial-time algorithms known to solve ISO but there is evidence that it is not NP-complete, see, e.g., [16]) and it is rarely difficult in practice.

The group $G \leq \mathrm{Aut}(T)$ induces an equivalence relation on the set of truth assignments of L, wherein X is equivalent to Y if $Y = {}^{g}X$ for some $g \in G$; thus, the equivalence classes are precisely the *orbits* of G on the set of assignments. Note, further, that any orbit either contains only models of T or contains no models of T. This indicates why symmetries should reduce search: we can determine whether T has a model by visiting each equivalence class rather than visiting each truth assignment.

We illustrate this with an example:

Example: Let T be $a \vee \overline{c}$, $b \vee \overline{c}$, $a \vee b \vee c$, $\overline{a} \vee \overline{b}$ and let $G = \langle (a\ b) \rangle$. It is clear that $(a\ b) \in \mathrm{Aut}(T)$, in fact it is a syntactic symmetry. The two models of T are $(1, 0, 0)$ and $(0, 1, 0)$ (where the first, second and third coordinates are true/false values of a, b and c respectively). Clearly, this permutation maps models to models. We can "break" this symmetry by adding the clause $b \rightarrow a$ which eliminates one of the models, $(0, 1, 0)$, leaving us with only one model from the orbit. Thus the symmetry-breaking formula for T is $(b \rightarrow a)$.

In general, we introduce an ordering on the set of variables, and use it to construct a lexicographic order on the set of assignments. We will then add

a formula that is true of only the lexically largest model under this ordering, within each orbit.[3] Equation (1) is an example of such a formula.

The size of the lex-leader formulas we obtain for abelian groups is $O(n^3\mu(\log n))$ where n is the size of the permutation domain (Theorem 5.4). We remark that n is not necessarily the size of the input problem. If the input is a boolean formula, then n is the number of variables and the size of the formula could possibly be much larger than $O(n^3)$. At the same time, the symmetry group may not act on all n points, so the bound we obtain for abelian groups might be overly pessimistic. Another option might be to break symmetries partially by writing lex-leaders for only some orbits of assignments.

It is also worth noting that that the problem of finding $\Lambda(G)$ for an arbitrary abelian $G \leq \mathrm{Sym}(\Omega)$ does arise from consideration of some boolean formula T. Specifically, the action of any such G can be extended to some polynomial-size $\bar{\Omega} \supseteq \Omega$ such that $G = \mathrm{Aut}(T)$, where T is defined on the variables $\bar{\Omega}$. With respect to an ordering of $\bar{\Omega}$ that begins with Ω, the lex-leader approach to finding a symmetry-breaking formula includes finding a suitable $\Lambda(G)$ for the given Ω action.

Since we allow an arbitrary group of symmetries as input, the symmetry breaking formula depends only the group and not on the input theory. Because of this, we may ignore the presence of the input theory and focus on the size of lex-leader formulas for various groups. As a result, Theorems 5.1, 5.2, 5.3 and 5.4 make no mention of theories.

5 Statement of Results

We now summarize our results in the following theorems. Proofs are included in subsequent sections.

Theorem 5.1 *There are an infinite number of pairs G, Ω, where $G \leq \mathrm{Sym}(\Omega)$, such that the number of non-prunable clauses in $\Lambda_{\mathrm{nat}}(G, \rho)$ is c^n for all possible orderings ρ of Ω, where c is a constant > 1 and $n = |\Omega|$. In fact, these groups G have orbits of size ≤ 2 and are, therefore, elementary abelian 2-groups.*

Nevertheless, we show:

Theorem 5.2 *Let $G \leq \mathrm{Sym}(\Omega)$ have orbits of size ≤ 2. Then, for any ordering ρ of Ω, one can find a lex-leader formula $\Lambda(G, \rho)$ of size $O(n^3)$.*

However, we also prove that unless $P = NP$, there is no polynomial-time algorithm that computes a lex-leader formula for an arbitrary group G (*even for abelian G*):

[3] We note that this is surely not the *only* way to create symmetry-breaking formulas. One can break symmetries by adding any formula that is true of one member of each equivalence class.

Theorem 5.3 *The problem of testing whether a 0/1 string X is the lex-leader in its G-orbit is coNP-complete. This is the case even if G is abelian with orbits of size 3.*

We remark that a slightly weaker result with orbits of size 4 can be deduced from [2] (Proposition 3.1). A result of this form was also noted in [7] (Theorem 3.2) but the groups were nonabelian and the orbits unbounded. In that case the groups were explicitly constructed as the automorphisms of specified theories. It is possible, though less convenient in this case, to show how the group class underlying Theorem 5.3 arises as automorphism groups.

Finally, we show that the hardness suggested by Theorem 5.3 can be circumvented by a careful choice of variable ordering. We prove:

Theorem 5.4 *For abelian groups $G \leq \mathrm{Sym}(\Omega)$, one can find an ordering ρ of Ω and a lex-leader formula $\Lambda(G, \rho)$ of size $O(n^3\,\mu(\log n))$ ($|\Omega| = n$) where $\mu(r)$ is the time to multiply two r-digit integers.*

Proofs appear as follows: Theorem 5.1 in Section 6.1, Theorem 5.2 in Section 7.4, Theorem 5.3 in Section 6.2, Theorem 5.4 in Section 7.5.

6 Hardness of Lex-Leader Formulas

In this section, we study obstructions to the construction of certain lex-leader formulas. We show an exponential lower bound to the "natural" formula $\Lambda_{\mathrm{nat}}(G)$ (Section 6.1) even for groups with orbits of size 2. We also show that, in general, determining lex-leadership with respect to given orders is NP-hard (Section 6.2) even for abelian groups with orbits of size 3. The reader should contrast these results to the positive results described in Theorems 5.2, 5.4.

6.1 Exponential Lower Bounds for the "Natural" Formula

In this subsection, we exhibit an exponential lower bound on the size of the naïve lex-leader formula, $\Lambda_{\mathrm{nat}}(G)$, proving Theorem 5.1.

Given $\Omega = \{1, 2, \ldots, n\}$, $G = \langle S \rangle \leq \mathrm{Sym}(\Omega)$, recall from Equation (1) that the formulas associated with $g \in G$ are $C(g, i)$:

$$[({}^gX)_{i-1} \;=\; X_{i-1}] \rightarrow [X(i) \geq ({}^gX)(i)]\,, \text{ for } i = 1, \ldots, n \tag{4}$$

We now consider the case when n is even and the orbits of G are $\{2i-1, 2i\}$ for $1 \leq i \leq n/2$. Then G is an elementary abelian 2-group (every element in G has order 2) and G can be identified with a subspace of $\mathbb{Z}_2^n$ as follows: every permutation g in G corresponds to a vector v_g in $\mathbb{Z}_2^{n/2}$ such that $v_g(i) = 1$ iff $(2i-1)^g = 2i$, where $1 \leq i \leq n/2$ and $v_g(i)$ is the ith coordinate of v_g. So the group $G = \langle S \rangle$ corresponds to the vector subspace $V \leq \mathbb{Z}_2^{n/2}$ where $\{v_s |\ s \in S\}$ is now a set of *basis vectors* of V.

Equation (4) is necessarily trivially true when $i^g = i$. Because G has orbits of size 2, it is also a true when i is even: suppose i is even and $i^g \neq i$. Then this means $i^g = i-1$ and $(i-1)^g = i$. This, in turn, means that the antecedent of (4) implies $(X(i-1) = X(i))$. The consequent is $X(i) \geq X(i-1)$. If the antecedent is true, the consequent is trivially true and the whole expression is satisfied. If the antecedent is false, the whole expression is again true. Thus we need to consider clauses of the form $C(g, 2i-1)$, when $(2i-1)^g \neq 2i-1$ (which forces $(2i-1)^g = 2i$). (A very similar argument shows that the clause $C(g,i)$ is trivial for all $g \in G$ and i such that $i^g \leq i$ for *any* group G.)

If i is odd and $i^g \neq i$, then the expression $[({}^gX)_{i-1} = X_{i-1}]$ reduces to equality of X over the 2-element orbits where g moves points. Thus we may rewrite Equation (4) for $C(g, 2i-1)$ to get

$$\left[\bigwedge_{1 \leq k \leq i-1} \{X(2k-1) = X((2k-1)^g)\}\right] \rightarrow X(2i-1) \geq X((2i-1)^g)$$

We say that $C(g, 2i-1)$ is *nontrivial* if $(2i-1)^g \neq 2i-1$.

