

SYMMETRY BREAKING AND FAULT TOLERANCE IN BOOLEAN  
SATISFIABILITY

by

AMITABHA ROY

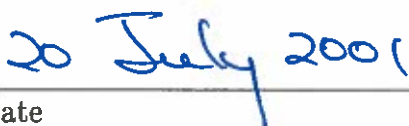
A DISSERTATION

Presented to the Department of Computer  
and Information Science  
and the Graduate School of the University of Oregon  
in partial fulfillment of the requirements  
for the degree of  
Doctor of Philosophy

August 2001

"Symmetry Breaking and Fault Tolerance in Boolean Satisfiability," a dissertation prepared by Amitabha Roy in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science. This dissertation has been approved and accepted by:

  
Chair of the Examining Committee

  
Date

Committee in charge:      Dr. Eugene M. Luks, Chair  
   Dr. Andrzej Proskurowski  
   Dr. Christopher B. Wilson  
   Dr. William Kantor

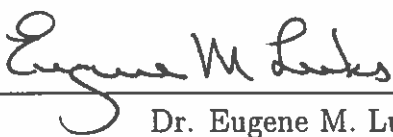
Accepted by:

  
Vice Provost and Dean of the Graduate School

© 2001 Amitabha Roy

An Abstract of the Dissertation of  
Amitabha Roy for the degree of Doctor of Philosophy  
in the Department of Computer and Information Science  
to be taken August 2001

Title: SYMMETRY BREAKING AND FAULT TOLERANCE IN  
BOOLEAN SATISFIABILITY

Approved:   
Dr. Eugene M. Luks

Use of inherent symmetries to speed computation has been an effective technique in many constraint satisfaction problems. Typically this involves modifying a search algorithm to exploit the symmetry. As an alternative, we study a general scheme wherein symmetries are used to modify the input problem itself. Thus instead of having to reformulate each advance in search technology, we add a “symmetry breaking” formula that can be used as a preprocessor to existing or future constraint solvers.

A *symmetry breaking formula* is a boolean formula that is satisfied by exactly one member from each set of symmetric points in the original search space. For example, we choose this member to be the lexicographic leader in the orbit of assignments under the action of a permutation group on the input variables.

A main computational hurdle is that it is often intractable to generate the entire lex leader predicate. Indeed, we prove the existence of groups for which the

*smallest* lex leader predicate is of exponential size. These intractable examples suggest consideration of *Sperner families* of sets whose incidence vectors form a subspace of  $Z_2^n$ . However we show how to construct succinct lex leader formulas for abelian groups and groups with bounded orbit projections (and hence also the groups corresponding to Sperner families). Our formulas exploit the polynomial time algorithmic machinery developed to solve the lex leader problem for “good groups”, e.g., solvable groups or more generally for groups with bounded non-cyclic composition factors.

A dual goal to efficiency of search is robustness of solutions. We desire that the solutions produced not be “brittle”: an optimal solution is undesirable if any unforeseen event makes it untenable (e.g. a resource suddenly becoming unavailable in a resource allocation problem). To model this concept of fault tolerance, we introduce the notion of  $\delta$ models: these are satisfying assignments of a boolean formula for which any small alteration, such as a single bit flip, can be repaired by another small alteration, yielding a nearby satisfying assignment. We study computational problems associated with  $\delta$ models and some combinatorial generalizations thereof.

## CURRICULUM VITA

NAME OF THE AUTHOR: Amitabha Roy

PLACE OF BIRTH: Calcutta

DATE OF BIRTH: Sep 29, 1971

## GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon.  
Indian Institute of Technology, Kanpur.

## DEGREES AWARDED:

- Doctor of Philosophy in Computer and Information Science,  
University of Oregon, 2001.
- Master of Science in Computer and Information Science,  
University of Oregon, 1996.
- Bachelor of Technology in Computer Science and Engineering,  
Indian Institute of Technology (1994).

## AREAS OF SPECIAL INTEREST:

Computational Algebra  
Combinatorics

## PROFESSIONAL EXPERIENCE:

Teaching and Research Assistant, Department of Computer and  
Information Science, University of Oregon, Eugene, 1994-2001.

## GRANTS:

Partially supported by National Science Foundation grant (CCR9820945), 1999-2001, to Dr. Eugene M. Luks.

## PUBLICATIONS:

James Crawford, Matthew Ginsberg, E. M. Luks and Amitabha Roy. Symmetry Breaking Predicates for Search Problems. In *Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning*, (KR '96), 1996, pp 148-159.

Matthewn Ginsberg, Andrew Parkes and Amitabha Roy. Supermodels and Robustness. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence (AAAI), 1998, pp 334-339.

David Joslin and Amitabha Roy. Exploiting Symmetries in lifted CSPs. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence (AAAI), 1997, pp 197-203.

Amitabha Roy and Christopher Wilson. Supermodels and Closed Sets. In *Electronic Colloquium on Computational Complexity (ECCC)*, 1999.

## ACKNOWLEDGEMENTS

I want to thank my advisor, Eugene M. Luks, for the many hours that he spent explaining problems and ideas and for being my mentor for all these years. In addition, thanks are due to Christopher Wilson for being a good friend and collaborator. I want to express appreciation to the members of the Computational Intelligence Research Laboratory for their support and encouragement, especially to Matthew Ginsberg, Andrew Parkes, David Etherington, James Crawford and David Joslin. My friends made my graduate life so pleasant, to name a few: Andrzej Proskurowski, Miley Semmelroth, John (and Janet) Fiskio-Lasseter, Takunari Miyazaki, Kevin Glass, Aaron Fabbri, Yolanda Reimer, Ferenc Rákóczi, Dennis Gray.

This work was partially supported by a National Science Foundation grant (CCR9820945) to Dr. Eugene M. Luks.

Last but not the least, I want to thank my parents, my wife Amrita, Charubrata Goswami and Maitrayee Majumdar for their unwavering love and support.



DEDICATION

For Baba, Ma and Bibi.

## TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION . . . . .	1
1. Motivation . . . . .	1
2. Search and Symmetries . . . . .	2
3. Fault Tolerance . . . . .	6
II. SYMMETRY BREAKING FORMULAS . . . . .	8
1. Definitions and Notations . . . . .	8
2. Lex-leader Formulas – Definitions . . . . .	10
3. Statement of Results . . . . .	15
4. The Algorithmic Formula . . . . .	17
5. Exponential Lower Bounds for Lex-Leader Formulas . . . . .	20
6. Sperner Subspaces . . . . .	30
7. Polynomial Size Lex-Leader Formulas for Abelian Groups . . . . .	49
8. Lex-Leader Formulas for $\mathcal{P}_d$ Groups . . . . .	69
III. FAULT TOLERANCE IN BOOLEAN SATISFIABILITY . . . . .	75
1. Definitions and Notations . . . . .	75
2. Complexity of Finding $\delta$ Models . . . . .	77
3. Finding $\delta$ Models for Restricted Boolean Formulas . . . . .	80
4. Stable Sets: Definitions and Notations . . . . .	95
5. Extremal Properties of Stable Sets . . . . .	97
6. Examples of Stable Sets . . . . .	116
7. Summary and Future Work . . . . .	123
BIBLIOGRAPHY . . . . .	124

## LIST OF FIGURES

Figure	Page
1. Gadget for 2-SAT . . . . .	90
2. Sparse Stable Family of Size 10 for $n = 6$ . . . . .	120
3. Sparse Stable Family of Size 32 for $n = 8$ . . . . .	121

## LIST OF TABLES

Table	Page
1. Boolean Formulas for Permutation Groups . . . . .	77
2. Complexity of Finding $\delta$ models . . . . .	98
3. Upper and Lower Bounds for Stable Families . . . . .	126

## CHAPTER I

### INTRODUCTION

#### 1. Motivation

This thesis studies methods to develop efficient algorithms to solve constraint satisfaction problems (CSPs). Specifically, it considers two distinct but related problems:

- the use of symmetry in search
- fault tolerance in CSPs.

Both problems are inspired by artificial intelligence applications, especially scheduling and planning problems.

Use of inherent symmetry to speed computation has been an effective technique in many constraint satisfaction problems. Typically this involves modifying the search algorithm to exploit the symmetry present in the input. Since this forces us to tie symmetry exploitation to the specific search algorithm, this approach would require us to reformulate each advance in search technology. As an alternative, Crawford et. al. [13] developed the notion of symmetry breaking formulas, a novel scheme wherein symmetries are exploited by changing the input and not the search algorithm. A symmetry breaking formula is an extra set of constraints that is added to the input before the search algorithm starts. Since this is essentially a preprocessing step, this method can be used as a front-end to existing or future constraint solvers, thus

avoiding the need to re-engineer the search algorithm itself. This method is described briefly in Section 2 of this chapter and the technical details of the particular problem we address are in Chapter 2.

While speed of computation is an important factor, the *nature* of solutions produced is also of concern. Sometimes a solution may be brittle i.e in an optimal solution for a scheduling problem, it might be crucial that a certain task finish by a fixed deadline. In brittle solutions, unforeseen obstacles (e.g. a task failing to finish by a deadline) can be catastrophic to optimality. Ginsberg et. al. [18] formalized a notion of “robust” solutions which allow for recovery from such unforeseen events. These robust solutions allowed for small perturbances in the optimal solution, but also allowed quick recovery from those perturbances. We study the computational complexity of finding these solutions, showing that it is NP-hard to find them in general. We also exhibit instances where it is possible to find these robust solutions in polynomial time. We study a class of combinatorial structures (stable sets) that arise naturally in this context. We study their extremal properties, prove lower and upper bounds on the maximal sizes of these structures and give explicit constructions. This problem is described briefly in Section 3 of this chapter and the technical details are in Chapter 3.

## 2. Search and Symmetries

Many computational problems have symmetries. For example, a scheduling problem that attempts to schedule millions of tasks with deadlines could have many tasks that are identical. Algorithms to exploit symmetries have been used to solve some important open problems, famous examples being the non-existence of projective planes of order 10 [26] and the four-color theorem [2]. From a computational

perspective, exploiting symmetries has become a standard tool in solving large search problems [25]. Since symmetries arise as permutations which preserve properties of the input, techniques from computational group theory can be used to develop efficient search algorithms.

Abstractly defined, a search problem consists of a large (usually exponentially large) collection of possibilities, the *search space*, and a predicate. The task of the search algorithm is to find a point in the search space that satisfies the predicate. Search problems arise naturally in many areas of artificial intelligence, operations research and mathematics.

The use of symmetries in search problems is conceptually simple. If several points in the search-space are related by a symmetry then we never want to visit more than one of them. With regard to taking computational advantage of the symmetries, past work has focused on specialized search algorithms that are guaranteed to examine only a single member of each symmetry class [10]. Unfortunately, this makes it difficult to combine symmetry exploitation with other work in satisfiability or constraint satisfaction, such as flexible backtracking schemes [17, 19] or non-systematic approaches [32, 39]. Given the rapid progress in search techniques generally over the past few years, tying symmetry exploitation to a specific search algorithm seems premature.

The approach we take here is different. Rather than modifying the search algorithm to use symmetries, we will use symmetries to modify (and hopefully simplify) the problem being solved. In tic-tac-toe, for example, we can require that the first move be in the middle, the upper left hand corner, or the upper middle (since doing this will not change our analysis of the game in any interesting way). In general, our

approach will be to add additional constraints, *symmetry-breaking formulas*, that are satisfied by exactly one member of each set of symmetric points in the search space. Since these constraints will be in the same language as the original problem (propositional satisfiability for purposes of this thesis) we can run the symmetry detection and utilization algorithm as a preprocessor to any satisfiability checking algorithm.

This approach has two fundamental obstacles. The first obstacle is that there is no known polynomial-time algorithm for detecting all the symmetries of the input. This problem is equivalent to the *graph isomorphism* problem which asks the following question: given two graphs, is there a bijection between the vertices which preserves adjacencies? This problem is also not known to be NP-complete, though there is evidence that it is probably not so [24]. Nevertheless, graph isomorphism is rarely difficult in practice, as has been profoundly demonstrated by the efficient *nauty* system [30]. Furthermore, it has been shown that, on average, graph isomorphism is in linear time using even naive methods [3] and in polynomial time for a wide class of graphs [28, 4]. The second obstacle is that even after detection is complete, computing the full symmetry-breaking formula appears to be intractable.

Our goal in Chapter 2 of this thesis is to explore the second obstacle. In particular, we will be interested in permutation groups for which we can write a polynomial-size symmetry breaking formula. We make some assumptions on the kind of symmetry breaking formula we consider. We define an ordering of the underlying set that the group of symmetries acts on and use that to define a lexicographic (dictionary) order on the set of all possible solutions in the search space. We consider those symmetry breaking formulas which are true of only lexically largest element from a set of symmetrical points in the search space. We call this *the lex-leader formula*. We show that



for groups with a very simple structure (elementary abelian 2-groups with orbits of size 2) naive lex-leader formulas are of exponential size. The naive lex-leader formula uses exactly the same number of variables as there are points in the permutation domain. This exponential size is because of a combinatorial bottleneck which we can formulate in terms of lattices and anti-chains. This has led us to consider a class of combinatorial objects, *Sperner spaces*, a generalization of Sperner families in extremal set theory [15, 42], whose structure is responsible for this exponential lower bound.

However we show how to construct succinct lex-leader formulas for abelian groups and groups with polynomially bounded orbit projections (and hence also the groups corresponding to Sperner families). We can achieve polynomial-size formulas for these groups when we are allowed to use a small number of extra variables in addition to those which represent the permutation domain. Our formulas exploit the polynomial-time algorithmic machinery developed to solve the lex-leader problem for “good groups” (e.g. solvable groups or more generally for groups with bounded non-cyclic composition factors) by Luks and Babai[6]. The choice of ordering of the permutation domain is also significant in our ability to write polynomial-size symmetry breakers (a situation reflected in the algorithmic setting: with arbitrary orderings, finding lex-leaders is NP-hard even for abelian 2-groups [6]).