Thus we can prune $\Lambda_{\mathrm{nat}}(G)$ to $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ defined by the following equation:

$$\widetilde{\Lambda}_{\mathrm{nat}}(G) = \bigwedge_{g \in G} \bigwedge_{\substack{1 \leq i \leq n/2 \\ (2i-1)^g \neq 2i-1}} C(g, 2i-1) \tag{5}$$

We will now show that there are groups G for which the number of non-prunable clauses of $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ have exponential size.

For $g \in G$, $1 \leq i \leq n/2$, let $v_{g,i} \in \mathbb{Z}_2^i$ be the projection of v_g in the first i coordinates, i.e., $v_{g,i}(j) = 1$ iff $(2j-1)^g = 2j$ for $1 \leq j \leq i$. Observe that if $C(g, 2i-1)$ is nontrivial then $v_g(i) = 1$. For $v, w \in \mathbb{Z}_2^k$, let $v \preceq w$ iff $v(i) \leq w(i)$ for all $1 \leq i \leq k$. In other words, the order $\preceq$ is the lattice-theoretic order defined by set inclusion. For $1 \leq i \leq n/2$, define

$$V_i = \{v_{g,i} \in V \,|\, v_g(i) = 1\}.$$

V_i is also a lattice under the partial order defined by set-theoretic inclusion (inherited from $\mathbb{Z}_2^i$). Note by definition, the zero vector is not in V_i.

Lemma 6.1 *Let $C(g_1, 2i_1-1)$ and $C(g_2, 2i_2-1)$ be two non-trivial clauses in $\widetilde{\Lambda}_{\mathrm{nat}}(G)$. Then $C(g_1, 2i_1-1)$ prunes $C(g_2, 2i_2-1)$ iff $i_1 = i_2$ and $v_{g_1,i} \preceq v_{g_2,i}$ where $i = i_1 = i_2$.*

Proof: The "only-if" direction is easy to prove. We now prove the non-trivial direction.

($\Rightarrow$) Suppose $i_1 \neq i_2$. We exhibit an X which makes $C(g_1, 2i_1-1)$ true and $C(g_2, 2i_2-1)$ false, contradicting the hypothesis. Define $I_1 = \{l \mid v_{g_1,i_1}(l) = 1\}$ and $I_2 = \{l \mid v_{g_2,i_2}(l) = 1\}$. Note that $i_1 \in I_1$ and $i_2 \in I_2$.

We define X as follows:

$$\begin{array}{lll} X(2k-1)=0, & X(2k)=0 & \text{if } k \in I_2, k \neq i_2 \\ X(2i_2-1)=0, & X(2i_2)=1 & \\ X(2k-1)=1, & X(2k)=0 & \text{if } k \notin I_2 \end{array}$$

Every coordinate not in I_1 or I_2 is set to 0 in X. The clause $C(g_2, 2i_2-1)$ is false under this X. We show that if $i_1 \neq i_2$ and $I_1 \not\subseteq I_2$, the clause $C(g_1, i_1)$ is true, contradicting the hypothesis.

The antecedent of $C(g_j, 2i_j-1)$ for $j \in \{1,2\}$ is

$$\bigwedge_{k \in I_j \setminus \{i_j\}} X(2k-1) = X(2k)$$

and the consequent of $C(g_j, 2i_j-1)$ for $j \in \{1,2\}$ is

$$X(2i_j-1) \geq X(2i_j).$$

If $i_1 \neq i_2$, the consequent of $C(g_1, 2i_1-1)$, i.e., $X(2i_1-1) \geq X(2i_1)$ is true because either $i_1 \notin I_2$, in which case $X(2i_1-1)=1, X(2i_1)=0$ or $i_1 \in I_2$ in which case $X(2i_1-1)=0, X(2i_1)=0$ since $i_1 \in I_2 \setminus \{i_2\}$. Hence in either case, $C(g_1, 2i_1-1)$ is true.

Suppose $i_1 = i_2$ but $I_1 \not\subseteq I_2$. (Note that this is equivalent to $v_{g_1,i} \not\preceq v_{g_2,i}$ where $i = i_1 = i_2$) Then there is some $l \in I_1 \setminus I_2$ such that the term $X(2l-1) = X(2l)$ appears in the antecedent of $C(g_1, 2i_1-1)$. So the antecedent of $C(g_1, 2i_1-1)$ is false. Hence $C(g_1, 2i_1-1)$ is true. $\square$

In general, it is possible that a clause in $\Lambda_{\text{nat}}(G)$ for an arbitrary group G, cannot be pruned away by a single other clause but some conjunction of clauses prunes it. For groups under consideration, we show that this not possible.

Lemma 6.2 *Let $\mathcal{C} = \{C(g_1, 2i_1-1), C(g_2, 2i_2-1), \ldots, C(g_k, 2i_k-1)\}$ be a collection of clauses such that their conjunction*

$$\bigwedge_{C \in \mathcal{C}} C$$

prunes a clause $C(g, 2i-1)$ then each $C \in \mathcal{C}$ prunes $C(g, 2i-1)$.

Proof: Let $I = \{l \,|\, v_{g,i}(l) = 1\}$ and assign X as follows. For all $l \in I, l \neq i$ let $X(2l-1)=0, X(2l)=0$ and $X(2i-1)=0, X(2i)=1$. For all $l \notin I$ let $X(2l-1)=1, X(2l)=0$. Observe that $C(g, 2i-1)$ is false for this X. If for $1 \leq j \leq k$, we have $i_j \neq i$, then X makes $C(g_j, 2i_j-1)$ true. Hence we must have $i_j = i$ for each $1 \leq j \leq k$. If $i_j = i$ but $v_{g_j,i} \not\preceq v_{g,i}$, then $C(g_j, 2i_j-1)$ is true. However X makes $C(g, i)$ false. Hence it must be the case that for each j, $i_j = i$ and $v_{g_j,i} \preceq v_{g,i}$. Now Lemma 6.1 implies that $C(g_j, 2i_j-1)$ prunes $C(g, 2i-1)$. $\square$

The following lemma gives a combinatorial interpretation to logical prunability in $\widetilde{\Lambda}_{\text{nat}}(G)$:

Lemma 6.3 *A non-trivial clause $C(g, 2i-1)$ in $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ is non-prunable iff $v_{g,i}$ is minimal in V_i.*

Proof: ($\Leftarrow$) The clause $C(g, 2i-1)$ is prunable if there is some set of clauses $C(g_{i_j}, 2i_j - 1)$ in $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ which prunes it. Lemma 6.2 implies that this means that $C(g_{i_j}, 2i_j - 1)$ prunes $C(g, 2i-1)$ for each j. Lemma 6.1 now implies that $i_j = i$ and $w = v_{g_{i_j},i} \prec v_{g,i}$.

The reverse direction follows from Lemma 6.1. □

In particular, Lemma 6.3 provides a bijection between the non-prunable clauses in $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ and the minimal elements of the lattice V_i. Define $\min(V_i) = \{v \in V_i |\ \forall w \in V_i, w \preceq v \rightarrow v = w\}$, i.e., $\min(V_i)$ is the set of minimal elements of V_i. We can thus conclude

Proposition 6.1 *Let $G \cong V \leq \mathbb{Z}_2^n$. The number of non-prunable formulas in $\widetilde{\Lambda}_{\mathrm{nat}}(G)$ is $\sum_{i=1}^{n} |\min(V_i)|$.*

Henceforth, we will work with these groups in their vector space representation, i.e., as subspaces of $\mathbb{Z}_2^n$ for some n. Our goal will be to exhibit subspaces of $\mathbb{Z}_2^n$ with exponentially large $|\min(V_n)|$ – these will represent groups with an exponential number of distinct non-prunable clauses.

We define the subspace $V = V(n) \leq \mathbb{Z}_2^{2n+1}$ as follows. For $S \subseteq \{1, \ldots, n\}$ let $v_S \in V(n) \leq \mathbb{Z}_2^{2n+1}$ be defined as follows:

$$v_S(i) \quad = \quad \begin{cases} 1 & \text{if } i \in S \\ v_S(i-n) + |S| \mod 2 & \text{if } n+1 \leq i \leq 2n \\ |S| \bmod 2 & \text{if } i = 2n+1 \\ 0 & \text{otherwise} \end{cases}$$

In other words, v_S has the incidence vector of S in the first n coordinates, either a copy of the same incidence vector in the next n coordinates (if $|S|$ is even) or the incidence vector of the complement of S in the next n coordinates (if $|S|$ is odd). The last coordinate of v_S is the "parity check" bit of S.

Set

$$V(n) = \{v_S |\ S \subseteq \{1, \ldots, n\}\}.$$

Lemma 6.4 *Any vector $v_S \in V$ with $|S|$ odd is minimal in $V \setminus \{0\}$.*

Proof: Suppose $v_{S'} \in V$ be such that $v_{S'} \prec v_S$ and $S' \neq S$ and $S' \neq \emptyset$. This necessarily implies that $S' \subseteq S$. If $|S'|$ is even, then $S' \subseteq \bar{S}$ (looking at the last $n+1$ coordinates). This means that $S' = \emptyset$ (a contradiction). If $|S'|$ is odd and $S' \neq S$, then $v_{S'} \prec v_S$ implies $S' \subseteq S$ when you consider the first n coordinates and $S \subseteq S'$ when you consider the last $n+1$ coordinates. So $S = S'$, a contradiction. □

Lemma 6.5 *For any ordering of coordinates,*

$$|\min(V_{2n+1})| \geq 2^{n-2}.$$

Proof: The set of vectors $\mathcal{M} = \{v_S \mid S \subseteq \{1, 2, \ldots, n\}, |S| \text{ is odd } \}$ remains minimal in V irrespective of the ordering of coordinates. Thus in any ordering of coordinates, $\min(V_{2n+1}) = \mathcal{M} \cap V_{2n+1}$. Since at least half of the vectors in $\mathcal{M}$ have 1's in the $(2n+1)$-th coordinate, $\min(V_{2n+1}) \geq |\mathcal{M}|/2$. Since $|\mathcal{M}| = 2^{n-1}$, we have the desired result. □

Observe that as long as the orbits of G in Ω all have size 2 then we only need to consider $C(g, i)$ where i is the first element in its orbit and the position of the second element is not relevant. Hence for the corresponding group $G = G(n) \leq \mathrm{Sym}(\Omega)$ (where $|\Omega| = 4n + 2$ and $G(n) \equiv V(n) \leq \mathbb{Z}_2^{2n+1}$) the number of non-prunable clauses is at least 2^{n-2} for any ordering of the variables. Thus we have a proof of Theorem 5.1.

Remark **[Sperner spaces]**: We have seen that $\Lambda_{\mathrm{nat}}(G)$ cannot be pruned below an exponential size when G corresponds to a vector space with an exponential number of minimal vectors. This would be the case if the vector space were such that *all* non-zero vectors were incomparable (in the inclusion order). This suggested to us a concept of *Sperner spaces*. These are subspaces of $\mathbb{Z}_2^n$ such that, for all non-zero vectors $v, w \in V$, $v \preceq w \rightarrow v = w$. The terminology stems from a relation to the Sperner families (see [9]) of extremal set theory. These structures have also arisen in the study of statistical designs ([15], [21]). In a future paper ([20]), we further investigate the combinatorics of Sperner spaces. We show, in particular, that with high probability a random subspace of $\mathbb{Z}_2^n$ is Sperner. This indicates an abundance of groups satisfying the conclusion of Theorem 5.1. We also show that testing whether a group is a Sperner space is coNP-complete.

6.2 Order Sensitivity of Lex-Leader Formulas

Since a lex-leader formula for $G \leq \mathrm{Sym}(\Omega)$ has to assume that Ω is ordered, it is conceivable that the size of the lex-leader formula could vary widely depending on what ordering was chosen for Ω. This is because the complexity of finding lex-leaders is dependent on the input ordering. While this problem is solvable in polynomial time for even solvable (and beyond) groups when we assume an ordering of the permutation domain, it is NP-hard (and not known to be in NP) for elementary abelian 3-groups for some orderings of the permutation domain, as asserted in Theorem 5.3.

Proof of Theorem 5.3

We show that testing whether there exists $g \in G$ such that ${}^gX > X$ is NP-complete. This is clearly equivalent to the original problem. This is done via a reduction from Exact 3-Cover [10]:

Problem: Exact 3-Cover
Input: A set Γ and a collection Θ of 3-element subsets of Θ.
Question: Is there a sub-collection $\Theta' \subseteq \Theta$ such that $\Gamma = \bigcup_{\theta \in \Theta'} \theta$.
Given an instance (Γ, Θ) of Exact 3-Cover, we construct a linearly ordered set Ω, $G \leq \mathrm{Sym}(\Omega)$, and a string X on Ω as follows.

Ω: Let Δ be the set of unordered pairs of elements of Γ, and let $\Psi = \{\{\theta, \theta'\} \mid \theta, \theta' \in \Theta,\ \theta \cap \theta' \neq \emptyset\}$, the set of unordered pairs of intersecting triples. Let $\Phi = \Gamma \cup \Delta \cup \Psi$. Set $\Omega := \{1,2,3\} \times \Phi$, fix any linear ordering of Φ and order Ω lexicographically (so that $i \times \Phi$ precedes $j \times \Phi$ if $i < j$).

G: First let s denote the 3-cycle $(1,2,3)$. For $\theta \in \Theta$, we define $g_\theta \in \mathrm{Sym}(\Omega)$ so that

$$(i,\omega)^{g_\theta} = \begin{cases} (i^{s^{-|\omega \cap \theta|}}, \omega) & \text{for } \omega \in \Gamma \cup \Psi \\ (i^{s^{|\omega \cap \theta|}}, \omega) & \text{for } \omega \in \Delta \end{cases}$$

(so the nontrivial orbits of g_θ are 3-cycles). Set $G = \langle \{g_\theta \mid \theta \in \Theta\} \rangle$.

X: Let X take the value 1 on $\{1,3\} \times \Phi$ and 0 on $\{2\} \times \Phi$.

Suppose that $\Theta' \subseteq \Theta$ is a perfect cover of Γ and set $g := \prod_{\theta \in \Theta'} g_\theta$. We show that ${}^gX$ takes the value 1 on $\Phi \times 1$: for $\gamma \in \Gamma$, $|\gamma \cap \theta| = 1$ for exactly one $\theta \in \Theta'$ so that $(1,\omega)^g = (3,\omega)$ and ${}^gX(1,\omega) = X(3,\omega) = 1$; for $\psi \in \Gamma \cup \Psi$, $|\psi \cap \theta| \leq 1$ for all $\theta \in \Theta'$ so that $(1,\psi)^g$ is either $(1,\psi)$ or $(3,\psi)$ and, in either case ${}^gX(1,\psi) = 1$; for $\delta \in \Delta$, $\sum_{\theta \in \Theta'} |\delta \cap \theta| = 2$, so again, ${}^gX(1,\delta) = X(3,\delta) = 1$. But ${}^gX$ also takes the value 1 on some elements of $\Phi \times 2$ since, for $\gamma \in \Gamma$, ${}^gX(2,\gamma) = X(1,\gamma) = 1$. Hence ${}^gX$ precedes X in lexicographical order.

Conversely, suppose that ${}^gX$ precedes X for some $g \in G$. Let such $g = \prod_{i=1}^{r} g_{\theta_i}$. We may assume no θ_i appears ≥ 3 times in this product, else we could drop out those three occurrences. No θ_i could appear twice, otherwise for $\gamma \in \theta_i$, ${}^gX(1,\gamma) = X(2,\gamma) = 0$, in which case X would precede ${}^gX$. Now set $\Theta' := \{\theta_i\}_{1 \leq i \leq r}$. If $\theta, \theta' \in \Theta'$, $\theta \cap \theta' = \emptyset$, since otherwise $\psi = \{\theta, \theta'\} \in \Psi$ and so $(1,\psi)$ would be moved by both g_ψ and $g_{\psi'}$ forcing ${}^gX(1,\psi) = X(2,\psi) = 0$. To show that Θ' is a cover, fix $\alpha \in \theta_1$ and let $\beta \in \Gamma$ be arbitrary and let $\delta = \{\alpha, \beta\}$; since $\alpha \in \delta \cap \theta_1$, $\sum_{1 \leq i \leq r} |\delta \cap \theta_i| \geq 1$, but equality cannot hold since that would imply ${}^gX(1,\delta) = X(2,\delta) = 0$ in which case X would precede ${}^gX$; hence $\beta \in \theta_i$ for some i. $\square$

7 Lex-Leader Formulas for Abelian Groups

In this section, we show how to write succinct lex-leader formulas for abelian groups. Let $G \leq \mathrm{Sym}(\Omega)$ be abelian. We reorder the orbits of G so that points in the same orbit appear consecutively. Let us suppose that there are r orbits. For an assignment $X \in 2^\Omega$, let $X_{\{i\}}$ ($X_{[i]}$) denote its projection in the i-th orbit (resp. first i orbits).