Polynomial-time algorithms for good groups (assuming a certain ordering of the permutation domain) imply that there is a polynomial-size lex-leader formula for these groups. This is a consequence of Cook’s theorem [16] which guarantees, by asserting that SAT is NP-complete, the existence of a “small” boolean formula equivalent to every “yes” instance of a decision problem in NP (and hence also for every problem which admits a polynomial time solution). This approach to building

a lex-leader formula is too general and too unwieldy: the formulas depend on the algorithm used to solve the lex leader formula and they are typically larger than the formulas we obtain. But as a consequence of the existence of efficient algorithms for good groups [6], it might be possible to generalize our constructions to these groups.

### 3. Fault Tolerance

The concept of  $\delta$ models, introduced in [18] as “supermodels”, formalizes a notion of fault tolerant satisfying assignments to boolean formulas. In this thesis, we study the problem of identifying these  $\delta$ models and generalize this notion of fault tolerance.

The motivation for studying  $\delta$ models in the artificial intelligence/planning community was to build search algorithms finding robust solutions to problems (typically in scheduling or planning domains). These solutions have the property that if an expected resource is suddenly unavailable, then a minimal modification to the solution produces an equally acceptable alternative. Recently, this idea has been used in [7].

This notion of a  $\delta$ model is similar to that of the *sensitivity* of boolean functions, see e.g. [8], [27]. Roughly speaking, the sensitivity of a function is the average number of input bits whose flip will change the value of the function. For a  $\delta$ model, however, we require that if a bit flip changes the outcome of the formula, there must be some other way to restore the original outcome. Thus, a formula with a  $\delta$ model could have either low or high sensitivity.

More formally, a  $\delta$ model of a boolean formula  $F$  is a satisfying assignment  $\alpha$  of  $F$ ,  $F(\alpha) = 1$ , such that for every  $i$ , if we negate the  $i$ th bit of  $\alpha$ , there is another bit  $j \neq i$  of  $\alpha$  which we can negate to get another satisfying assignment (we call satisfying assignments *models* of boolean formulas). That is, if  $\delta_i(\alpha)$  is the function which negates the  $i$ th bit of  $\alpha$ , then  $(\forall i)(\exists j \neq i)F(\delta_j(\delta_i(\alpha))) = 1$ . In Chapter 3,

we study the complexity of finding  $\delta$ models for restricted classes of formulas. It was shown in [18] that determining whether a formula has a  $\delta$ model is NP-complete. We restrict the problem here to formulas which are instances 2-SAT, Horn SAT, and Affine SAT (recall that in all three cases membership testing can be done in polynomial time, see Shaefer's dichotomy theorem [37]).

We extend the notion of a  $\delta$ model to that where a model remains a model after an arbitrary sequence of  $k$  single breaks and single repairs. This leads us to investigate arbitrary degrees of fault tolerant models,  $\delta^*$ models (which are models after any sequence of breaks and single repairs) being of particular interest. These models lead us to study the combinatorics of *stable families*: these are collection of subsets such that for each set in the family and for each break to that set there must be a repair which yields another member of the family.

We study the extremal structure of stable families. One useful restriction is when we force a break to have exactly one repair. We refer to such stable families as "sparse". Formally a sparse stable set is a family of subsets of a set  $[n]$ , such that for all  $X \in \mathcal{F}$ ,  $\forall i \in [n], \exists! j \neq i, \delta_j(\delta_i(X)) \in \mathcal{F}$ . Here  $\delta_i(X) = X \Delta \{i\}$ , which is exactly equivalent to flipping the  $i$ -th bit in the incidence vector of  $X$ .

We shall that if  $\mathcal{F}$  is sparse stable then  $2^{n/2} \leq |\mathcal{F}| \leq \frac{2^{n+1}}{n+2}$ . Constructing sparse stable sets which achieve the lower bound is easy. So far, there is a gap between the largest sparse stable families we can construct and the upper bound. We also consider the sizes of the largest minimal sparse stable sets (a minimal sparse stable set does not contain a smaller sparse stable set). Using exhaustive search, we prove that there are minimal sparse stable set of size  $80^{n/10}$ .

## CHAPTER II

## SYMMETRY BREAKING FORMULAS

1. Definitions and Notations

Let  $G$  be a group. We write  $H \leq G$  when  $H$  is a subgroup of  $G$ . If  $H \leq G$ , then a right transversal of  $H$  in  $G$  is a complete set of right coset representatives of  $H$  in  $G$ . The group consisting of all permutations of a set  $\Omega$  is called the symmetric group, denoted by  $\text{Sym}(\Omega)$ .  $G$  is said to act on  $\Omega$  if there is a homomorphism  $\phi : G \rightarrow \text{Sym}(\Omega)$ . Let  $\omega \in \Omega$  and  $g \in G$ , then  $\omega^g$  is the image of  $\omega$  under  $\phi(g)$ . Also  $\omega^G = \{\omega^g \mid g \in G\}$  is the orbit of  $G$  that contains  $\omega$ . A group is said to be transitive if  $\omega^G = \Omega$ . The point stabilizer of  $\omega$  is the subgroup  $G_\omega = \{g \in G \mid \omega^g = \omega\}$ . The pointwise stabilizer of  $\Delta \subset \Omega$  is  $G_{(\Delta)} = \bigcap_{\delta \in \Delta} G_\delta$ . When  $\Omega$  is ordered as  $\omega_1, \omega_2, \dots, \omega_n$ , then  $\Omega_i = \{\omega_1, \omega_2, \dots, \omega_i\}$  and  $G_i = G_{(\Omega_i)}$ . Let  $\Delta$  be an orbit of  $G$  on  $\Omega$ . For  $g \in G$ ,  $g^\Delta$  is the restriction of  $g$  on  $\Delta$ . The orbit constituent  $G^\Delta = \{g^\Delta \mid g \in G\}$  is the projection of  $G$  onto  $\Delta$ . A  $\mathcal{P}_d$  group is a group  $G \leq \text{Sym}(\Omega)$ , where  $|\Omega| = n$  such that the size of each orbit constituent is at most  $n^d$ . A group  $G$  is said to act regularly on  $\Omega$  if  $G_\omega = 1$  for all  $\omega \in \Omega$ .

Groups are input (and output) via generators. We write  $G = \langle X \rangle$  when the set  $X \subset G$  generates  $G$ . Subgroups of  $\text{Sym}(\Omega)$  have succinct descriptions in terms of generators : they have generating sets of size  $O(|\Omega|)$  [14]. A very useful data structure for permutation groups is a strong generating set (SGS), first introduced by Sims[41].

Given a chain

$$G = G^0 \geq G^1 \geq G^2 \geq \dots \geq G^m = 1$$

of subgroups of  $G$ , an SGS with respect to this chain is a set  $T \subset G$  such that  $\langle T \cap G^i \rangle = G^i$ , for each  $i$ . We shall use the “point stabilizer” series as our subgroup chain, i.e.,  $G^i = G_i$  is the subgroup of  $G$  that fixes the first  $i$  points of  $\Omega$ . Then an example of an SGS with respect to this chain is the set  $R = \cup_{i=1}^m R_i$  where  $R_i$  is a complete right transversal of  $G_i$  in  $G_{i-1}$ . A permutation  $\pi$  is said to sift through this chain if it can be expressed as product  $r_m r_{m-1} \dots r_1$  where  $r_i \in R_i$ .

We refer to any standard text (e.g [21]) for basic facts about groups. For permutation groups, we refer to [45] and [14].

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  $X : L \rightarrow \{0, 1\}$ . 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 [16].

Let  $T$  be a theory. A set of clauses  $\mathcal{C} = \{C_1, C_2, \dots, C_r\}$  in  $T$  is said to prune  $\mathcal{C}' = \{C'_1, C'_2, \dots, C'_s\}$  in  $T$  if any model of  $\wedge_{C \in \mathcal{C}} C$  is a model of  $\wedge_{C' \in \mathcal{C}'} C'$ . A sub-collection of clauses  $\mathcal{C}'$  in  $T$  is said to be prunable if there exists a set of clauses

$\mathcal{C} = \{C_1, C_2, \dots, C_r\}$  in  $T$  which prunes  $C'$ ;  $C'$  is non-prunable otherwise. To avoid trivialities, we require that if  $\mathcal{C}$  prunes  $C'$ , then  $C' \notin \mathcal{C}$ .

## 2. Lex-leader Formulas – Definitions

In this section, we formalize the notion of lex-leader formulas in the context of a permutation group acting on a set of points (Subsection 2.1). We also discuss how lex-leader formulas can be used to augment boolean theories so as to break symmetries in the input as preprocessing step before search (Subsection 2.2).

### 2.1: Lex-Leader Formulas for Permutation Groups

Let  $\Omega$  denote the set  $\{1, 2, \dots, n\}$  and  $G \leq \text{Sym}(\Omega)$ . Let  $2^\Omega$  denote the set of functions from  $\Omega$  to  $\{0, 1\}$ .  $G$  acts on  $2^\Omega$  via  $X \mapsto {}^gX$  for  $g \in G$ ,  $X \in 2^\Omega$  where  $({}^gX)(i) = X(i^g)$ .<sup>1</sup> Under the action of  $G$ ,  $2^\Omega$  breaks up into orbits under the action of  $G$ .

There is a natural lexicographic order in  $2^\Omega$ :  $X < Y$  if  $X \neq Y$  and  $X(i) < Y(i)$  for the least  $i$  such that  $X(i) \neq Y(i)$ .

Our goal is to write a formula in propositional logic that is true of only one member from each orbit of functions, which we call a *canonical member*. In this thesis, 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,  $Y < X$ . Formally, a lex-leader formula for  $G$  is a boolean formula  $\phi_L(G)$  defined over  $n$  variables, whose models are lex-leaders in their orbits. Frequently, we will define  $\phi_L(G)$  over a larger set of

---

<sup>1</sup>It is natural to write this as a “left action”, e.g., we have  ${}^{g_1 g_2}X = {}^{g_1}({}^{g_2}X)$ , whereas expressing the image of  $X$  under  $g_1$  by  $X^{g_1}$  would lead to the awkward relation  $X^{g_1 g_2} = (X^{g_2})^{g_1}$ .

variables and require that the projection of its models in a fixed set of  $n$  coordinates are lex-leaders in their  $G$ -orbits. It will be clear from the context when we use these extra variables.

For any  $X \in 2^\Omega$  define  $X_i$  to be the restriction of  $X$  to  $\Omega_i$ , i.e.,  $X_i$  is an  $i$ -tuple consisting of the first  $i$  coordinates of  $X$ . We will write  $\text{Fix}(g, X, i)$  to mean the boolean formula  $({}^g X)_i = X_i$ , i.e., the formula  $[(X(1) = X(1^g))] \wedge [X(2) = X(2^g)] \wedge \dots \wedge [X(i) = X(i^g)]$  (we substitute  $X(i^g)$  for  $({}^g X)(i)$ ). We use  $X(1), X(2), \dots, X(n)$  as variable names for formulas without any confusion with the function  $X$  evaluated at points  $1, 2, \dots, n$ .

We write  $\text{Geq}(g, X, i)$  to mean the formula  $X(i) \geq X(i^g)$ . Observe that  $X(i) \geq X(i^g)$  is just a mnemonic for the boolean expression  $X(i^g) \rightarrow X(i)$ .

We now show how to write a very naive lex-leader formula. Let  $g \in G$  and  $X \in 2^\Omega$ . Consider the following formula

$$\bigwedge_{1 \leq i \leq n} \text{Fix}(g, X, i-1) \rightarrow \text{Geq}(g, X, i) \quad (\text{II.1})$$

By our definition of lexical order, any  $X$  which satisfies the Equation (II.1) has the property that  $X \geq {}^g X$ . Thus the conjunction of all the formulas associated with all  $g \in G$ , namely,

$$\bigwedge_{g \in G} \bigwedge_{i=1}^n \text{Fix}(g, X, i-1) \rightarrow \text{Geq}(g, X, i) \quad (\text{II.2})$$

will be true of only the lexical leader of each orbit of functions.

We rewrite Equation II.2 and define the lex-leader formula  $LL(G)$  as follows:

$$LL(G) = \bigwedge_{g \in G} \bigwedge_{i=1}^n C(g, i) \quad (\text{II.3})$$

where  $C(g, i) = \text{Fix}(g, X, i-1) \rightarrow \text{Geq}(g, X, i)$ .

Equation (II.3) could have duplicate clauses. For example, consider the case when  $G = S_3$ . 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 II.3. Notice that the group elements  $(1\ 2)$  and  $(1\ 2\ 3)$  both belong to the same right coset of  $G_1$ . This intuition allows us to eliminate duplicate clauses: for each  $i$ , we include clauses  $C(g, i)$  for each coset representative  $g$  of  $G/G_i$ . This approach can still leave us with  $\sum_{i=0}^{n-1} |G/G_{i+1}|$  clauses (which could be of exponential size).

The question is: can we prune  $LL(G)$  further? For example, the clause  $C((1, 3), 1) = (X(1) \geq X(3))$  prunes the clause

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

While  $LL(G)$  might be of exponential size in the input (recall that permutation groups are input via a small set of generators), one might hope to prune it to polynomial size by removing such redundant clauses. But we shall see that this is not the case (Theorem 3.1 (i)).

## 2.2: Symmetry Breaking Formulas

Let  $T$  be a theory over the  $n$  variable set  $L$ . Define  $\neg L = \{\neg x \mid x \in L\}$  to be the set of negated literals. Let  $\bar{L} = L \cup \neg L$ . Since we require theories to be in conjunctive normal form, we can write  $T$  as a set  $\{C_i \mid 1 \leq i \leq m\}$  where each  $C_i$  is



a disjunction of literals, represented as a set  $\{l_{ij} \mid 1 \leq j \leq c_i, l_{ij} \in \bar{L}\}$  where  $c_i = |C_i|$ .

The group  $\text{Sym}(L)$  has a natural action on the (infinite) set of theories  $T$  over  $L$ . We first extend the action of  $\text{Sym}(L)$  on  $L$  to the set of literals  $\bar{L} = L \cup \neg L$  as follows: if  $\neg x \in \bar{L}$  and  $g \in \text{Sym}(L)$ , then  $(\neg x)^g = \neg(x^g)$ . This naturally defines an action on a clause: if  $C = \{l_1, l_2, \dots, l_r\}$  where  $l_i \in \bar{L}$  then  $C^g = \{l_1^g, l_2^g, \dots, l_r^g\}$ . Now the action on the set of theories is obvious: if  $T = \{C_1, C_2, \dots, C_m\}$  then  $T^g = \{C_1^g, C_2^g, \dots, C_m^g\}$ .