To express lex-leadership of X, we wish to assert that for each $1 \leq i \leq r$,

$$\neg \exists g : \ (g \in G) \wedge ({}^gX_{[i-1]} = X_{[i-1]}) \wedge ({}^gX_{\{i\}} > X_{\{i\}}) \tag{6}$$

We show that for abelian groups, we can express the condition (${}^gX_{[i-1]} = X_{[i-1]}$) as a system of equations, using a duality result (described in Section 7.1). Similarly, the conditions ($g \in G$) and (${}^gX_{\{i\}} > X_{\{i\}}$) can be expressed as a system of linear equations over an appropriately defined module. When we consider a subspace of $\mathbb{Z}_2^n$ as our group G, this module is, not surprisingly, a vector space over $\mathbb{Z}_2$. Thus Equation (6) asserts the non-existence of a solution to a set of linear equations. We first show in Section 7.3 how one can express the nonsolvability of a system of equations defined modulo arbitrary m as the satisfiability of a succinct boolean formula.

7.1 Duality

At a few critical points, it is necessary to exploit a well-known duality of abelian groups. We refer to [13, Section 13.2] or [18, Section 1.9] for background.

Suppose $G = \mathbb{Z}_{m_1} \oplus \cdots \oplus \mathbb{Z}_{m_k}$. The character group of G, denoted by G^*, can viewed as the (additive) group of homomorphisms $G \to \mathbb{Z}_m$, where $m = \mathrm{lcm}(m_1, \ldots, m_k)$, the exponent of G [18]. The group G^* is then isomorphic to G.

For computational purposes, it is convenient to fix a (non-canonical) isomorphism between G and G^* via a bilinear form on G. Namely, for $1 \le i \le k$, let z_i be a fixed generator of $\mathbb{Z}_{m_i}$. Then, for $h = \sum_{i=1}^k b_i z_i$, $g = \sum_{i=1}^k c_i z_i$ we define

$$h \cdot g = \sum_{i=1}^{k} \frac{m}{m_i} b_i c_i \bmod m.$$

Thus, if $\phi \in G^*$, we can take $b_i \in \mathbb{Z}$ such that $\frac{m}{m_i} b_i = \phi(z_i)$ (such b_i exists since $m_i \phi(z_i) = \phi(m_i z_i) = 0$) so that $h = \sum_{i=1}^k b_i z_i$ satisfies $h \cdot g = \phi(g)$ for all $g \in G$. It follows that the map $h \mapsto F_h$ where $F_h(g) = h \cdot g$ identifies G with G^*. Furthermore, if we define, for $H \le G$, $H^\perp = \{g \in G \mid F_g(H) = 0\}$, then we have the well-known result

Lemma 7.1 *For $H \le G$, $H^{\perp\perp} = H$.*

Hence, given a generating set $\{\sum_i a_{ji} z_i\}_{j \in J}$ for $H^\perp$, then $\sum_i t_i z_i \in H$ iff $\sum_i a_{ji} t_i \equiv 0 \bmod m$ for $j \in J$.

We use this fundamental result in two ways.

First, given a permutation group H by generators, one can compute a generating set for $H^\perp$ in polynomial time. This essentially involves solving a system of equations. This is used in Section 7.4 and Section 7.5, when we wish to express membership of a permutation h in H by specifying that it has to be annihilated by the generators of the dual $H^\perp$.

But we also consider instances where H is initially known only as the subgroup that fixes a string X but, fortunately, G is a small (listable) group. In this case, we can express membership in H by asserting orthogonality to all those elements of G that fix X. This is used in Section 7.5 when we wish to express membership of a permutation g in the subgroup K fixing a string X, again, via

a system of equations expressing that g is annihilated by the permutations in $K^{\perp}$.

7.2 Nonsolvability as Solvability

An essential ingredient in our ability to write succinct lex-leader formulas for abelian groups is that we can express the nonsolvability system of linear equations $Ax \equiv b \bmod p^e$ (p prime) as the solvability of another system of linear equations mod p^e. The following (folklore) lemma reduces nonsolvability to solvability:

Lemma 7.2 *The system of equations $Ax \equiv b \bmod p^e$ (p is prime) is not solvable iff the system*

$$[A\ b]^T y \equiv (0, 0, \ldots, 0, p^{e-1})^T \bmod p^e$$

is solvable.

Proof: The system is solvable iff b is in subgroup H of $\mathbb{Z}_{p^e}^r$ generated by the columns of A (where r is the number of rows in A). This is the case iff $b \in H^{\perp\perp}$ (as in Section 7.1). Hence the system is nonsolvable iff there is some $x \in \mathbb{Z}_{p^e}^r$ such that $x \in H^{\perp}$ but $x \cdot b \not\equiv 0 \bmod p^e$. By taking a multiple of such x if necessary, we have $[A\ b]^T x \equiv (0, 0, \ldots, p^{e-1})^T \bmod p^e$. $\square$

Remark:

i) A more general result can be stated that relates the nonsolvability of $A\,x \equiv b \bmod m$ (where m is any positive integer) to the solvability of any system in a collection of linear systems $A_p x \equiv b_p \bmod p^e$ for each prime $p \,|\, m$ where p^e is the largest power of p dividing m.

ii) This reduction of nonsolvability to solvability is a polynomial time reduction and in particular, the size of the system $[A\,b]^T y \equiv (0, 0, \ldots, p^{e-1})^T \bmod p^e$ is $(n+1) \times n$ if $A\ x \equiv b \bmod p^e$ was an $n \times n$ system.

7.3 Solvability as Boolean Satisfiability

In this section, we show how to express (non)solvability of a system of equations modulo m as the satisfiability of a succinct boolean formula.

Let $\mu(r)$ denote the time to multiply two r-bit integers. Since division has the same complexity as multiplication, we can assume that we can add, subtract, multiply and divide r-bit integers in $\mu(r)$ time. It is well-known (e.g., see von zur Gathen[11], chapter 8) that $\mu(r) = O(r \log r \log\log r)$.

We prove the following theorem:

Theorem 7.3 *Let $Ax \equiv b \bmod m$ be a system of equations where A is a $n \times n$ matrix. Then one can find a boolean formula $\overline{\phi}(A, b)$ of size $O(n^2\,\mu(\log m))$ which is satisfiable iff $\mathcal{E}$ is not solvable.*

Theorem 7.3 is first shown in the special case when the equations are defined over $\mathbb{Z}_2$. Let $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ be an n-bit vector from $\mathbb{Z}_2^n$. Also, let $b \in \mathbb{Z}_2$. Let $E\,(= E(\epsilon, b))$ denote the equation $\sum_{i=1}^{n} \epsilon_i x_i = b$.

Lemma 7.4 *One can construct a boolean formula ϕ of size $\Theta(n)$ which is satisfiable iff E is solvable.*

Proof: Observe that E is solvable iff the equations $\mu_1 = \epsilon_1 x_1$, $\mu_i = \mu_{i-1} + \epsilon_i x_i$ for $2 \leq i \leq n-1$ and $b = \mu_{n-1} + \epsilon_n x_n$ are simultaneously solvable where $\mu_i, 1 \leq i \leq n-1$ are new variables. This system is solvable iff the boolean formula

$$(\mu_1 \leftrightarrow (\epsilon_1 \wedge x_1)) \wedge \bigwedge_{2 \leq i \leq n-1} (\mu_i \leftrightarrow (\mu_{i-1} \oplus (\epsilon_i \wedge x_i))) \wedge (b \leftrightarrow (\mu_{n-1} \oplus (\epsilon_n \wedge x_n)))$$

is satisfiable ($\oplus$ refers to the exclusive-or operator). $\Box$

Given a system of equations, we can now apply the construction in Lemma 7.4 to each equation.

Proposition 7.1 *Let $Ax = b$ be a system of equations over $\mathbb{Z}_2$ where A is an $m \times n$ matrix. Then one can find a boolean formula $\phi(A, b)$ of size $\Theta(mn)$ which is satisfiable iff $Ax = b$ is solvable.*

We have seen in Lemma 7.2 (also see remark following Lemma) that we can express the nonsolvability of a system $A\,x \equiv b \bmod 2$ as the solvability of the system $[A\,b]^T y \equiv (0, 0, \ldots, 1)^T \bmod 2$. So now Proposition 7.1 implies that one can find a boolean formula expressing nonsolvability of $A\,x \equiv b \bmod 2$.

Proposition 7.2 *Let $Ax = b$ be a system of equations over $\mathbb{Z}_2$ (where A is an $m \times n$ matrix). Then one can find a boolean formula, $\overline{\phi}(A, b)$, of size $\Theta(mn)$, which is satisfiable iff $Ax = b$ is* not *solvable.*

Remark: The ability to write a system of equations which is solvable iff $Ax = b$ is not solvable allowed us to express nonsolvability as solvability. Since we want a boolean formula which is satisfiable iff $Ax = b$ is not solvable, we remind the that reader that it *does not* suffice to put a negation sign in front of $\phi(A, b)$ (the "solvability" formula of Proposition 7.1).

We now prove Theorem 7.3. We first develop machinery, akin to Proposition 7.1 for vector spaces, to represent arithmetic mod m as satisfiability of a boolean formula.

Recall that a boolean circuit C is a directed acyclic graph (DAG) whose vertices are labeled with the names of Boolean connectives $\wedge$, $\vee$, $\neg$ (the logic gates) or variables (inputs). Each boolean circuit computes a boolean function $f : \{0,1\}^m \to \{0,1\}^n$ that is a mapping from the values of its m input variables to the values of its n outputs. The size of a circuit $s(C)$ is the number of logic gates. We also assume that the fan-in of a circuit (the in-degree of any vertex) is at most 2. To take care of trivialities, we make the assumptions that $s(C) = \Omega(m)$ and $m = \Theta(n)$.

The following lemma is folklore and is easy to prove:

Lemma 7.5 *Let C be a circuit computing a boolean function $f(x_1, x_2, \ldots x_m) = (y_1, y_2, \ldots, y_n)$. Then one can construct a boolean formula $\mathcal{F}(C)$ of size $O(s(C))$ defined over $x_1, x_2, \ldots x_m, y_1, y_2, \ldots y_n$ (and additional variables) whose models are such that the value of $(y_1, y_2, \ldots y_m)$ is $f(\alpha_1, \alpha_2, \ldots, \alpha_m)$ where α_i is the value of x_i in the model.*

Recall that $\mu(n)$ is the time to multiply two n-bit integers (equivalently, $\mu(n)$ is the size of a circuit that computes the product of the integers). It is well-known ([22]) that all primitive operations (addition, subtraction, multiplication, division) of n-bit integers can be done by circuits of size $O(\mu(n))$.

Let $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ be an n-bit vector from $\mathbb{Z}_m^n$. Also, let $b \in \mathbb{Z}_m$. Let $E\,(= E(\epsilon, b))$ denote the equation $\sum_{i=1}^{n} \epsilon_i x_i = b$ over $\mathbb{Z}_m$.

Lemma 7.6 *One can construct a boolean formula ϕ of size $O(n\ \mu(\log m))$ which is satisfiable iff E is solvable.*

Proof: E is solvable iff the following system of equations is solvable:

$$\begin{aligned}
&\mu_i = \epsilon_i x_i, 1 \leq i \leq n \\
&\gamma_1 = \mu_1, \\
&\gamma_i = \gamma_{i-1} + \mu_i, 2 \leq i \leq n-1 \\
&b = \gamma_{n-1} + \mu_n
\end{aligned}$$

For each equation above, let the right-hand side represent a function computed by a circuit C (this circuit does computation modulo m) and assume $y_1, y_2, \ldots y_{\lceil \log m \rceil}$ are the output bits for C. Let the variable on the left-hand side be x, represented by bits $x_1, x_2, \ldots, x_{\lceil \log m \rceil}$. Then we write a formula $\mathcal{F}(C) \wedge \bigwedge_i (x_i = y_i)$ equivalent to this particular equation where $\mathcal{F}(C)$ is as described in Lemma 7.5. The conjunction of formulas for each equation is our desired ϕ. Correctness and size estimates are easy to prove. $\square$

The next lemma now follows.

Lemma 7.7 *Let $Ax = b$ be a system of equations over $\mathbb{Z}_m$ where A is an $n \times n$ matrix. Then one can construct a boolean formula $\phi(A, b)$ of size*

$$O(n^2\, \mu(\log m))$$

which is satisfiable iff $Ax = b$ is solvable.

We can now write a boolean formula expressing nonsolvability over $\mathbb{Z}_{p^e}$, which proves Theorem 7.3.

Lemma 7.8 *Let $\mathcal{E} : Ax \equiv b \bmod p^e$ be a system of equations where A is a $n \times n$ matrix. Then one can construct a boolean formula $\overline{\phi}(A, b)$ of size $O(n^2\, \mu(\log p^e))$ which is satisfiable iff $\mathcal{E}$ is not solvable.*

Proof: Lemma 7.2 reduces nonsolvability of $\mathcal{E}$ to solvability of the system $[A\ b]^T y = (0, 0, \ldots, p^{e-1})^T$. Hence $\mathcal{E}$ is solvable iff $\overline{\phi}(A, b) = \phi([A\ b]^T, (0, 0, \ldots, p^{e-1})^T)$ is satisfiable. The bound now follows. $\Box$

A useful technique in writing lex-leader formulas for abelian groups is the ability to rewrite a system of equations, where each equation is defined over a possibly different modulus, to an equivalent system (or systems) of equations over a uniform modulus.

Let $\mathcal{E}$ refer to the following $r \times s$ system of equations:

$$\sum_{1 \leq j \leq s} A(i, j) x_j \equiv b_i \bmod m_i$$

for integers $m_i \leq n$ for some n and $1 \leq i \leq r$.

We now show how to express the (non)solvability of $\mathcal{E}$ as the (non)solvability of a set of systems of equations (each defined over a uniform *small* prime power modulus).

Lemma 7.9 *One can write $O(n/\log n)$ systems of equations $\mathcal{E}_p$ for each prime $p \,|\, m_i$ for some i, such that $\mathcal{E}$ is solvable iff each such system $\mathcal{E}_p$ is solvable. Furthermore, each system $\mathcal{E}_p$ is defined over $\mathbb{Z}_{p^e}$ for some integer e such that $p^e \leq n$, and has $O(r)$ equations in $O(s)$ unknowns.*

Proof: Each equation $\sum_{1 \leq j \leq s} A(i, j) x_j = b_i \bmod m_i$ is solvable iff $\sum_{1 \leq j \leq s} A(i, j) x_j = b_i \bmod p^{e_i}$ is solvable for each prime p such that $p^{e_i} \mid m_i$ and $p^{e_i+1} \nmid m_i$. Thus by the Chinese remainder theorem, $\mathcal{E}$ is solvable iff each of the systems of equations $\mathcal{E}_p : \sum_{1 \leq j \leq s} A(i, j) x_j = b_i \bmod p^{e_i}, 1 \leq i \leq r$ is solvable for each such prime $p \mid m_i$ for some $1 \leq i \leq r$. Note that $\mathcal{E}_p$ might contain fewer than r equations, since it might be the case that $e_i = 0$ for some i and so we can remove the trivial equation $\sum_{1 \leq j \leq s} A(i, j) x_j = b_i \bmod 1$ from $\mathcal{E}_p$. It might also have fewer than s variables, if certain variables only appear with coefficients which are powers of p^{e_i}.

We can further rewrite $\mathcal{E}_p$ as a system of equations, where each equation is defined modulo p^e where $e = \max\{e_i\}$. To do this we multiply both sides of each equation $\sum_{1 \leq j \leq s} A(i, j) x_j = b_i \bmod p^{e_i}$ (where we now can assume that $e_i \neq 0$) by p^{e-e_i} to get the equivalent equation:

$$\sum_{1 \leq j \leq s} p^{e-e_i} A(i, j) x_j = p^{e-e_i} b_i \bmod p^e$$

We thus get a equivalent system of equations $\mathcal{E}_p$ defined over $\mathbb{Z}_{p^e}$ by applying the above transformation to each equation in $\mathcal{E}_p$.

Observe that the number of systems $\mathcal{E}_p$ is $O(n/\log n)$ (by the Prime Number Theorem, [23, ch 10]) since for some i, $p \,|\, m_i$ and $m_i \leq n$. $\Box$

Lemma 7.8 exhibits a boolean formula $\overline{\phi}_p$ of size $O(rs\,\mu(\log p^e)) = O(rs\,\mu(\log n))$ (since $p^e \leq n$) which is satisfiable iff $\mathcal{E}_p$ is not solvable (where, recall, $\mu(r)$ is the time to multiply two r-bit integers).

Thus we have the following lemma:

Lemma 7.10 *Let $\mathcal{E}$ be the following $r \times s$ system of equations:*

$$\sum_{1 \le j \le s} A(i,j)x_j \equiv b_i \bmod m_i$$

for integers $m_i \le n$ for some n and $1 \le i \le r$. Then one can construct a boolean formula $\overline{\phi}$ of size $O(rs(n/\log n)\mu(\log n))$ which is satisfiable iff $\mathcal{E}$ is not solvable.