As discussed,  $\text{Sym}(L)$  has a natural action on the set of assignments of a theory. If  $X$  is a truth assignment to variables in  $L$ , then  $g \in \text{Sym}(L)$  maps  $X$  to  ${}^gX$  where  $({}^gX)(v) = X(v^g)$ .

A permutation  $g \in \text{Sym}(L)$  is an *automorphism* (also called a symmetry) of the theory  $T$  if  $T^g = T$ . Note that since we require clauses to be disjunctions of distinct literals, an automorphism is a well-defined structural (rather than just a logical) equivalence of two theories. Let  $G = \text{AUT}(T) \leq \text{Sym}(L)$  denote the subgroup of all automorphisms of  $T$ . Hence, if  $g \in \text{AUT}(T)$ ,  $X(T^g) = ({}^gX)(T)$ .

Thus, we have the immediate consequence that any symmetry of  $T$  maps models of  $T$  to models of  $T$ , and non-models of  $T$  to non-models:

**Proposition 2.1.** Let  $T$  be a theory over  $L$ ,  $g \in \text{AUT}(T)$ , and  $X$  a truth assignment of  $L$ . Then  $X$  is a model of  $T$  iff  ${}^gX$  is a model of  $T$ .

The group  $\text{AUT}(T)$  induces an equivalence relation on the set of truth assignments of  $L$ , wherein  $X$  is equivalent to  $Y$  if  $Y = {}^gX$  for some  $g \in \text{AUT}(T)$ ; thus, the equivalence classes are precisely the *orbits* of  $\text{AUT}(T)$  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 can be used to reduce search: we can determine

whether  $T$  has a model by visiting each equivalence class rather than visiting each truth assignment.

More generally, a symmetry breaking formula is chosen so that it is true of exactly one element in each orbit of assignments to variables  $L$  in a theory  $T$ . We illustrate this with an example: let  $T$  be the theory  $a \vee \bar{c}$ ,  $b \vee \bar{c}$ ,  $a \vee b \vee c$ ,  $\bar{a} \vee \bar{b}$ . It is clear that  $(a \ b) \in \text{AUT}(T)$ . 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). As required by Proposition 2.1, 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. 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.<sup>2</sup> Equation II.3 is an example of such a formula.

The basic idea then is to generate a symmetry breaking formula (e.g.,  $\phi_L(\text{AUT}(T))$ ) and augment the original theory with this formula (e.g., build the theory  $T' = T \wedge \phi_L(\text{AUT}(T))$ ). Proposition 2.1 guarantees that  $T'$  is satisfiable iff  $T$  is satisfiable. Moreover models of  $T'$  are also models of  $T$ , each model being the (unique) lex-leader from its own orbit of assignments. One would expect that this would guide the search algorithm used to find models to automatically search non-symmetrical regions of the search space, thus improving efficiency. This observation has been borne out experimentally as described in [13, 22].

---

<sup>2</sup>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.

The problem of finding generators for the automorphism group of  $T$  is an interesting problem in its own right and one which we don't address in this thesis. This problem is equivalent to the *graph isomorphism problem* (ISO) [12]. The complexity of ISO is one of the outstanding problems in computer science: there are no polynomial time algorithms known to solve ISO and it is also not known to be NP-complete (though there is evidence that it is not NP-complete [24]). The problem of finding automorphisms of theories is dealt with in [13] which also studies the effect of augmenting theories with lex-leader formulas on the performance of search algorithms.

The problem that we do address is the complexity of generating “small” lex-leader formulas when the group of symmetries is already known. In this chapter, we prove exponential lower bounds for  $LL(G)$  even when  $G$  is restricted to groups with orbits of size 2 (which forces  $G$  to be elementary abelian 2-group or in other words, a vector space over  $GF(2)$ ). However we also show that if we are allowed to add a polynomial number of extra variables, we can write a polynomial size lex-leader formula  $\phi_L(G)$  for a large class of groups which also include these elementary abelian 2-groups. We summarize the main results of this chapter in the next section.

### 3. Statement of Results

Our goal is to study the size of  $\phi_L(G)$  and  $LL(G)$  (Equation II.3) for various classes of groups. Observe that our goal is not just to prove an exponential lower bound to  $LL(G)$  – after all this formula could have numerous identical clauses and also have numerous redundant prunable clauses. Our aim to prove lower bounds is more ambitious – we want to prove a lower bound on the size of the minimal equivalent formula in  $LL(G)$ , i.e., the number of non-prunable clauses.

Note that any definition of lexical order on the set of assignments presupposes an ordering of the underlying set that  $G$  acts on. It is possible that some orderings may lead to a lex-leader formula (as prescribed by II.3) with no small equivalent formulae, whereas a different ordering might lead to a more tractable lex-leader formula, though we do not have real examples to exhibit this (see end of Section 5 for a discussion of effect of order). Then any theorem on lex-leader formula assumes an implicit understanding of the order of the underlying set.

Let  $d$  be a fixed constant. Recall that a  $\mathcal{P}_d$  group is a group  $G \leq \text{Sym}(\Omega)$ , where  $|\Omega| = n$  and the size of every orbit constituent of  $G$  is at most  $n^d$ .

We now summarize our results in the next theorem, whose proof is delegated to subsequent sections.

Theorem 3.1.

- (i) There exist groups  $G \leq \text{Sym}(\Omega)$  for which the number of non-prunable clauses in  $\text{LL}(G)$  is  $c^n$  for all possible orderings of  $\Omega$ , where  $c$  is a constant  $> 1$  and  $n = |\Omega|$ . However, for these groups, there is a lex-leader formula  $\phi_L(G)$ , of size  $O(n^3)$  for any ordering of  $\Omega$ , that uses  $O(n^3)$  additional variables.
- (ii) Let  $G \leq \text{Sym}(\Omega)$  be an abelian group. Then one can find an ordering of  $\Omega$  in polynomial time such that there is a lex-leader formula  $\phi_L(G)$  of size  $O(n^5 \log \log n \log \log \log n)$  ( $|\Omega| = n$ ) defined over a polynomial (in  $n$ ) number of variables.
- (ii) Let  $G = \langle X \rangle \leq \text{Sym}(\Omega)$  be a  $\mathcal{P}_d$  group. Then one can reorder  $\Omega$  in polynomial time such that there exists a lex-leader formula  $\phi_L(G)$  of size  $O(d n^{2d+5} \log n)$ , where  $|\Omega| = n$ .

Theorem 3.1 (i) is proved in Section 5 (Theorem 5.5 and Corollary 5.7) and (Theorem 7.5). Part (ii) is proved in Theorem 7.19. Part (iii) is proved in Theorem 8.1 (i).

Observe that part (iii) subsumes part (ii), since abelian groups are in  $\mathcal{P}_1$ . However the bound for abelian groups obtained by (iii) is much worse than what we obtain from (ii).

#### 4. The Algorithmic Formula

We now consider an alternative approach to Theorem 3.1 for writing short lex-leader formulas for abelian and  $\mathcal{P}_d$  groups and in general for a broader class of groups.

First we define the following decision question:

##### Lex-Leader

*Input:*  $G = \langle S \rangle \leq \text{Sym}(\Omega)$  and  $X \in 2^\Omega$  (input as an  $n$ -bit string).

*Question:* Is  $X$  the lex-leader in  $X^G$ ?

Recall that  $X^G = \{gX \mid g \in G\}$  is the orbit of  $G$  that contains  $X$ .

Following [6], define the composition width of  $G$ , denoted by  $\text{cw}(G)$  to be the smallest positive integer  $d$  such that every non-abelian composition factor of  $G$  embeds in the symmetric group  $S_d$ . Thus for solvable groups whose composition factors are cyclic (and hence, abelian)  $\text{cw}(G) = 1$ .

**Lemma 4.1.** [6, Proposition 3.7] Let  $G \leq \text{Sym}(\Omega)$  where  $\Omega$  is an ordered set of size  $n$ . Then there is a canonical reordering of  $\Omega$  relative to which the Lex-Leader problem is solvable for every  $X \in 2^\Omega$  in time  $O(n^{\omega(d)+c})$  where  $d = \text{cw}(G)$  and where  $\omega(d) < 3.4$  for solvable groups and  $< d \log d + c$  in general. Furthermore such an ordering can be determined in polynomial time.

Remark: Using techniques of Luks [5], the running time of the algorithm can be reduced to  $O(n^{d+c})$ .

If we define  $\Gamma_d$  to be the class of groups  $G$  with  $\text{cw}(G) \leq d$  then Lemma 4.1 guarantees a polynomial time algorithm for **Lex-Leader**. To see, how this algorithm translates to a lex-leader formula, we appeal to Cook's classical theorem about the NP-completeness of SAT which we state below.

Lemma 4.2. [16, Cook] Let  $L$  be a language in NP and let the input string  $x$  be an instance of  $L$ . Then one can write a boolean formula  $\phi_L(x)$  which has a satisfying assignment iff  $x \in L$ . Furthermore, this boolean formula is of size  $O(p(n)^4)$  where  $p(n)$  is the time bound of the non-deterministic Turing machine that decides  $x \in L$  where  $n = |x|$ .

The formula  $\phi_L(x)$  depends on the algorithm (i.e., the Turing machine) used to decide whether  $x \in L$  and hence we call it the "algorithmic formula".

Now define the language

$$L = \{(G, X) \mid G \in \Gamma_d, G \leq \text{Sym}(\Omega), X \in 2^\Omega \text{ is a lex-leader in } X^G\}$$

Because of Lemma 4.1,  $L \in P \subset NP$ , so there is a formula  $\phi_L(G, X)$  which is satisfiable iff  $X$  is a lex-leader in  $X^G$ , where  $G$  is a  $\Gamma_d$  group. This formula is defined over the variables  $x_1, x_2, \dots, x_n$  representing the input bits of  $X$  and other variables. Hence the restriction of satisfying assignments of  $\phi_L(G, X)$  to  $x_1, x_2, \dots, x_n$  must be lexical leaders in their string orbits. Thus  $\phi_L(G, X)$  is indeed a lex-leader formula for  $G$ . It is of size  $O(n^{4\omega(d)+c'})$ . This is a formula of polynomial size – albeit a polynomial with a big constant exponent.

A comparison of  $\phi_L(G, X)$  and the formulas  $\phi_L(G)$  in Theorem 3.1 (ii) and (iii) is in order.  $\phi_L(G, X)$  is larger than  $\phi_L(G)$  and depends on the algorithm used to solve the lex-leader problem. Our formulas for abelian groups do not depend on the algorithm explicitly.

Remarks:

- (i) While our constructions work for abelian groups and  $\mathcal{P}_d$  groups, it would be interesting to generalize this to  $\Gamma_d$  groups. This should be possible because there is already a polynomial-size algorithmic formula  $\phi_L(G, X)$  for these groups.
- (ii) The Lex-Leader problem is not even known to be in NP for general groups whereas we have a polynomial time algorithm for an interesting class of groups, hence Cook's formula is of polynomial size. It might be interesting to improve the bound  $O(p(n)^4)$  for languages in  $P$ , which might mean smaller algorithmic formulas for lex-leaders. But whatever the improvement possible, the algorithmic formula will have to have size  $\Omega(n^{d+c})$  for these groups, as the algorithm in Lemma 4.1 (with Luks's improvement, see remark following Lemma 4.1) itself takes this much time to solve the lex-leader problem (of course, it is possible that a more efficient algorithm exists). Hence a fair comparison at this stage might be between the *time* to solve the lex-leader problem and the *size* of  $\phi_L(G)$ . Except in the case when  $G \leq Z_2^n$  when  $|\phi_L(G)| = O(n^3)$  (Theorem 3.1 (i)) and the time to solve the lex-leader problem for these groups is also  $O(n^3)$ ,  $|\phi_L(G)|$  is worse than the timing of the lex-leader formula for abelian (and  $\mathcal{P}_d$ ) groups. One possible reasoning for this anomaly might be the fact that the algorithm works on a specific string at each stage whereas the formula  $\phi_L(G)$  has to somehow encode what happens for all possible strings. But it is still an interesting

open question whether one can write  $\phi_L(G)$  of size  $O(p(n))$  where  $p(n)$  is the timing of the algorithm Lemma 4.1 to solve **Lex-leader** for  $G$ .

### 5. Exponential Lower Bounds for Lex-Leader Formulas

In this section, we exhibit an exponential lower bound on the size of the “naive” lex-leader formula,  $LL(G)$ , proving Theorem 3.1 (i).

Given  $\Omega = \{1, 2, \dots, n\}$ ,  $G \leq \text{Sym}(\Omega)$ , recall from Equation II.3 that the formulas associated with  $g \in G$  are  $C(g, i)$ :

$$[({}^gX)_{i-1} = X_{i-1}] \rightarrow [X(i) \geq ({}^gX)(i)], \text{ for } i = 1, \dots, n \quad (\text{II.4})$$

Consider the case when  $n$  is even and  $G$  stabilizes each of the sets  $\{2i - 1, 2i\}$  for  $1 \leq i \leq n/2$ .

Then  $G$  is an elementary abelian 2-group and can be identified with a subspace of  $Z_2^n$  as follows

$$g \in G \Leftrightarrow v_g \in V \leq Z_2^{n/2} \text{ where } v_g(i) = 1 \text{ iff } (2i - 1)^g = 2i$$

where  $v_g(i)$  is the  $i$ th coordinate of  $v_g$  where  $1 \leq i \leq n/2$ .

For  $g \in G$ , observe that

$$X(2j - 1) = ({}^gX)(2j - 1) \text{ iff } X(2j) = ({}^gX)(2j)$$

Thus in particular equation (II.4) is necessarily true when  $i^g = i$  or when  $i$  is even.



If  $i$  is odd and  $i^g \neq i$  then II.4 is equivalent to

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

We say that  $C(g, 2i-1)$  is trivial if it is a tautology, i.e., if  $(2i-1)^g = 2i-1$ .