Proof: The formula is

$$\bigvee_p \overline{\phi}_p.$$

Since the number of primes $\le n/\log n$, the size of this formula is $O(n/\log n \times rs\mu(\log n))$. $\square$

Remark: As we noted in proof to Lemma 7.9, the system $\mathcal{E}_p$ obtained from $\mathcal{E}$ by taking remainders mod p^e may end up with far fewer than the original s variables. This might lead to substantial savings in the size of the resulting boolean formula. If we assume that for each p, $\mathcal{E}_p$ has $O(r)$ equations and N_p variables, then the size of $\overline{\phi}_p$ is $rN_p\mu(\log p^e)$. As a result, the size of $\overline{\phi}$ in Lemma 7.10 becomes $\sum_{p\in\mathcal{I}} rN_p\mu(\log p^e)$ where $\mathcal{I} = \{p \mid p \text{ is prime and } p \mid m_i \text{ for some } i\}$ is the set of primes to consider. This leads to an order of magnitude savings in the size of the lex-leader formula for abelian groups, where $\sum_{p\in\mathcal{I}} N_p$ is small (much smaller than the pessimistic estimate of $O(ns/\log n)$).

7.4 Groups with Orbits of Size 2

In this section, we show how one can use linear algebra to write short lex-leader formulas for G when G is a subspace of $\mathbb{Z}_2^n$, thus proving Theorem 5.2 .

Let $G \le \mathrm{Sym}(\Omega)$ be as described in Section 6.1, i.e., $G \equiv W \le \mathbb{Z}_2^{n/2}$ be a group acting on n points $[n] = \{1, 2, \ldots, n\}$ where the orbits of G are the sets $\{2i-1, 2i\}$ for each $1 \le i \le n/2$ (after suitable reordering of Ω if necessary). Observe that $g \in G \equiv w \in W$ where $w_i = 1$ iff $(2i-1)^g = 2i$.

The assignment $X \in 2^{[n]}$ is a lex-leader under the action of G iff the following holds for each $1 \le i \le n/2$:

$$\neg w : \ (w \in W) \wedge (X_{[i-1]} = {}^wX_{[i-1]}) \wedge ({}^wX_{\{i\}} > X_{\{i\}}) \tag{7}$$

We now show that each subexpression in parenthesis in Equation (7) can be replaced by a set of linear equations over $\mathbb{Z}_2$:

$\underline{\boldsymbol{X_{[i-1]} = {}^wX_{[i-1]}}}$

The following lemma expresses this condition as a system of equations.

Lemma 7.11 *Let $X \in 2^{[n]}$ and $w \in W \le \mathbb{Z}_2^{n/2}$. For $1 \le i \le n/2$, one can write a system of linear equations which is satisfied iff $X_{[i-1]} = {}^wX_{[i-1]}$.*

Proof: Define the variable a_k to be 1 iff $X(2k-1) = X(2k)$, i.e.,

$$a_k \leftrightarrow X(2k-1) = X(2k).$$

If $X_{[i-1]} = {}^wX_{[i-1]}$, then for each orbit $j \leq i-1$, if $w_j = 1$ we must have $X(2j-1) = X(2j)$ (i.e., $a_j = 1$). We can express this condition by the linear equation in $\mathbb{Z}_2$:

$$(1 - a_j)w_j = 0$$

Thus we can express $X_{[i-1]} = {}^wX_{[i-1]}$ by the system of linear equations in $\mathbb{Z}_2$:

$$(1 - a_j)w_j = 0 \text{ for each } j,\ 1 \leq j \leq i-1.$$

The number of such equations is $O(n)$. □

$^wX_{\{i\}} > X_{\{i\}}$

The following lemma expresses this condition as the solvability of a linear system.

Lemma 7.12 *Let $X \in 2^{[n]}$ and $w \in W \leq \mathbb{Z}_2^{n/2}$. For $1 \leq i \leq n/2$, one can write a system of linear equations which is satisfied iff ${}^wX_{\{i\}} > X_{\{i\}}$.*

Proof: If $w_i = 0$, then clearly ${}^wX_{\{i\}} = X_{\{i\}}$. So,

$$({}^wX_{\{i\}} > X_{\{i\}}) \equiv (w_i = 1) \wedge (X(2i-1) = 0) \wedge (X(2i) = 1).$$

The right-hand side is clearly a system of linear equations. □

$w \in W$

The following lemma, a direct consequence of Lemma 7.1, shows that membership in W can be expressed as a set of linear equations.

Lemma 7.13 *Let $W \leq \mathbb{Z}_2^n$ and let $w \in \mathbb{Z}_2^n$. One can write a homogeneous system of equations over variable w_i, for $1 \leq i \leq n$, which is satisfied when $w \in W$.*

Proof: Given $W \leq \mathbb{Z}_2^n$ via a set of basis elements, one can find a basis S for $W^\perp \leq \mathbb{Z}_2^n$ in polynomial time (this step can be a preprocessing step before the lex-leader formula is constructed). This is equivalent to solving a set of linear equations. Now, because of Lemma 7.1, a vector $w \in \mathbb{Z}_2^n$ belongs to W iff $w \cdot x = 0$ for each vector $x \in S$. This, in turn, is another system of $O(n)$ linear equations in w_i. □

Combining Lemmas 7.11, 7.12 and 7.13, we have the following corollary:

Corollary 7.14 *Let $X \in 2^{[n]}$ and $W \leq \mathbb{Z}_2^{n/2}$. One can write $n/2$ systems of equations $\mathcal{E}(i)$ (for $1 \leq i \leq n/2$) each of which is nonsolvable iff X is a lex-leader under W. Furthermore, $\mathcal{E}(i)$ has $O(n)$ equations in $O(n)$ unknowns.*

Proof: For each $1 \leq i \leq n/2$, we have from Equation (7) that X is a lex-leader iff,

$$\neg w : \ (w \in W) \ \wedge \ (X_{[i-1]} = {}^{w}X_{[i-1]}) \ \wedge \ ({}^{w}X_{\{i\}} > X_{\{i\}})$$

Lemmas 7.11, 7.12, 7.13 imply that we can replace each of the conditions $(w \in W)$, $(X_{[i-1]} = {}^{w}X_{[i-1]})$ and $({}^{w}X_{\{i\}} > X_{\{i\}})$ by a system of equations. Let $\mathcal{E}(i)$ denote the resulting (aggregate) system of equations. Clearly $\mathcal{E}(i)$ has at most $2n+1$ equations and is defined over the unknowns w_i, $1 \leq i \leq n/2$.

Thus lex-leadership of X is equivalent to the *nonsolvability* (because of the negated existential quantifier in the expression in Equation (7)) of a system of equations $\mathcal{E}(i)$ for each i. $\square$

Hence we want a boolean formula which is satisfiable iff $\mathcal{E}(i)$ is not solvable. Proposition 7.2 shows that one can efficiently construct such a boolean formula $\overline{\phi}(i)$ of size $O(n^2)$.

Hence

$$\bigwedge_{1 \leq i \leq n/2} \overline{\phi}(i)$$

is satisfiable iff X is a lex-leader.

Thus we have a proof of the following theorem:

Theorem 7.15 *Let $G \leq \mathrm{Sym}(\Omega)$ be a group with orbits of size ≤ 2. Then for all orderings of Ω one can construct a lex-leader formula $\Lambda(G)$ of size $O(n^3)$.*

Thus while $\Lambda_{\mathrm{nat}}(G)$ for some groups of this class was of exponential size for any ordering of Ω (Theorem 5.1), $\Lambda(G)$ is of polynomial size if of polynomial size for every order.

7.5 Abelian Groups: General Case

In the general case, the projection of abelian $G \leq \mathrm{Sym}(\Omega)$ in each orbit is isomorphic to a direct product of cyclic groups. In this subsection, we consider this general case.

Let $\Delta_1, \Delta_2, \ldots, \Delta_r$ be the orbits of G in Ω. *We assume that Ω is ordered so that, for $i < j$ the points in Δ_i appear before the points Δ_j* (in particular, each orbit is contiguous). Recall that an abelian transitive group is regular [8] so that $|G^{\Delta_i}| = |\Delta_i|$. We write $g(i)$ for the projection of $g \in G$ in G^{Δ_i}. Since the points in the same orbit appear together, we can number the points in Δ_i as $\{0, 1, 2, \ldots, |\Delta_i| - 1\}$ without any confusion. For a string X, we let $X_{\{i\}}(j)$ refer to the value of X at the j-coordinate in its restriction to Δ_i, where $1 \leq i \leq r$ and $0 \leq j \leq |\Delta_i| - 1$.

Our goal, as before, is to rewrite the expression for lex-leader, namely, for each $1 \leq i \leq r$,

$$\neg\exists g: \ (g \in G) \wedge ({}^{g}X_{[i-1]} = X_{[i-1]}) \wedge ({}^{g}X_{\{i\}} > X_{\{i\}}) \tag{8}$$

as the nonsolvability of a system of equations over an appropriately defined module.

We now consider each subexpression in parenthesis inside Equation (8) and rewrite it as a system of equations:

$\underline{{}^{g}\boldsymbol{X}_{[i-1]} = \boldsymbol{X}_{[i-1]}}$

We focus on the j-th orbit ($j < i$) and show that $X_{\{j\}} = {}^{g}X_{\{j\}}$ can be expressed as a system of equations. We assume that group $H = G^{\Delta_j} = \mathbb{Z}_{m_1} \oplus \mathbb{Z}_{m_2} \oplus \cdots \oplus \mathbb{Z}_{m_t}$ and write $g(j,l)$ for the projection of $g(j)$ in $\mathbb{Z}_{m_l}$. With respect to this decomposition, we use the bilinear from $k, h \mapsto k \cdot h$ defined in Section 7.1.

Define for $k, h \in H$, and $X_{\{j\}}$:

$$\alpha(X_{\{j\}}, k, h) := \left(\left(\bigwedge_{0 \leq i \leq m-1} X_{\{j\}}(i) = X_{\{j\}}(i^h) \right) \rightarrow k \cdot h = 0 \right)$$

where $m = |H| = |\Delta_j|$. Observe that $\alpha(X_{\{j\}}, k, h)$ is equivalent to the condition $k \cdot h = 0$ for $h \in H$ such that ${}^{h}X_{\{j\}} = X_{\{j\}}$. For all other h, $\alpha(X_{\{j\}}, k, h)$ is true.

Also define

$$\beta(X_{\{j\}}, k) := \bigwedge_{h \in H} \alpha(X_{\{j\}}, k, h).$$

Let $K = K(X_{\{j\}})$ be the subgroup of H stabilizing $X_{\{j\}}$. Hence,

$$\begin{aligned} \beta(X_{\{j\}}, k) &:= \forall_{h \in K(X_{\{j\}})}(k \cdot h = 0) \\ &= (k \in K^{\perp}) \end{aligned}$$

For each $k \in H$, we include the linear equation[4]:

$$\beta(X_{\{j\}}, k) * [k \cdot g(j)] = 0 \tag{9}$$

So the number of equations is $|\Delta_j|$.

When $k \notin K^{\perp}$, the coefficient $\beta(X_{\{j\}}, k)$ is 0 so Equation (9) is trivially satisfied. The coefficient is 1 if $k \in K^{\perp}$. Hence, the set of equations says precisely that $g(j) \in K^{\perp\perp} = K$, i.e., g is in the stabilizer of $X_{\{j\}}$.

We can form equations of the form (9) for each of the first $i-1$ orbits for a total of $\sum_{j<i} |\Delta_j|$ equations. Thus we have the following lemma:

[4]More precisely, this is a congruence mod $\mathrm{lcm}(m_1, \ldots, m_k)$ involving variables $g(j,l)$

Lemma 7.16 *Let G be as above and let $g \in G$, $X \in 2^\Omega$. One can write a system of linear equations $\mathcal{E}_1(g,i)$ which is satisfied iff $X^g_{[i-1]} = X_{[i-1]}$. Furthermore $\mathcal{E}_1(g,i)$ has $O(n)$ equations in $O(n)$ unknowns.*

$^g\boldsymbol{X_{\{i\}} > X_{\{i\}}}$

We express this condition as the solvability of a (collection of) linear systems in the following lemma:

Lemma 7.17 *Let G be as above and let $g \in G$, $X \in 2^\Omega$. One can write a collection $\{\mathcal{E}_2(g,i,h) \mid h \in G^{\Delta_i}\}$ of linear equation-systems such that $^gX_{\{i\}} > X_{\{i\}}$ iff $\mathcal{E}_2(g,i,h)$ is satisfied for some $h \in G^{\Delta_i}$.*

Proof: Again suppose $G^{\Delta_i} \equiv \mathbb{Z}_{m_1} \oplus \cdots \oplus \mathbb{Z}_{m_t}$, so $h (\in G^{\Delta_i}) \equiv (h(1), h(2), \ldots, h(t))$ where $h(i) \in \mathbb{Z}_{m_i}$. The boolean variable $S(h,i)$ for each $h \neq 0$ expresses the condition $({}^hX > X)$ as follows:

$$S(h,i) = \bigvee_{0 \le j \le m-1} \left[\left(\bigwedge_{0 \le \ell < j} X_{\{i\}}(\ell) = X_{\{i\}}(\ell^h) \right) \wedge \left(X_{\{i\}}(j) < X_{\{i\}}(j^h) \right) \right]$$

where $|\Delta_i| = m$.

For $g \in G$, we can express $^gX_{\{i\}} > X_{\{i\}}$ as

$$\bigvee_{h \in G^{\Delta_i}} \left(S(h,i) \wedge (g(i) = h) \right).$$

We express the condition that $g(i) = h$ by a system of equations $g(i,j) = h(j) \bmod m_i$ for $1 \le i \le t$. We can thus express each clause $S(h,i) \wedge (g(i) = h)$ as a system of linear equations $\mathcal{E}_2(g,i,h)$ as follows:

$$S(h,i) g(i,j) \equiv h(j) \bmod m_j \quad \text{for each } 1 \le j \le t.$$

Thus $^gX_{\{i\}} > X_{\{i\}}$ iff one of the equation systems $\mathcal{E}_2(g,i,h)$ is satisfied for some h. $\square$

$\boldsymbol{g \in G}$

Let $D = G^{\Delta_1} \oplus \cdots \oplus G^{\Delta_r}$. Using the fixed cyclic decompositions of G^{Δ_i}, we obtain a cyclic decomposition of D. For $d \in D$, let $d(i)$ be the projection of d in G^{Δ_i} and then $d(i,j)$ the projection of $d(i)$ in jth cyclic factor of G^{Δ_i}.

Now, viewing G as a subgroup of D, we let $K = G^\perp \le D$ as in Section 7.1; a generating set Q for K can be found by solving a linear system. Then, for $g \in D$, we have $g \in G$ iff $g \cdot q = 0$ for all $q \in Q$. But observe that $g \cdot q = 0$ expands to an equation of the from $\sum_{i,j} a_{ij} d(i,j) g(i,j) \equiv 0 \bmod m$ (m being the lcm of the orders of the cyclic factors). We denote the resulting system by

$\mathcal{E}_3(g)$. The number of equations is $|Q| = O(n)$ and the unknowns are the $g(i,j)$ and consistent with the variables arising in the systems $\mathcal{E}_1(g,i)$ and $\mathcal{E}_2(g,i,h)$.

Thus Equation (8) asserts that for each $1 \leq i \leq r, h \in G^{\Delta_i}$,

$$\neg\exists g: \ \mathcal{E}(i,h) \tag{10}$$

where $\mathcal{E}(i,h)$ is $\mathcal{E}_3(g) \wedge \mathcal{E}_1(g,i) \wedge \mathcal{E}_2(g,i,h)$. Thus, in effect, Equation (8) asserts the nonsolvability of each system in a collection of n linear equation-systems $\{\mathcal{E}(i,h) \mid 1 \leq i \leq r, h \in \Delta_i\}$

The number of equations in each system $\mathcal{E}(i,h)$ is $O(n)$ and each system has $O(n)$ variables $g(k,l)$'s. Each equation in $\mathcal{E}(i,h)$ is defined either modulo the size of a cyclic factor in D or m where m is the lcm of the sizes of the cyclic factors in D. Now, Lemma 7.10 implies that one can construct a boolean formula $\overline{\phi}(i,h)$ of size $O((n^3/\log n)\mu(\log n))$ which is satisfiable iff $\mathcal{E}(i,h)$ is not solvable. To be precise, $\mathcal{E}(i,h)$ does not satisfy all the hypotheses of Lemma 7.10 because some of the equations are defined modulo large integers ($> n$). However, it is easy to see that when we break $\mathcal{E}(i,h)$ into its prime-power systems, we need only consider primes that are $\leq n$. In the analysis of the final size of the formula in the lemma, this is what is significant.