We remove the clauses that are trivially true from  $LL(G)$  to obtain the formula

$$N(G) = \bigwedge_{g \in G} \bigwedge_{\substack{1 \leq i \leq n/2 \\ (2i-1)^g \neq 2i-1}} C(g, 2i-1) \quad (\text{II.5})$$

Recall that, as in the definition of  $LL(G)$ , there will be several identical clauses in  $N(G)$ . However we are ultimately concerned with the number of distinct non-prunable clauses. It suffices to prove an exponential lower bound on the number of non-prunable distinct clauses in  $N(G)$ .

For  $g \in G$ ,  $1 \leq i \leq n/2$ , let  $v_{g,i} \leq Z_2^i$  where  $v_{g,i}(j) = 1$  iff  $(2j-1)^g = 2j$  for  $1 \leq j \leq i$ .  $v_{g,i}$  is thus the projection of  $v_g$  onto the first  $i$  coordinates. If  $C(g, 2i-1)$  is non-trivial then  $v_g(i) = 1$ . For  $v, w \in Z_2^k$ ,  $v \preceq w$  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.

**Lemma 5.1.** Let  $C(g_1, 2i_1-1)$  and  $C(g_2, 2i_2-1)$  be two non-trivial clauses in  $N(G)$ . 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.** ( $\Leftarrow$ ) Trivial.

( $\Rightarrow$ ) 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:

$$X(2k - 1) = X(2k) = 0 \quad \text{if } k \in I_2, k \neq i_2$$

$$X(2i_2 - 1) = 0, X(2i_2) = 1$$

$$X(2k - 1) = 1, X(2k) = 0 \quad \text{if } k \notin I_2$$

$C(g_2, 2i_2 - 1)$  is false under this  $X$ . We show that if  $i_1 \neq i_2$  and  $I_1 \not\subseteq I_2$ , the  $C(g_1, i_1)$  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  $v_{g_1, i} \not\leq v_{g_2, i}$  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 the clause  $C(g_1, 2i_1 - 1)$  is true.  $\square$

It is possible that a clause cannot be pruned by a single other clause but some conjunction of clauses prunes it. For groups under consideration, we show that this is not possible.

Lemma 5.2. Let  $\mathcal{C} = \{C(g_1, 2i_1 - 1), C(g_2, 2i_2 - 1), \dots, C(g_k, 2i_k - 1)\}$  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  $I = \{l \mid v_{g,i}(l) = 1\}$  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  $X(2l - 1) = 1, X(2l) = 0$  (note: this is the same  $X$  as in the last lemma). If for  $1 \leq j \leq k$ , if  $i_j \neq i$ , 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 5.1 implies that  $C(g_j, 2i_j - 1)$  prunes  $C(g, 2i - 1)$ .  $\square$

Lemma 5.2 gives a combinatorial characterization of lex-leader formulas. For  $1 \leq i \leq n/2$ , define

$$V_i = \{v_{g,i} \in V \mid (2i - 1)^g = 2i\}.$$

$V_i$  is a lattice under the partial order defined by set-theoretic inclusion i.e  $v \preceq w$  in the partial order iff  $v(l) \leq w(l)$  for all  $1 \leq l \leq i$ . We can then prove the following:

**Lemma 5.3.** A clause  $C(g, 2i - 1)$  in  $N(G)$  is non-prunable iff  $v_{g,i}$  is minimal in  $V_i$ .

**Proof.** ( $\Leftarrow$ ) The clause  $C(g, 2i - 1)$  is prunable iff there is some set of clauses  $C(g_{i_j}, 2i_j - 1)$  in  $N(G)$  which prunes it. Lemma 5.2 implies that this means that  $C(g_{i_j}, 2i_j - 1)$  prunes  $C(g, 2i - 1)$  for each  $j$ . Lemma 5.1 now implies that  $i_j = i$  and  $w = v_{g_{i_j}, i} \prec v_{g,i}$ . The reverse direction is trivial to prove.  $\square$

In particular, Lemma 5.3 implies that we never need to compare  $V_i$  and  $V_j$  for prunability. Lemma 5.3 provides a bijection between the non-prunable formulas in  $N(G)$  and the minimal elements of the lattice  $V_i$ .

Define  $\min(V_i) = \{v \in V_i \mid \forall w \in V_i, w \preceq v \rightarrow v = w\}$ , i.e.,  $\min(V_i)$  is the set of minimal elements of  $V_i$ .

We can thus conclude

**Theorem 5.4.** Let  $G \cong V \leq Z_2^n$ . The number of non-prunable formulas in  $N(G)$  is

$$\eta(V) = \sum_{i=1}^n |\min(V_i)|.$$

Henceforth, we will work with these groups in their vector space representation, i.e., as subspaces of  $Z_2^n$  for some  $n$ . Our goal will be to exhibit subspaces of  $Z_2^n$  with exponentially large  $|\min(V_n)|$  – these will represent groups with an exponential number of distinct non-prunable clauses. The proof of the following theorem exhibits such a construction.

**Theorem 5.5.** There exist subspaces  $V \leq Z_2^{2n+1}$  for which

$$|\min(V_{2n+1})| \in \Omega(2^n).$$

**Proof.** Let  $G \leq \text{Sym}(\Omega)$  and  $G \equiv V(n) \leq Z_2^{2n+1}$ . For  $S \subseteq \{1, \dots, n\}$  let  $v_S \in V(n) \leq Z_2^{2n+1}$  be defined as follows:

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

Set

$$V(n) = \{v_S \mid S \subseteq \{1, \dots, n\}\}.$$

There are  $2^{n-1}$  elements in  $V_{2n+1}$ . We claim that all of them are minimal in  $V_{2n+1}$ . To see this, let  $v_S, w_{S'} \in V_{2n+1}$  with  $v_S \prec w_{S'}$ . Clearly,  $S \subset S'$ , i.e., there exists a  $j$ ,  $1 \leq j \leq n$ , such that  $v_S(j) = 0, w_{S'}(j) = 1$ . Note that since  $v_S, w_{S'} \in V_{2n+1}$ ,  $|S| \equiv 1 \pmod{2}$  and  $|S'| \equiv 1 \pmod{2}$ . Observe that, by definition,  $v_S(j+n) = v_S(j) + |S| \pmod{2} = 1$  and  $w_{S'}(j+n) = w_{S'}(j) + |S'| \pmod{2} = 0$ , so  $v_S(j+n) > w_{S'}(j+n)$ . Hence  $v_S \not\prec w_{S'}$ : a contradiction.  $\square$

If we change the order of coordinates, then the size of  $\min(V_{2n+1})$  may change. However it is easy to prove that for the vector space in  $V(n)$  in Theorem 5.5, for all reordering of coordinates,  $|\min(V_{2n+1})| \in \Omega(2^n)$ . But recall that our goal in

Theorem 3.1 (i) was to prove an exponential lower bound for *all* reorderings of  $\Omega$  where  $G \leq \text{Sym}(\Omega)$ . There are reorderings of  $\Omega$  which do not keep the points in the same orbit together. We now prove that for the groups under consideration, the size of  $N(G)$  depends only on the ordering of orbits and not the ordering of the points.

Given any ordering  $\mathcal{P} = \{\omega_1, \omega_2, \dots, \omega_n\}$  of  $\Omega$ , a canonical reordering is an order  $\mathcal{P}_0$  such that the points of each orbit are together and which preserves the orbit ordering of  $\mathcal{P}$ , i.e., if a point  $i$  appears before  $j$  in  $\mathcal{P}$  (where  $i$  and  $j$  are in different orbits), the orbit containing  $i$  appears before the orbit containing  $j$  in  $\mathcal{P}_0$ . The ordering within each orbit is also important in  $\mathcal{P}_0$ : if  $\omega_i, \omega_j$  are in the same orbit and  $\omega_i$  appears before  $\omega_j$  in  $\mathcal{P}$ , then  $\omega_i$  also appears before  $\omega_j$  in  $\mathcal{P}_0$ .

Lemma 5.6. Let  $\mathcal{P}$  be any ordering of  $\Omega$ . Then each non-trivial clause of  $\text{LL}(G)$  in this order is logically equivalent to a non-trivial clause in  $N(G)$  under  $\mathcal{P}_0$ .

Proof. Observe that a clause  $C(g, i)$  in  $\text{LL}(G)$  under ordering  $\mathcal{P}$  is

$$\left[ \bigwedge_{j < i} X(\omega_j) = X(\omega_j^g) \right] \rightarrow [X(\omega_i) \geq X(\omega_i^g)] \quad (\text{II.6})$$

If the clause  $C(g, i)$  is non-trivial then  $\omega_i \neq \omega_i^g$  and  $\omega_i^g \notin \{\omega_1, \omega_2, \dots, \omega_{i-1}\}$ . Clearly we can replace an equality appearing once in the antecedent in Equation II.6 by two equalities (but remain logically equivalent) as follows: we replace the expression  $X(j) = X(j^g)$  with the expression  $X(j) = X(j^g) \wedge X(j^g) = X(j)$ . Observe that the resulting clause is exactly the clause  $C(g, i)$  in  $\text{LL}(G)$  under the order  $\mathcal{P}_0$ .  $\square$

Similarly we can show that any non-trivial clause under the ordering  $\mathcal{P}_0$  also appears as a logically equivalent clause in the ordering  $\mathcal{P}$ . This means that only the ordering of coordinates in the vector space representation can possibly affect the

size of  $\text{LL}(G)$ . Thus we have a proof of the following corollary which now proves Theorem 3.1 (i).

Corollary 5.7. There exist groups  $G \leq \text{Sym}(\Omega)$  such that the number of non-prunable clauses of  $\text{LL}(G)$  is  $\Omega(2^{n/2})$  (where  $|\Omega| = n$ ) for all possible orderings of  $\Omega$ .

While the construction in Theorem 5.5 is such that  $N(G)$  remains exponentially large for all orderings of coordinates (because  $|V_{2n+1}|$  remains exponentially large for all orders), it would be interesting to see whether there are subspaces  $V \leq Z_2^n$ , where  $\eta(V)$  (or  $|\min(V_n)|$ ) changes drastically when the order of coordinates is changed. Unfortunately, we do not know of examples where we can exhibit such sensitivity to the order. Experimental results strongly indicate that reordering will have at best modest (polynomial) effect on the number of minimal elements in each lattice  $V_i$ . But we cannot prove this. We formalize this open problem below.

Let  $\eta(V, \pi)$  denote the value of  $\eta(V)$  when the coordinate set  $[n] = \{1, 2, \dots, n\}$  is ordered as  $\pi$ . There are  $n!$  possible orderings, each corresponding to a permutation  $\pi \in \text{Sym}([n])$ . We make the following conjecture.

Conjecture 5.8. There exists a constant  $c$  such that for all  $V \leq Z_2^n$ ,

$$\frac{\max\{\eta(V, \pi) \mid \pi \in \text{Sym}([n])\}}{\min\{\eta(V, \pi) \mid \pi \in \text{Sym}([n])\}} \leq n^c.$$

We cannot even prove this when we change an ordering  $\pi$  by swapping a pair of coordinates. Let  $S \subset \text{Sym}([n])$  be the set of  $\binom{n}{2}$  transpositions from  $\text{Sym}([n])$ . Then

we cannot even prove that for all  $V \leq Z_2^n$  and all orderings  $\pi$ ,

$$\frac{\eta(V, \pi)}{\min\{\eta(V, \pi\tau) \mid \tau \in S\}} \leq n^c.$$

The above problem is open even if  $S$  contains a single transposition  $((n-1 \ n)$ , say). We can however prove the following result which provides a general proof that for every ordering of coordinates  $|V_{2n+1}| = \Omega(2^n)$  in Theorem 5.5.

**Lemma 5.9.** Let  $V \leq Z_2^n$  be a subspace of dimension  $m$  such that  $|\min(V_n)| = 2^{m-1}$  for some ordering  $\pi$  of coordinates. Then for all possible re-orderings of coordinates,  $|\min(V_n)| \geq 2^{m-2}$ .

**Proof.** First note that  $|\min(V_n)| = 2^{m-1}$  means that all vectors in  $V_n$  are minimal. Let  $\pi'$  be another ordering of coordinates such that  $n^{\pi'} = r$ , i.e., it maps the  $n$ -th coordinate in  $\pi$  to the coordinate  $r$ . Let

$$S(r, n) = \{v \in V_n \mid v(r) = v(n) = 1\}$$

be the set of vectors in the order  $\pi'$  with 1's in coordinate  $r$  and  $n$ . It is easy to prove that  $|S(r, n)| \geq (1/2) 2^{m-1}$ . We claim that each  $v \in S(r, n)$  is minimal in  $V_n$  in the order  $\pi'$ . Suppose not: then there is some vector  $w \in V_n$  with  $w(n) = 1$  such that  $w \prec v$ . We claim that  $w(r) = 0$ . If not, then  $v$  would not be minimal in  $V_n$  in the original order  $\pi$ . Now observe that  $w \prec v$  implies that  $w + v \prec v$  and  $(w + v)(r) = 1$  so that again  $v$  cannot be minimal in  $V_n$  in the order  $\pi$ . So  $v$  has to be minimal in  $V_n$ .  $\square$

It is conceivable (but is an open question) that for some groups  $G \leq \text{Sym}(\Omega)$



the number of non-prunable elements in  $LL(G)$  will be sensitive to the ordering of  $\Omega$ . This is illustrated very nicely by the **Lex-Leader** problem defined in Section 4. While this problem is solvable in polynomial time for  $\Gamma_d$  groups when we assume an ordering of the permutation domain, it is NP-hard (and not known to be in NP) for elementary abelian 2-groups for some orderings of the permutation domain. This result is Proposition 3.1 (which we quote below) from [6] which paper also proved the result in Lemma 4.1 as a “striking counterpoint” to how the order of the permutation domain affects the computational complexity of the lex-leader problem.

Lemma 5.10. [6, Proposition 3.1] The problem of finding the lexicographic leader in the  $G$ -orbit of  $X \in 2^\Omega$  for  $G \leq \text{Sym}(\Omega)$  is NP-hard even if  $G$  is restricted to be an elementary abelian 2-group.

Lemma 5.10 says that some orders are “bad” and Lemma 4.1 says that some orders are “good” and that those good orders can be found efficiently. Thus we have the following corollary:

Corollary 5.11. Unless  $\text{NP} = \text{co-NP}$ , there is no polynomial time algorithm that computes a lex-leader formula  $\phi_L(G)$  for an arbitrary group  $G \leq \text{Sym}(\Omega)$ .

Proof. If such an algorithm existed, it would provide a reduction from the problem in Lemma 5.10 to SAT, hence forcing an NP-hard problem in co-NP to lie in NP, which forces  $\text{NP} = \text{co-NP}$ . □

Corollary 5.11 is not a deterrent to finding one can generate efficient lex-leader formulas for special classes of groups. In fact, Theorem 3.1 (ii) and (iii) show how to construct such formulas for two interesting classes of groups.

In summary, intractable examples are thus those vector spaces where an exponential number of vectors in one of the lattices for a coordinate are minimal. So if the vector space was such that *all* vectors were incomparable (in the inclusion order) then clearly it will be one of the intractable examples. This generalization leads to the concept of Sperner spaces introduced in the next section.

## 6. Sperner Subspaces

Sperner subspaces are subspaces  $V = \{v_1, \dots, v_{2^m}\}$  of  $Z_2^n$  where for all non-zero vectors  $v, w \in V$ ,  $v \preceq w \rightarrow v = w$ . Sperner subspaces can be easily seen as a generalization of Sperner families in an algebraic setting. Recall that a Sperner family [42] is a collection  $\mathcal{F}$  of subsets of a finite set  $X$  such that for all  $A, B \in \mathcal{F}$ ,  $A \subseteq B \Rightarrow A = B$ . Sperner subspaces are Sperner families closed under symmetric differences:

$$\forall A, B \in \mathcal{F}, A \neq B \rightarrow A \Delta B \in \mathcal{F}$$

where  $A \Delta B = (A \setminus B) \cup (B \setminus A)$ .

To see that the two definitions of Sperner spaces are equivalent, interpret the sets in the second definition as incidence vectors. Symmetric difference of sets then corresponds to addition in  $\text{GF}(2)$ .

Sperner families under varieties of restrictions is a well-researched area in extremal set theory, see [1, 15, 44]. A combinatorial structure equivalent to Sperner spaces is explored in Miklós[31]. These structures first arose in a paper by Katona and Srivastava [23] in the context of statistical designs. In his paper, Miklós considers the extremal properties of subspaces of  $Z_2^n$  where any two non-zero vectors have a non-empty intersection as sets. It is easy to see that this is exactly equivalent to the

“Sperner space” condition. In fact, our NP-hardness proof of Theorem 6.7 relies on this version of the Sperner condition.

In the rest of this section, we show that exponentially large Sperner spaces exist via a probabilistic argument (Subsection 6.1). We show explicit constructions of Sperner spaces (Subsection 6.3) and prove an upper bound to the maximum dimension of a Sperner space (Subsection 6.2). We also study the properties of cosets of a vector space which are Sperner (Subsection 6.4). Since Sperner spaces turn up as the intractable instances for the approach outlined in Section 2, checking whether a vector space (input as a set of basis vectors) is Sperner is important. We show that this problem is NP-hard (Subsection 6.5). We also consider the average case: for a random subspace  $V$  of  $Z_2^n$ , we estimate the number of minimal elements in  $V_n$  (Subsection 6.6).

### 6.1: Exponentially Large Sperner Spaces

We now give a probabilistic proof that exponentially large Sperner subspaces of  $Z_2^n$  exist, a result which we obtained independent of Miklós’s identical result.

**Theorem 6.1.** [Babai, Luks, Roy] For  $n \geq 5$ , there exists a Sperner subspace of dimension  $n \log(2/\sqrt{3})$ .

**Proof.** We define a probability space over  $Z_2^n$ . Choose a random set  $B = \{g_1 \dots g_m\}$  of  $m$  linearly independent vectors where  $1 \leq m \leq n$  and

$$g_i(j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } 1 \leq j \leq m \text{ and } j \neq i \end{cases}$$

For coordinates  $m+1 \leq j \leq n$ ,  $g_i(j)$  is 0 or 1 equiprobably and independently. Let  $V$  be the span of  $B$ .  $V$  is thus uniformly distributed over  $2^{m(n-m)}$  vector spaces of dimension  $m$ .

Any  $v \in V$  is uniquely expressible in the form  $g_S = \sum_{i \in S} g_i$  where  $S \subseteq \{1 \dots m\}$ . Observe that if  $v = g_S, w = g_{S'}$  and  $v \prec w$ , then  $S \subset S'$ . Hence

$$\Pr(\exists u, v \in V, u \prec v) = \Pr(\exists S, S', S \subset S', g_S \prec g_{S'})$$

We claim that for  $S \subseteq \{1 \dots m\}$ ,  $g_S$  is uniformly distributed over  $Z_2^{n-m}$ . To see this, we prove that for each coordinate  $m+1 \leq j \leq n$ ,  $g_S(j)$  is 0 or 1 equiprobably. For each such  $j$ , define  $I_j = |\{i \mid g_i(j) = 1, i \in S\}|$ . Now  $g_S(j) = 1$  iff  $I_j \equiv 1 \pmod{2}$ . Now

$$\Pr(I_j \equiv 1 \pmod{2}) = \frac{2^{|S|-1}}{2^{|S|}} = 1/2.$$

Hence  $g_S$  is uniformly distributed over  $Z_2^{n-m}$ .

For  $S \subset S' \subseteq \{1 \dots m\}$ ,  $\Pr(g_S \prec g_{S'}) = (3/4)^{n-m}$ . Observe that

$$|\{(S, S') \mid S \subseteq S'\}| = \sum_{k=0}^m \binom{m}{k} 2^{m-k} = 3^m.$$

We can then conclude that

$$\begin{aligned} \Pr(\exists u, v \in V, u \prec v) &\leq \sum_{S, S', S \subset S'} \Pr(g_S \prec g_{S'}) \\ &= \sum_{S, S', S \subset S'} \left(\frac{3}{4}\right)^{n-m} < \sum_{S, S', S \subset S'} \left(\frac{3}{4}\right)^{n-m} \end{aligned}$$

Thus,

$$\Pr(\exists u, v \in V, u \prec v) \leq 3^m \left(\frac{3}{4}\right)^{n-m}$$

Hence if  $3^m \left(\frac{3}{4}\right)^{n-m} = 1$ , i.e., when  $2^m = (2/\sqrt{3})^n$ ,  $\Pr(\exists u, v \in V, u \prec v) < 1$  which means there is some vector space  $V$  of dimension  $m$  which is Sperner. Thus there is a Sperner space of dimension  $n \log(2/\sqrt{3})$ .  $\square$

## 6.2: Upper Bound on Sperner Dimension

We now prove an upper bound on the dimension of Sperner subspaces.

**Theorem 6.2.** Let  $V \subseteq Z_2^n$  for  $n \geq 3$ , be a Sperner subspace of dimension  $m$ . Then  $m \leq \frac{n+1}{2}$ .

**Proof.** Let  $V$  be an  $m$ -dimensional Sperner subspace of  $Z_2^n$  where  $m > \frac{n+1}{2}$ . Let  $\mathcal{B} = \{g_1 \dots g_m \mid \forall g_i, g_i(i) = 1, \forall j, 1 \leq j \leq m, j \neq i \rightarrow g_i(j) = 0\}$  be a basis in canonical form for  $V$ .<sup>3</sup> Then any  $v \in V$  is some  $g_S$  where  $S \subseteq \{1 \dots m\}$ . For any  $g_S$  let  $\hat{g}_S$  be the projection of  $g_S$  onto the last  $n - m$  coordinates,  $(g_S(m+1), \dots, g_S(n))$ .

**Lemma 6.3.**  $\forall i, j, \hat{g}_i = \hat{g}_j \rightarrow i = j$ .

**Proof.** Suppose that for some  $i \neq j, 1 \leq i, j \leq m, \hat{g}_i = \hat{g}_j$ . Then  $\forall k, k \notin \{i, j\}, (g_i + g_j) \prec (g_i + g_j + g_k)$  violating the Sperner condition. Note that such a  $k$  exists because  $m > 2$  since  $m > \frac{n+1}{2} \geq 2$  as  $n \geq 3$ .  $\square$

---

<sup>3</sup>The order of coordinates is not significant in these discussions. So WLOG we can assume that the first  $m$  coordinates are the "canonical" coordinates

Now consider  $\hat{\mathcal{B}} = \{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_m \mid g_i \in \mathcal{B}\}$ . From the above lemma,  $|\hat{\mathcal{B}}| = m$ . Let  $W$  be the linear span of  $\hat{\mathcal{B}}$ . Clearly  $W \leq Z_2^{n-m}$ . Let  $\dim(W)$  be the dimension of  $W$ .

**Lemma 6.4.**  $\dim(W) \geq m - 1$ .

**Proof.** Suppose that  $\dim(W) < m - 1$ . Then  $\hat{\mathcal{B}}$  is a dependent set of vectors. Hence there exist  $S \subset \{1 \dots m\}$  and  $j \in \{1 \dots m\} - S$  such that  $|S| < m - 1$  and  $\hat{g}_S = \hat{g}_j$ . Note that  $|S \cup \{j\}| \leq m - 1 < m$ . Hence there exists a  $k \notin S \cup \{j\}$ . Note that using arguments similar to the previous lemma, we have  $g_{S \cup \{j\}} \prec g_{S \cup \{j\}} + g_k$  which violates the Sperner condition.  $\square$

However  $\dim(W) \leq n - m$  since  $W \leq Z_2^{n-m}$ . This means that  $m - 1 \leq n - m$ , i.e.,  $m \leq (n + 1)/2$  which contradicts our hypothesis.  $\square$

### 6.3: Explicit Constructions

We now give an explicit construction for Sperner subspaces of dimension  $\sqrt{n}$ . A better explicit construction, of dimension  $n^{3/4}$  is given in [31].

For each  $n > 1$  and  $1 \leq m \leq n$ , define  $T(m, n)$  to be true iff there is a Sperner subspace of dimension  $m$  in  $Z_2^n$ .

**Proposition 6.5.**  $T(m, n) \Rightarrow T(m + 1, m + n + 1)$

**Proof.** Let  $\mathcal{B}_m = \{g_1 \dots g_m \mid \forall g_i, g_i(i) = 1, \forall j, 1 \leq j \leq m, j \neq i \rightarrow g_i(j) = 0\}$  be a basis in row-reduced echelon form.

We construct a basis  $B_{m+1} = \{h_1 \dots h_{m+1}\}$  for a vector space  $V$  of dimension  $m+1$  in  $Z_2^{m+n+1}$  as follows:

$$h_1(j) = \begin{cases} 0 & \text{if } 2 \leq j \leq n+1 \\ 1 & \text{if } j = 1 \text{ or } n+2 \leq j \leq n+m+1 \end{cases}$$

The remaining vectors  $\{h_i \mid i \geq 2\}$  are defined as follows:

$$h_i(j) = \begin{cases} g_i(j-1) & \text{if } 2 \leq j \leq n+1 \\ 1 & \text{if } j = n+i \\ 0 & \text{otherwise} \end{cases}$$

$B_{m+1}$  is a set of  $m+1$  linearly independent vectors - the vectors are actually in canonical form. Let  $V_{m+1}$  denote the linear span of the vectors in  $B_{m+1}$ .

Claim 6.6.  $V_{m+1} \leq Z_2^{n+m+1}$  is Sperner.

Proof. Any vector  $v \in V_{m+1}$  is of the form  $h_S$  where  $S \subseteq \{1 \dots m+1\}$ . If  $S = \{a\}$  where  $a \in \{1 \dots m\}$  we write  $h_a$  as a shorthand for  $h_{\{a\}}$ .

If  $V_{m+1}$  is not Sperner, then there exists  $S \subset S' \subseteq \{1 \dots m+1\}$  with  $h_S < h_{S'}$ . Now if  $S \neq \{1\}$ , then  $h_S \not< h_{S'}$  since  $h_S[2 \dots n+1] = g_A$  and  $h_{S'}[2 \dots n+1] = g_B$  where  $A = \{i \mid i+1 \in S\}$  and  $B = \{i \mid i+1 \in S'\}$ , since  $h_1(j) = 0$  for all  $2 \leq j \leq n+1$  and  $g_A \not< g_B$ . Thus we need only consider the case  $S = \{1\}$  and  $S' \subseteq \{1 \dots m+1\}$  where  $1 \in S'$ . Consider  $h_{S' \setminus \{1\}}$ . For some  $j$ ,  $n+2 \leq j \leq n+m+1$ ,  $h_{S' \setminus \{1\}}(j) = 1$ . Therefore  $h_1 \not< h_{S'}$  since  $h_1(j) = 1$  and  $h_{S'}(j) = 0$ .  $\square$

Thus if  $T(m, n)$  is true, so is  $T(m+1, n+m+1)$ .  $\square$

Observe that  $T(2, 3)$  is true: the linear span of  $\{(1, 0, 1), (0, 1, 1)\}$  is a Sperner space. Starting from  $T(2, 3)$  we can thus construct an infinite family of Sperner spaces by applying the extension shown above. Thus we will have  $T(m, s)$  true where

$$s = 3 + \sum_{i=3}^m i \in \Theta(m^2).$$

Thus we can explicitly construct an infinite family of Sperner spaces of dimension  $\Theta(\sqrt{n})$  in  $Z_2^n$ .

#### 6.4: Sperner Cosets

Recall that in the proof of Theorem 5.5, we constructed a vector space  $V \leq Z_2^{2n+1}$  such that  $V_{2n+1}$  was a Sperner family.  $V_{2n+1}$  is also a coset  $W + x$  in  $V$ , where  $W = \{v \in V \mid v(2n+1) = 0\}$  and  $x$  is any vector in  $V$  with  $x(2n+1) = 1$ .

We thus define a Sperner coset to be coset in  $Z_2^n$  (i.e., it is  $W + x$  for some  $W \leq Z_2^n$  and  $x \in Z_2^n$ ) which is a Sperner family.

The proof of Theorem 5.5 gives an explicit construction of Sperner cosets of dimension  $n/2 - O(1)$  in  $Z_2^n$ .