Thus X is a lex-leader iff $\mathcal{E}(i,h)$ is nonsolvable for each i and each $h \in G^{\Delta_i}$, i.e., iff the following boolean formula is satisfiable

$$\bigwedge_{1 \leq i \leq r} \bigwedge_{h \in G^{\Delta_i}} \overline{\phi}(i,h) \tag{11}$$

This gives us a lex-leader formula of size $O(n^4 \mu(\log n)/\log n)$.

A Tighter Analysis

As we remarked after Lemma 7.10, the above bound for the lex-leader formula overcounts by an order of magnitude. This is because, en route to Lemma 7.10, when we break $\mathcal{E}(i,h)$ into its prime-power systems $\mathcal{E}_p$, $\mathcal{E}_p$ has far fewer variables than n (the original number of variables in $\mathcal{E}(i,h)$). We now show that more careful counting leads to a smaller estimate of the final lex-leader formula.

When the orbit constituents are written as sums of cyclics, we may have assumed each of these cyclics is of prime power order. Let N_p be the number of cyclic summands of p-power order. The following lemma is well-known:

Lemma 7.18 *The number of cyclic factors of abelian $G \leq \mathrm{Sym}(\Omega)$ is $O(n)$ where $|\Omega| = n$.*

Proof: Since G is a subdirect product of its orbit constituents $\{G^{\Delta_i}\}$, we have $|G| \leq \prod_i |G^{\Delta_i}| = \prod_i |\Delta_i| \leq 3^{n/3}$ (the last inequality follows from $\sum_i \Delta_i = n$). But the number of cyclic factors of G is clearly $O(\log |G|)$. $\Box$

Since the total number of cyclic summands is $O(n)$, $\sum_p N_p = O(n)$. When the system of equations are broken into primary-parts, then the number of essential variables in the system $\mathcal{E}_p$ for any prime p is N_p. When one consider the dual system (e.g., in going from nonsolvability to solvability) the number of equations becomes N_p.

Since there are three components in the system of equations in Equation (10), we consider what happens in each component when we pass mod p^e.

First, we consider the systems that arise from expanding "inner products": the system expressing $g \in G$ ($\mathcal{E}_3(g)$) and the system expressing g-invariance of $X_{[i-1]}$ ($\mathcal{E}_1(g,i)$). The summands expanding the inner product are of the form $g(k,l) \times x \times m/q$ (recall definition of inner product, Section 7.1) where m is the exponent (i.e., the lcm of the cyclic prime-power factors) of the relevant group, and also the modulus for the equation (congruence), and q is the order of the (k,l)-th cyclic factor. By assumption $q = p^a$ for some prime p. When the equation (congruence) is considered mod any prime p' *other than* p, this summand disappears because $m/q \equiv 0 \bmod p'$. Hence, the variable $g(k,l)$ is retained only in the systems written for the prime p.

Next, consider the equations that arise from expressing ${}^g X_{\{i\}} > X_{\{i\}}$. These are of the form

$$S(h,i)\, g(i,j) \equiv h(j) \bmod m_j$$

with j varying over the cyclic factors in the orbit and m_j the order of the corresponding cyclic factor and is therefore p^a, where again p is the prime associated with the variable $g(i,j)$. This equation cannot be included in any $\mathcal{E}_q$ for a prime q different from p because q does not divide m_j.

Lastly, we consider the equations that arise from expressing $g \in G$. As in the case for ${}^g X_{[i-1]} = X_{[i-1]}$, we retain only those coefficients in the inner product terms which appear with m/p^a in $\mathcal{E}_p$.

Hence we have the following lemma:

Lemma 7.19 *For prime p, the number of variables in $\mathcal{E}_p$ is N_p.*

Using the fact that $\sum N_p = O(n)$, this means that each $\overline{\phi}_p$ expressing nonsolvability of $\mathcal{E}_p$ is of size $O(nN_p\mu(\log p^e))$ (via Lemma 7.8), so that the formula $\overline{\phi}(i,h) = \bigvee_p \overline{\phi}_p$ expressing nonsolvability of $\mathcal{E}(i,h)$ is of size $\sum_p nN_p\mu(\log p^e) = O(n^2\mu(\log n))$. Since the number of pairs $\{(i,h)\}$ is $O(n)$, the resulting lex-leader formula, namely, Equation (11), is of size $O(n^3\mu(\log n))$. This proves Theorem 5.4.

8 Future Work

We note that a generalization to arbitrary nonabelian groups is unlikely; indeed, it is shown in [2] that testing lex-leadership is NP-hard even for the ordering scheme that we use for abelian groups. On the other hand, that same paper describes an polynomial-time algorithm for *testing* lex-leadership for a group

class that includes all solvable groups. While such a result already implies at least a polynomial-size lex-leader formula, the conversion of the known algorithm yields an unwieldy formula even in the abelian case (where it is at least 6 orders of magnitude larger than what we present herein). In subsequent work, we intend to consider the feasibility of $\Lambda(G)$ for the "good" groups of [2].

ACKNOWLEDGMENT

We are grateful to James Crawford and Matt Ginsberg of the Computational Intelligence Research Laboratory for inviting us to participate in projects that inspired this research.

References

[1] F. A. Aloul, I. L. Markov, and K. A. Sakallah. Efficient symmetry breaking for boolean satisfiability. In *Proc. Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.

[2] L. Babai and E. M. Luks. Canonical labeling of graphs. In *Proc. 15th ACM Symp. on Theory of Computing*, pages 171–183, 1983.

[3] R. Backofen and S. Wilf. Excluding symmetries in constraint-based search. In *Proceedings of 5 th International Conference on Principle and Practice of Constraint Programming (CP'99)*, volume 1713 of *Lecture Notes in Computer Science*, pages 73–87. Springer-Verlag, 1999.

[4] B. Benhamou. Study of symmetry in constraint satisfaction problems. In *Principles and Practice of Constraint Programming (PPCP-94)*, 1994.

[5] C. A. Brown, L. Finkelstein, and P. W. Purdom. Backtrack searching in the presence of symmetry. In T. Mora, editor, *Applied algebra, algebraic algorithms and error correcting codes, 6th international conference*, pages 99–110. Springer-Verlag, 1988.

[6] J. Crawford. A theoretical analysis of reasoning by symmetry in first-order logic (extended abstract). In *Workshop notes, AAAI-92 workshop on tractable reasoning*, pages 17–22, 1992.

[7] J. Crawford, M. Ginsberg, E. M. Luks, and A. Roy. Symmetry breaking predicates for search problems. In *Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning (KR '96)*, pages 148–159, 1996.

[8] J. D. Dixon and B. Mortimer. *Permutation groups.* Springer-Verlag, New York, 1996.

[9] K. Engel. *Sperner Theory*, volume 65 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1997.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W. H. Freeman and Company, New York, 1979.

[11] Jvz. Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.

[12] T. Grüner, R. Laue, and M. Meringer. Algorithms for group actions applied to graph generation. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 28:113–122, 1997.

[13] M. Hall. *The Theory of Gxroups.* Chelsea Publishing Company, New York, second edition, 1976.

[14] D. Joslin and A. Roy. Exploiting symmetries in lifted csps. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 197–203. American Association for Artificial Intelligence (AAAI), 1997.

[15] G. Katona and J. Srivastava. Minimal 2-coverings of a finite affine space based on GF(2). *J. Statist. Plann. Inference*, 8(3):375–388, 1983.

[16] J. Köbler, U. Schöning, and J. Torán. *The graph isomorphism problem: its structural complexity.* Birkhäuser Boston Inc., Boston, MA, 1993.

[17] C.W.H. Lam, L.H. Thiel, and S. Swiercz. The non-existence of finite projective planes of order 10. *Canadian Journal of Math*, XLI:1117–1123, 1989.

[18] S. Lang. *Algebra.* Addison-Wesley Publishing Company, 1999.

[19] E. M. Luks. Permutation groups and polynomial-time computation. In W. M. Kantor L. Finkelstein, editor, *Groups and Computation, Workshop on Groups and Computation*, volume 11 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 139–175, 1993.

[20] E. M. Luks and A. Roy. In preparation.

[21] D. Miklós. Linear binary codes with intersection properties. *Discrete Appl. Math.*, 9(2):187–196, 1984.

[22] J. Savage. *Models of Computation, Exploring the Power of Computing.* Addison-Wesley, 1998.

[23] H. N. Shapiro. *Introduction to the Theory of Numbers.* Wiley-Interscience, 1983.