The following argument shows that Sperner cosets can have dimension at most  $n/2$ . Let  $W + x$  be a Sperner coset, where  $W \leq Z_2^n$  is a subspace of dimension  $m$  and  $x \in Z_2^n$ . Observe that without loss of generality, we may assume that for all  $w \in W$ ,  $w(n) = 0$  and  $\mathcal{B} = \{g_1, g_2, \dots, g_m\}$  is a basis for  $W$  in row-reduced echelon form in the first  $m$  coordinates. Furthermore, we may also assume that  $x(n) = 1$  and  $x(i) = 0$  for all  $1 \leq i \leq m$  (otherwise, subtract the necessary basis vectors from  $x$  to satisfy this condition and denote the resulting vector as  $x$ ). Let  $\hat{g}_i$  denote the projection of the basis vector  $g_i$  onto the last  $n - m$  coordinates. We claim that  $\hat{\mathcal{B}} = \{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_m\}$



are independent vectors in  $Z_2^{n-m}$ . If not, there is some subset  $S \subset \{1, 2, \dots, m\}$  and a vector  $\hat{g}_S = \sum_{i \in S} \hat{g}_i$  in  $Z_2^{n-m}$  such that  $\hat{g}_S(i) = 0$  for all  $m+1 \leq i \leq n$ . Observe that  $x \prec g_S + x$ , which violates the Sperner condition. Hence  $\mathcal{B}$  is independent, which implies that  $m \leq n/2$ .

Sperner cosets are more natural examples of “bad groups” for symmetry breaking as they are much less restrictive than Sperner spaces.

### 6.5: Identification of Sperner Spaces

We consider the complexity of the question: given a subspace  $V$  of  $Z_2^n$  in the form of a basis, is it a Sperner space? We prove below that the complement of the question (defined below) is NP-hard.

Let  $V \leq Z_2^n$  be a vector space over  $Z_2$ . For  $v \in V$ , define the *support* of  $v$  as  $\text{supp}(v) = \{i \mid 1 \leq i \leq n, v(i) = 1\}$ . We say that the non-zero vectors  $v, w \in V$  are disjoint if  $\text{supp}(v) \cap \text{supp}(w) = \emptyset$ . We observe that a Sperner space is a subspace  $V \leq Z_2^n$  where supports of any pair of non-zero  $v, w \in V$  have a non-trivial intersection (if the intersection was empty, observe that  $v \prec v + w$ , violating the Sperner condition).

We define the following problem:

#### DISJOINT VECTORS (DV)

Instance: A set of linearly independent vectors  $S \subseteq Z_2^n$

Question: Does there exist two disjoint vectors  $v, w$  in the vector space  $V = \langle S \rangle$  ?

DV is in NP: a non-deterministic Turing machine guesses  $v, w$  and checks (in polynomial time) that  $\text{supp}(v) \cap \text{supp}(w) = \emptyset$ . It also checks that  $v, w \in V$ : this can be done in polynomial time using standard linear algebra (e.g. Gaussian elimination).

Theorem 6.7. **DISJOINT VECTORS** is NP-hard.

Proof. We show a reduction from the following problem:

Exact 3-Cover(X-3C)

Instance( $X, M$ ): A set  $X$  and a collection  $M$  of 3-element subsets of  $X$ .

Question: Does there exist a collection of (disjoint) subsets of  $M$  such that their union is  $X$ ?

Let  $X = \{x_1, x_2 \dots x_s\}$  and  $M = \{m_1, m_2 \dots m_t\}$ , i.e.,  $s = |X|, t = |M|$ .

From an instance of X-3C as above, we construct an instance of DV as follows: we construct a vector space of dimension  $r = s + 2t + 2$  in  $Z_2^n$  where  $n = r + s^2 + 4t^2 + 3s + 4t + 3st \in O(r^2)$  which has disjoint vectors iff the instance  $(X, M)$  of X3C is a yes instance. We exhibit this vector space by constructing a basis of  $r$  vectors.

The basis is  $K = K_X \cup K_M \cup \bar{K}_M \cup \{a, b\}$  where  $K_X = \{v_1, v_2, \dots v_s\}$ ,  $K_M = \{w_1, w_2 \dots w_t\}$  and  $\bar{K}_M = \{\bar{w}_1, \bar{w}_2 \dots \bar{w}_t\}$ , where  $K_X, K_M, \bar{K}_M, \{a, b\}$  are all disjoint sets.

Informally, we will interpret  $v_i \in K_X$  as a representative of the element  $x_i \in X$  and  $w_i$  as (a representative of) the set  $m_i$  in  $M$ . The vectors  $\bar{w}_i$  will be “dual” vectors to  $w_i$  in a natural way. The vectors  $a, b$  are special vectors whose purpose will become clear later.

The vectors in  $K$  are in row-reduced echelon form in the first  $r$  coordinates. Let  $V = \langle K \rangle$ . Any  $v \in V$  is of the form  $g_S$  where  $S \subseteq K$ . If there are vectors  $g_S, g_{S'} \in V$  such that  $\text{supp}(g_S) \cap \text{supp}(g_{S'}) = \emptyset$  then clearly  $S \cap S' = \emptyset$ .

Informally, we want to achieve the following: if  $S, S'$  are 2 opposing sets such that  $g_S$  and  $g_{S'}$  are disjoint, then (i)  $a \in S, b \in S'$  (ii) every  $v_i$  sides with  $a$  in  $S$ , i.e.,  $K_X \subset S$  (iii) for each  $v_i \in S$ , at least one  $w_j$  sides with  $b$  in  $S'$  where  $w_j$  is a

representative of the set  $m_j$  where  $x_i \in m_j$  and (iv) all the  $w_j$ 's that side with  $b$  have to be disjoint. Note that if we have two opposing sets as above, we automatically can construct a cover for  $X$ .

We now proceed to specify the remaining set of  $n - r$  coordinates. We do this in steps, each step specifying a block of coordinates that corresponds to a specific gadget serving a particular purpose. <sup>4</sup> Coordinates that are left unspecified for vectors in the following description are set to 0.

Block 1: ( $v_i$  cannot oppose  $v_j$ )

We specify the  $s^2$  coordinates  $\alpha + 1 \dots \alpha + s^2$  where  $\alpha = r$ .

Fix a bijection from  $f$  from  $X \times X$  to  $[\alpha + 1 \dots \alpha + s^2]$ . For each  $p = (x_i, x_j) \in X \times X$ , define  $v_i(f(p)) = v_j(f(p)) = 1$ .

Block 2: ( $w_i$  cannot oppose  $w_j$ )

We specify the  $t^2$  coordinates  $\alpha + 1 \dots \alpha + t^2$ , where  $\alpha = r + s^2$ .

Fix a bijection  $f : M \times M \rightarrow \alpha + 1 \dots \alpha + t^2$ . For each  $p = (m_i, m_j) \in M \times M$ , define  $w_i(f(p)) = w_j(f(p)) = 1$ .

Block 3: ( $\bar{w}_i$  cannot oppose  $\bar{w}_j$ )

We specify the  $t^2$  coordinates from  $\alpha + 1 \dots \alpha + t^2$ , where  $\alpha = r + s^2 + t^2$ .

Fix a bijection  $f : M \times M \rightarrow \alpha + 1 \dots \alpha + t^2$ . For each  $p = (m_i, m_j) \in M \times M$ , define  $\bar{w}_i(f(p)) = \bar{w}_j(f(p)) = 1$ .

Block 4: ( $v_i$  cannot oppose  $a$ )

We specify the  $s$  coordinates from  $\alpha + 1 \dots \alpha + s$ , where  $\alpha = r + s^2 + 2t^2$ .

For each  $1 \leq i \leq s$  set  $v_i(\alpha + i) = a(\alpha + i)$ .

---

<sup>4</sup>As will be apparent certain blocks will be redundant, their purpose being fulfilled by other blocks. But for reasons of clarity we leave them in.

Block 5: (  $w_i$  cannot oppose  $b$  )

We now specify the  $t$  coordinates  $\alpha + 1 \dots \alpha + t$ , where  $\alpha = r + s^2 + 2t^2 + s$ .

For each  $1 \leq i \leq t$ , set  $w_i(\alpha + i) = b(\alpha + i) = 1$ .

Block 6: (  $\bar{w}_i$  cannot oppose  $a$  )

We specify the  $t$  coordinates  $\alpha + 1 \dots \alpha + t$  where  $\alpha = r + s^2 + 2t^2 + s + t$ .

For each  $1 \leq i \leq t$  set  $\bar{w}_i(\alpha + i) = a(\alpha + i) = 1$ .

Block 7: (  $\bar{w}_i$  cannot oppose  $v_j$  )

We specify the  $st$  coordinates  $\alpha + 1 \dots \alpha + st$ , where  $\alpha = r + s^2 + 2t^2 + s + 2t$ .

Fix a bijection  $f$  from  $X \times M$  to  $[\alpha + 1 \dots \alpha + st]$ . For each  $p = (x_j, m_i) \in X \times M$  set

$$v_j(f(p)) = \bar{w}_i(f(p)) = 1.$$

Block 8: ( If  $\bar{w}_i$  is opposite  $w_i$  then both  $a$  and  $b$  have to be used )

We now specify the set of  $2t$  coordinates  $\alpha + 1 \dots \alpha + 2t$ , where  $\alpha = r + s^2 + 2t^2 + s + 2t + st$ .

Set

$$\begin{aligned} b(\alpha + i) &= w_i(\alpha + i) = \bar{w}_i(\alpha + i) = 1 \\ a(\alpha + 2i) &= w_i(\alpha + 2i) = \bar{w}_i(\alpha + 2i) = 1 \end{aligned}$$

Block 9: (  $v_i$  cannot oppose  $w_j$  without help of  $a$  )

We specify the  $st$  coordinates  $\alpha + 1 \dots \alpha + st$  where  $\alpha = r + s^2 + 2t^2 + s + 4t + st$ .

Fix a bijection  $f$  from  $X \times M$  to  $[\alpha + 1 \dots \alpha + st]$ .

For each  $p = (x_i, m_j) \in X \times M$  set

$$v_i(f(p)) = w_j(f(p)) = a(f(p)) = 1.$$

Block 10: (  $v_i$  cannot oppose  $w_j$  without the help of  $\bar{w}_j$  )

We specify the  $st$  coordinates  $\alpha + 1 \dots \alpha + st$  where  $\alpha = r + s^2 + 2t^2 + s + 4t + 2st$ .

Fix a bijection  $f$  from  $X \times M$  to  $[\alpha + 1 \dots \alpha + st]$ . For each  $p = (x_i, m_j) \in X \times M$  set

$$v_i(f(p)) = w_j(f(p)) = \bar{w}_j(f(p)) = 1.$$

Block 11: ( All  $v_i$  have to be used if  $a$  and  $b$  are used )

We specify the  $s$  coordinates  $\alpha + 1 \dots \alpha + s$  where  $\alpha = r + s^2 + 2t^2 + s + 4t + 3st$ .

For each  $1 \leq i \leq s$  set  $v_i(\alpha + i) = a(\alpha + i) = b(\alpha + i) = 1$ .

Block 12: ( All  $w_i$ 's used have to be disjoint )

We specify the  $t^2$  coordinates  $\alpha + 1 \dots \alpha + 2t^2$  where  $\alpha = r + s^2 + 2t^2 + 2s + 4t + 3st$ .

Fix a bijection  $f$  from  $M \times M$  to  $\alpha + 1 \dots \alpha + 2t^2$ . For each pair  $p = (m_i, m_j) \in M \times M$  such that  $i \neq j$  and  $m_i \cap m_j \neq \emptyset$ , set

$$w_i(f(p)) = \bar{w}_j(f(p)) = 1.$$

Block 13: ( For each  $v_i$  used there has to be at least one (actually an odd number of)  $w_j$  opposite it where  $x_i \in w_j$  if  $b$  is already opposite  $v_i$  ) We specify the  $s$  coordinates  $\alpha + 1 \dots \alpha + s$  where  $\alpha = r + s^2 + 3t^2 + 2s + 4t + 3st$ .

For each  $x_i \in X$  set  $v_i(\alpha + i) = b(\alpha + i) = 1$ . Also set  $w_j(\alpha + i) = 1$  for all  $1 \leq j \leq t$  and  $x_i \in m_j$ .

Block 14 ( If  $\bar{w}_i$  is opposite  $w_k$  , then either  $w_i$  or  $\bar{w}_k$  has to be used)

We specify the  $t^2$  coordinates  $\alpha+1 \dots \alpha+t^2$  where  $\alpha = r+s^2+3t^2+3s+4t+3st$ .

Fix a bijection  $f$  from  $M \times M$  to  $\alpha+1 \dots \alpha+t^2$ . For each pair  $p = (m_i, m_j)$  such that  $i \neq j$  set

$$w_i(f(p)) = w_j(f(p)) = \bar{w}_j(f(p)) = \bar{w}_i(f(p)).$$

Lemma 6.8. Let  $g_S$  and  $g_{S'}$  be disjoint. Then either  $a \in S, b \in S'$  or  $a \in S', b \in S$ .

Proof. We need only consider the following cases:

Case 1:

Assume that  $v_i \in S$ . Clearly no other  $v_j$  can belong to  $S'$  (by Block 1). Also  $(\{a\} \cup K_{M'}) \cap S' = \emptyset$  by blocks 4 and 6. So if  $y \in S'$  then  $y$  is either some  $w_j$  or  $y = b$ .

Suppose  $y = b$ . Then by block 11,  $a$  has to be used. Now  $a$  cannot be in  $S'$  (by block 4) so  $a \in S$ . So we have  $a \in S, b \in S'$ .

Suppose instead that  $y = w_j$ . Now by block 9,  $a$  has to be used. By block 4,  $a$  has to go in  $S$ . If  $v_i \in S$  and  $w_j \in S'$  then by block 10,  $\bar{w}_j$  has to be used. Since  $\bar{w}_j$  cannot oppose  $a$  (block 6) it cannot be in  $S'$ . So  $\bar{w}_j \in S$ , i.e., it opposes  $w_j$ . But this it cannot do so without the use of  $b$  via block 8. But  $b$  cannot go into  $S$  because of block 5. So  $b \in S'$ . Thus  $a \in S, b \in S'$ .

Case 2:

Assume that  $w_i \in S$ . Then  $y \in S'$  is either  $a$  or  $v_j \in K_X$  or some  $\bar{w}_l$ . If  $y = v_j$  we are back to case 1. If  $y = a$  then because of block 9 there is some  $v_j \in S'$  ( $v_j \notin S$  because

of block 4) and again we are back to case 1. If  $y = \bar{w}_l$  then by block 14, either  $w_l \in S$  or  $\bar{w}_l \in S'$  (both could happen). In which case, we have either  $w_l \in S, \bar{w}_l \in S'$  or  $w_l \in S, \bar{w}_l \in S$  and by block 8,  $a, b$  have to be used.

Case 3:

Assume  $\bar{w}_i \in S$ . Then there is some  $y \in S'$  which is either some  $w_j$  or  $b$ . If  $y = b$  then by block 8,  $w_i \in S'$ . Then since  $w_i \in S', \bar{w}_i \in S$ , by block 14,  $a$  has to be used opposite  $b$ . If instead  $y = w_j$  then by block 14 and block 8,  $a$  and  $b$  have to be used.

Case 4:

Assume  $a \in S$ . Then  $y \in S'$  is either some  $w_j$  or  $b$ . If  $y = b$  we are done. Suppose  $y = w_j$ . By block 8,  $\bar{w}_j \in S$ . Then by block 8 again,  $b \in S'$ .

Case 5:

Assume that  $b \in S$ . Then  $y \in S'$  is either some  $v_j$  or some  $\bar{w}_k$  or  $a$ . If  $y = a$  we are done. If  $y = v_j$ , then by block 11,  $a \in S'$ . Suppose instead that  $y = \bar{w}_k$ , then by block 8,  $w_k \in S$ . Since  $w_k \in S, \bar{w}_k \in S'$ , by block 8,  $a \in S'$ .

□

Suppose we have 2 disjoint vectors  $g_S, g_{S'}$ . WLOG  $a \in S$  and  $b \in S'$ . Observe that because of block 11,  $K_X \subset S$ . For each  $v_x \in S$  at least one  $w_j$  where  $x \in m_j$  has to be in  $S'$  (block 13). For each such  $w_j \in S'$  we must have  $\bar{w}_j \in S$  (block 10).

Now observe that all the  $w_j$ 's in  $S'$  must be disjoint because of block 12. Thus the  $m_j$  corresponding to the  $w_j$ 's used constitute a perfect cover.

Suppose that the instance  $(X, M)$  of X3C was a yes instance. Let  $M' \subseteq M$  constitute a perfect cover for  $X$ . Observe then that the vectors  $g_S$  and  $g_{S'}$  are disjoint where  $S = K_X \cup \{a\} \cup \{\bar{w}_j \mid m_j \in M'\}$  and  $S' = \{b\} \cup \{w_j \mid m_j \in M'\}$ . Thus the instance of DV is also a yes instance.  $\square$

### 6.6: The Average Case

The probabilistic proof of Theorem 6.1 actually shows that Sperner spaces are plentiful, i.e., a random subspace of  $V \leq Z_2^n$  of dimension  $m = n \log \frac{2}{\sqrt{5}}$  is Sperner with high probability. This automatically implies that for a random subspace  $V$  of this dimension  $V_n$  will also be a Sperner coset with high probability. In this subsection, we consider the average case: given a random vector space  $V$ , what is the expected number of minimal elements in  $V_n$ ? This will give us an idea of how frequent the "bad" groups arise. If they are very rare, then in practice, it may be possible to write small lex-leader formulas for these groups, despite the intractable examples.

Let  $\mathcal{B} = \{g_1 \dots g_m\} \subseteq Z_2^n$  where  $1 \leq m < n/2$ , be a set of random vectors with a canonical projection in the first  $m$  coordinates. That is

$$g_i(j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } 1 \leq j \neq i \leq m \end{cases}$$

and for  $m + 1 \leq j \leq n$ ,  $g_i(j) = 0$  or 1 equiprobably and independently. Let  $V \leq Z_2^n$  denote the linear span of  $\mathcal{B}$ . Any vector  $v \in V$  is of the form  $g_S$  for some  $S \subseteq \{1 \dots m\}$ .

We want to compute the expected number of minimal elements in the lattice



of vectors in  $V$  with a 1 in the last (i.e.,  $n$  th) coordinate i.e. in the set  $V_n = \{v \in V \mid v(n) = 1\}$ .

For  $S \subseteq \{1 \dots m\}$ ,  $\Pr(g_S(n) = 1) = 1/2$ . Assume henceforth that  $g_S(n) = 1$  (we will eventually correct for the conditional probability). Let  $z_S = \{i \mid m+1 \leq i \leq n-1, g_S(i) = 0\}$ . Clearly  $z_S$  is uniformly distributed over  $Z_2^{n-m-1}$ . For  $i \in S$ , define  $\hat{g}_i$  for  $i \in S$  to be the projection of  $g_i$  onto the coordinates in  $z_S$ . Let  $t = |z_S|$  and

$$\mathcal{P}(\hat{S}) = \{\hat{g}_i \mid i \in S\} \subseteq Z_2^t.$$

We first make the following claim:

Claim 6.9.  $\exists w \in V, w \prec g_S$  iff  $\exists z \in V, z(n) = 1, z \prec g_S$ .

Proof. ( $\Rightarrow$ ) Suppose  $w(n) = 1$ , then we are done as  $z = w$ . Suppose  $w(n) = 0$ . Consider  $w + g_S$ . Since for all  $1 \leq i \leq n$ ,  $w(i) \leq g_S(i)$ ,  $w(i) + g_S(i) \leq g_S(i) + g_S(i) \leq g_S(i)$ . Hence  $w + g_S \prec g_S$  since  $w \prec g_S$ . Also  $(w + g_S)(n) = 1$  since  $g_S(n) = 1$  and  $w(n) = 0$ . The reverse direction is trivial.  $\square$

The above claim implies that to check whether a particular vector  $v \in V$  with  $v(n) = 1$  has another vector  $w \in V$  with  $w \prec v$  and  $w(n) = 1$ , we can effectively ignore the condition  $w(n) = 1$ .

Claim 6.10.  $\exists w \in V, w \prec g_S$  iff  $\langle \mathcal{P}(\hat{S}) \rangle \leq Z_2^n$  has dimension strictly less than  $|S| - 1$ .

Proof. ( $\Rightarrow$ ) Let  $w = g_A$  where  $A \subset S$  and  $w \prec g_S$ . Note that  $\mathcal{P}(\hat{S})$  cannot have dimension  $|S|$  since  $\sum_{v \in \mathcal{P}(\hat{S})} v = 0$ . If  $\mathcal{P}(\hat{S})$  has dimension  $|S| - 1$ ,  $\{\hat{g}_i \mid i \in A\}$  is

a set of linearly independent vectors. Hence  $\hat{g}_A \neq 0^t$ . This means that there is some coordinate  $j$  in  $z_S$  such that  $g_A(j) = 1$  and  $g_S(j) = 0$  which implies that  $w \not\prec g_S$ , a contradiction. The proof in the other direction is trivial.  $\square$

The set  $\mathcal{P}(\hat{S})$  consists of  $k = |S|$  vectors  $\hat{g}_1, \hat{g}_2, \dots, \hat{g}_k$  uniformly chosen from  $Z_2^t$  such that their sum is the zero vector  $0^t$ . Once  $\hat{g}_1 \dots \hat{g}_{k-1}$  are chosen, then  $\hat{g}_k$  is automatically determined ( $\because \hat{g}_k = \sum_{i=1}^{k-1} \hat{g}_i$ ).

Claim 6.11.  $\exists w \in V, w \prec g_S$  iff  $\{\hat{g}_1 \dots \hat{g}_{k-1}\}$  is linearly dependent.

Proof. ( $\Leftarrow$ ) Follows from Claim 6.10.

( $\Rightarrow$ ) Let  $w = g_A$  where  $A \subsetneq S$ . Consider the two cases:

Case 1: If  $A \subseteq \{1, 2, \dots, k-1\}$  then  $\{\hat{g}_i \mid i \in A\}$  is linearly dependent. Hence the super-set  $\{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_{k-1}\}$  is linearly dependent.

Case 2: Now suppose  $k \in A$ . Since  $w \prec g_A, \hat{g}_A = 0^t$  so

$$\sum_{i \in A \cap \{1, 2, \dots, k-1\}} \hat{g}_i + \hat{g}_k = 0^t.$$

But  $\hat{g}_k = \sum_{i=1}^{k-1} \hat{g}_i$ . So  $\sum_{i \in A \cap \{1, 2, \dots, k-1\}} \hat{g}_i + \sum_{i=1}^{k-1} \hat{g}_i = 0$ . Recall that  $A \not\subseteq \{1, 2, \dots, k\}$  so the last sum is a non-trivial linear combination of vectors  $\{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_{k-1}\}$  which means that these vectors are linearly dependent.  $\square$

Let  $\beta(t, s)$  be the probability that these  $s$  random vectors in  $Z_2^t$  are linearly independent. Thus  $g_S$  is not minimal with probability  $\beta(t, k-1)$ . Then the expected number  $\mu$  of minimal elements in  $V_n$  is

$$= \sum_{S \subset \{1, 2, \dots, m\}} \Pr(g_S(n) = 1) \Pr(g_S \text{ is minimal})$$

$$= \frac{1}{2} \sum_{k=1}^m \binom{m}{k} \sum_{t=0}^{n-m-1} \binom{n-m-1}{t} \frac{1}{2^{n-m-1}} \beta(t, k-1) \quad (\text{II.7})$$

as there are  $\binom{m}{k}$  possible  $S$  and  $2^{n-m-1}$  choices for  $z_S$  for each  $S$ .

We now compute  $\beta(t, s)$ . It is easy to see (see [11]) that  $s$  vectors chosen uniformly from  $Z_2^t$  are linearly independent with probability

$$\begin{aligned} &= \frac{(2^t-1)(2^t-2)\dots(2^t-2^{s-1})}{2^{ts}} \\ &= \prod_{i=0}^{s-1} (1 - 1/2^{t-i}) \end{aligned}$$

We now find a non-trivial lower bound for the expected value  $\mu$ .

We first derive an lower bound for  $\beta(t, s)$ .

**Lemma 6.12.** For all  $s \geq 0$  and  $t \geq 0$ ,

$$\beta(t, s) \geq 1 - \frac{2^s}{2^t}.$$

**Proof.** (By induction) First notice that

$$\beta(t, 1) = (1 - \frac{1}{2^t}) > 1 - (2/2^t).$$

Assume that  $\beta(t, i) \geq 1 - 2^i/2^t$  for all  $1 \leq i \leq k < t$ . Then

$$\begin{aligned} \beta(t, k+1) &= \beta(t, k) \left(1 - \frac{2^k}{2^t}\right) \\ &> \left(1 - \frac{2^k}{2^t}\right) \left(1 - \frac{2^k}{2^t}\right) \text{ since } \beta(t, k) > \left(1 - \frac{2^k}{2^t}\right) \text{ by the induction hypothesis} \\ &= \left(1 - \frac{2^k}{2^t}\right)^2 \end{aligned}$$

$$\begin{aligned}
&= 1 - \frac{2^{k+1}}{2^t} + \frac{4^k}{4^t} \\
&> 1 - \frac{2^{k+1}}{2^t}
\end{aligned}$$

Hence  $1 - \beta(t, s) \leq \frac{2^s}{2^t}$  (observe that when  $s > t$  or when  $s = t = 0$ , the lemma is trivially satisfied).  $\square$

Substituting the lower bound for  $\beta(t, k - 1)$  into the expression for  $\mu$ , we get,

$$\begin{aligned}
\mu &\geq \frac{1}{2} \sum_{k=1}^m \binom{m}{k} \sum_{t=0}^{n-m-1} \binom{n-m-1}{t} \frac{1}{2^{n-m-1}} \left(1 - \frac{2^{k-1}}{2^t}\right) \\
&= \frac{1}{2} \sum_{k=0}^m \binom{m}{k} \sum_{t=0}^{n-m-1} \binom{n-m-1}{t} \frac{1}{2^{n-m-1}} \left(1 - \frac{2^{k-1}}{2^t}\right) - O(1)
\end{aligned}$$

This implies:

$$\mu \geq 2^{m-1} - \frac{1}{3} 3^m \left(\frac{3}{4}\right)^{n-m} - O(1).$$

Experimental results show that the lower bound is indeed very accurate. We make the following claim:

Claim 6.13. Let  $V \leq Z_2^n$  be a random vector space of dimension  $m < n \log(4/3) \approx .41n$  with a canonical projection in the first  $m$  coordinates, then the average size of  $\min(V_n)$  is at least  $(1 - o(1))2^{m-1}$ , i.e., almost all the elements in the  $V_n$  are minimal.

Remark: It is also possible to find an upper bound to  $\mu$  by finding an upper bound to  $\beta(t, k - 1)$ . Numerical values indicate that these bounds are very close.

## 7. Polynomial Size Lex-Leader Formulas for Abelian Groups

Our main goal in this section is to exhibit polynomial size lex-leader formulas  $\phi_L(G)$  for  $G$  abelian (Subsection 7.4).

Recall from Section 2 that the lex-leader formula  $LL(G)$  for arbitrary  $G \leq \text{Sym}([n])$  is

$$\bigwedge_{i \in [n]} \text{Pivot}(i)$$

where

$$\text{Pivot}(i) = \bigwedge_{g \in G} \left\{ \bigwedge_{j \leq i-1} X(j) = X(j^g) \right\} \rightarrow X(i) \geq X(i^g)$$

We tackle abelian groups in stages. Since subspaces of  $Z_2^n$  were the pathological examples for  $LL(G)$ , we first show a polynomial size  $\phi_L(G)$  for these groups (Subsection 7.1). We then consider the case when the orbit constituents of abelian  $G$  are cyclic (Subsection 7.2). Then we consider the general situation (in Subsection 7.4). In all cases, we build  $\phi_L(G)$  from  $LL(G)$  and obtain a small formula by introducing a polynomial number of new variables.

In the abelian situation, an essential ingredient in our construction is the ability of finding short boolean formulas which are satisfiable iff a system of equations is *not solvable* over the ring of integers  $Z_m$  (not surprisingly, this ring is actually a field  $Z_p$  when we consider abelian groups which are subspaces of  $Z_p^n$ ). We show how to construct such short formulas in Subsection 7.3.

7.1: Lex-Leader Formulas for Subspaces of  $Z_2^n$ 

We begin with some preliminary results. Let  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  and  $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_n)$  be 2  $n$ -bit vectors from  $\text{GF}(2)^n$ . Also, let  $b \in \text{GF}(2)$ . Let  $E(= E(\mathbf{x}, \epsilon, b))$  denote the equation  $\sum_{i=1}^n \epsilon_i x_i = b$ .

**Lemma 7.1.** There exists a boolean formula  $\phi$  of size  $\Theta(n)$  defined over  $3n$  boolean variables ( $\epsilon_i, x_i, b$  and  $n - 1$  additional variables) 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). □

Given a system of equations, we can now apply the construction in Lemma 7.1 to each equation.

**Lemma 7.2.** Let  $Ax = b$  be a system of equations over  $\text{GF}(2)$  where  $A$  is an  $m \times n$  matrix. Then there is a boolean formula  $\phi(A, b)$ , which is satisfiable iff  $Ax = b$  is solvable. Furthermore,  $\phi(A, b)$  is of size  $\Theta(mn)$  and is defined over the  $m(2n + 1)$  variables  $A(i, j), b_i, x_i$  and  $m(n - 1)$  additional variables,

**Remark:** Observe that finding a solution to an  $n \times n$ -system of equations is in polynomial time (via Gaussian elimination). This algorithm runs in time  $O(n^3)$ . Cook's

theorem [16] now guarantees the existence of a boolean formula which encodes the computation path of this algorithm. Clearly such a formula would be asymptotically larger than the formula obtained by Lemma 7.2 (which is  $O(n^2)$ ).

Let  $Ax = b$  be a system of equations over  $\text{GF}(2)$ , where  $A$  is an  $m \times n$  matrix.

**Lemma 7.3.** There exists a system of equations  $Cx = d$  over  $\text{GF}(2)$ , where  $C$  is an  $(n + 1) \times m$  matrix, which is solvable iff  $Ax = b$  is not solvable.

**Proof.** Consider the vector space spanned by the rows for  $A$  together with  $b$ , i.e., the row space  $R$  of the matrix  $[A \ b]$ . The system of equations  $Ax = b$  is not solvable iff the vector  $[0 \ 0 \ \dots \ 1]$  is in the linear span of the vectors in  $R$  (which corresponds to a system of equations  $Cx = d$ ).  $\square$

Now Lemma 7.2 and Lemma 7.3 imply the following:

**Lemma 7.4.** Let  $Ax = b$  be a system of equations over  $\text{GF}(2)$  (where  $A$  is an  $m \times n$  matrix). Then there is a boolean formula of size  $\bar{\phi}(A, b)$ , of size  $\Theta(mn)$  defined over variables  $A(i, j), x_i, b_i$  and  $(n + 1)(m - 1)$  additional variables, which is satisfiable iff  $Ax = b$  is *not* solvable.

Let  $G \leq \text{Sym}(\Omega)$  be as described in Section 5, i.e.,  $G \equiv W \leq Z_2^{n/2}$  be a group acting on  $n$  points  $[n] = \{1, 2, \dots, n\}$  where the orbits of  $G$  are the sets  $\{2i - 1, 2i\}$  for each  $1 \leq i \leq n/2$  (after suitable reordering of  $\Omega$  if necessary). Obviously  $g \in G \equiv w \in W$  where  $w_i = 1$  iff  $(2i - 1)^g = 2i$ .

Recall from Equation II.5 that the lex-leader formula for  $G$  is

$$N(G) = \bigwedge_{g \in G} \bigwedge_{\substack{1 \leq i \leq n/2 \\ (2i-1)^g \neq 2i-1}} \text{Fix}(g, X, 2i-2) \rightarrow \text{Geq}(g, X, 2i-1)$$

Observe that in the last expression, we have

$$\text{Fix}(g, X, 2i-2) = \left( \bigwedge_{j < i, (2j-i)^g \neq 2j} X(2j-1) = X(2j) \right)$$

and since  $(2i-1)^g \neq 2i-1$ , we also have

$$\text{Geq}(g, X, 2i-1) = X(2i-1) \geq X(2i).$$

Define  $a_k = 1$  iff  $X(2k-1) = X(2k)$ . We rewrite  $N(G)$  so that the variables are  $w_i$  where  $w \in W$  and write  $N(W)$  instead of  $N(G)$ .

This allows us to simplify  $\text{Fix}(g, X, 2i-2)$  as follows (we write  $w$  for  $g$ ):

$$\begin{aligned} \text{Fix}(g, X, 2i-2) &\equiv \left[ \bigwedge_{k \leq i/2, w_k=1} a_k \right] \\ &= \bigwedge_{k \leq i/2} [(1 - a_k)w_k = 0 \pmod{2}] \end{aligned}$$

Similarly we simplify  $\text{Geq}(g, X, 2i-1)$  as follows:

$$\text{Geq}(g, X, 2i-1) = [w_{(i+2)/2} = 0 \vee (X(i+2) \rightarrow X(i+1))].$$



Thus

$$\begin{aligned}
 N(W) &= \bigwedge_{i \text{ even}} \bigwedge_{w \in W} \text{Fix}(g, X, 2i - 2) \rightarrow \text{Geq}(g, X, 2i - 1) \\
 &= \bigwedge_{i \text{ even}} [\neg \exists w \in W \left( \bigwedge_{k \leq i/2} (1 - a_k) w_k = 0 \wedge w_{(i+2)/2} = 1 \right) \vee \\
 &\quad (X(i+2) \rightarrow X(i+1))] \text{ (substituting the expressions for Fix and Geq)}
 \end{aligned}$$

We can simplify the last expression to

$$N(W) = \bigwedge_{i \text{ even}} [\Phi(W, X, i) \vee (X(i+2) \rightarrow X(i+1))] \quad (\text{II.8})$$

where

$$\Phi(W, X, i) = \neg \exists w \in W \left( \bigwedge_{k \leq i/2} (1 - a_k) w_k = 0 \wedge w_{(i+2)/2} = 1 \right).$$

Let  $\{b_1, b_2, \dots, b_r\}$  be a set of basis vectors of  $W$ . Then any  $w \in W$  can be expressed as a linear combination  $w = \sum c_i b_i$ .

Hence  $\Phi(W, X, i)$  can be rewritten as the following expression: (observe that this is not a strict boolean formula any more)

$$\neg \exists_{w \in \{0,1\}^n} \left( w = \sum c_i b_i \right) \wedge \left( \bigwedge_{k \leq i/2} (1 - a_k) w_k = 0 \wedge w_{(i+2)/2} = 1 \right).$$

The above expression is equivalent to the *non-solvability* (because of the negated existential quantifier in the expression) of a system of equations  $Ax = b$  over  $\text{GF}(2)$  where  $A$  is a  $(2n+1) \times 2n$  matrix and  $x$  is a  $2n \times 1$  vector of unknowns (the unknowns are  $w_i, c_i$ ). Note also that the system of equations is almost homogeneous – all but one coordinate of  $b$  is 0. Hence we want a boolean formula which is satisfiable iff  $Ax = b$  is not solvable. Lemma 7.4 shows how to construct such a boolean formula

$\bar{\phi} = \bar{\phi}(A, b)$  of size  $O(n^2)$  defined over  $A(i, j), b_i$  and  $O(n^2)$  additional variables ( $n - 1$  additional variables per equation).

Hence

$$N(W) = \bigwedge_{i \text{ even}} [\bar{\phi} \vee (X(i+2) \rightarrow X(i+1))].$$

This implies that  $N(W)$  has size  $O(n^3)$  and is defined over  $O(n^3)$  additional variables ( $O(n^2)$  for each even  $i$ ). We define  $\phi_L(G) = \phi_L(W) = N(W)$ . Thus we have a proof of the following theorem:

**Theorem 7.5.** Let  $G \leq \text{Sym}(\Omega)$  be a group with orbits of size  $\leq 2$ . Then for all orderings of  $\Omega$  there is a lex-leader formula  $\phi_L(G)$  of size  $O(n^3)$  defined over  $O(n^3)$  variables.

While  $\text{LL}(G)$  for some groups of this class was of exponential size for any ordering of  $\Omega$ ,  $\phi_L(G)$  is of polynomial size: however, as a penalty, we had to use up to  $O(n^3)$  extra variables in addition to  $X(1), X(2), \dots, X(n)$ .

We now show how to construct a lex-leader formula  $\phi_L(G)$  when we use only  $O(n)$  extra variables, however, the size of  $\phi_L(G)$  in this case is  $O(n^4)$ . We achieve this reduction primarily by reducing the number of variables used in Lemma 7.1.

Let  $x_1, x_2, \dots, x_n$  be boolean variables. Define  $E[n] = E(x_1, x_2, \dots, x_n)$  ( $O[n]$ ) to be true iff an even (resp. odd) number of the variables  $x_1, x_2, \dots, x_n$  are set to 1. The following fact is well-known.

**Lemma 7.6.** There is a boolean formula of  $n$  variables, of size  $\Theta(n^2)$ , that is satisfiable iff  $E[n]$  (or  $O[n]$ ) is true.

**Proof.** Without loss of generality, assume that  $n$  is a power of 2. Otherwise pad the variable set with new variables set to 0. Proof is by induction on  $n$ . Since

we are only looking for polynomial length formulas (asymptotically polynomial) the base cases are trivial. Observe that

$$\begin{aligned} E[n] &= [O(x_1, x_2, \dots, x_{n/2}) \wedge O(x_{n/2+1}, \dots, x_n)] \\ &\quad \vee [E(x_1, x_2, \dots, x_{n/2}) \wedge E(x_{n/2+1}, \dots, x_n)] \end{aligned}$$

Similarly,

$$\begin{aligned} O[n] &= [O(x_1, x_2, \dots, x_{n/2}) \wedge E(x_{n/2+1}, \dots, x_n)] \\ &\quad \vee [E(x_1, x_2, \dots, x_{n/2}) \wedge O(x_{n/2+1}, \dots, x_n)] \end{aligned}$$

The inductive hypothesis guarantees the existence of a boolean formula of size  $\Theta(n^2)$  for each of the 4 expressions on the right hand side of the above equation (note that the boolean formula for  $O[n]$  is not simply a negation of the formula for  $E[n]$ ). This works out to be a  $\Theta(n^2)$  formula for  $E[n]$  and  $O[n]$ .  $\square$

Now this means that in Lemma 7.2, we can get formula of size  $mn^2$  with no new variables being needed. This also means that in Lemma 7.4, we can write a boolean formula  $\bar{\phi}(A, b)$  of size  $\Theta(nm^2)$  that expresses the non-solvability of  $Ax = b$ . We use this construction to replace  $\phi(W, X, i)$  by a boolean formula of size  $O(n^3)$  in Equation II.8. This means that  $N(W)$  will have size  $O(n^4)$ . The only variables that are added are  $a_k, 1 \leq k \leq n/2$  and  $c_i, 1 \leq i \leq r$  (clearly  $r \leq n$ ).

Corollary 7.7. Let  $G \leq \text{Sym}(\Omega)$  be a group with orbits of size  $\leq 2$ . Then for all orderings of  $\Omega$ , there is a lex-leader formula  $\phi'_L(G)$  of size  $O(n^4)$  defined over  $O(n)$  variables.

It is worthwhile noting that Lemma 7.6 gives the smallest possible formula for  $E[n]$ . It is proved in [9] (page 787, 797) that any formula for parity that uses the connectives  $\wedge$ ,  $\vee$  and  $\neg$  has to have size  $\Omega(n^2)$ .

## 7.2: Abelian Groups with Cyclic Orbit Projections

Let  $G = \langle s_1, s_2, \dots, s_k \rangle \leq \text{Sym}(\Omega)$  (where  $\Omega = \{1, 2, \dots, n\}$ ) be an abelian group whose orbits are (after reordering  $\Omega$  if needed)  $\Delta_i = \{n_{i-1} + 1, n_{i-1} + 2, \dots, n_i\}$  for  $1 \leq i \leq r$ , where  $n_0 = 1, n_r = n$ . Let  $m_i = |\Delta_i|$ , so  $n = \sum_i m_i$ . We assume that the orbit constituents are cyclic groups, i.e.,  $G^{\Delta_i} \cong Z_{m_i}$ . Any  $g \in G$  can be written as an  $r$ -tuple  $(g(1), g(2), \dots, g(r))$  where  $g(i)$  is the projection of  $g$  into the  $i$ -th orbit. If necessary, we reorder the points in each orbit so that we have the isomorphism  $j \in Z_{m_i} \leftrightarrow x \in G^{\Delta_i}$  where  $l^x = l + j$  ( $l$  is the first point in the orbit). For an assignment  $X$ , let the variable  $e(X, i)$  be true iff  $X$  is invariant (i.e., all 0's or 1s) on orbit  $\Delta_i$ .

Let us focus on  $\text{Pivot}(l)$  where  $l$  is the first point in each orbit and rewrite it (after introducing new variables) to a formula of polynomial size. The arguments are exactly the same for  $\text{Pivot}(i)$  when  $i$  is any point in  $\Omega$ .

Let  $l$  be the first point in orbit  $t + 1$  for some  $0 \leq t \leq r - 1$ , i.e.,  $l = n_{i-1} + 1$ .

Recall the formula  $\text{Fix}(g, X, i)$  (defined in Section 2) which is satisfiable iff  $X(j) = X(j^g)$  for all  $j < i$ . It is easy to see that  $\text{Fix}(g, X, l - 1)$  is

$$\bigwedge_{i \leq t} ((g(i) = 0) \vee e(X, i)).$$

Since abelian groups act regularly on their orbits (see Dixon[14]), any group element either moves all points of the orbit (in which case  $X$  has to be invariant on the orbit)

or it fixes all points (in which case the value of  $X$  on that orbit is irrelevant). Observe that we can simplify the above formula to the system of arithmetic equations:

$$(1 - e(X, i)) g(i) = 0 \pmod{m_i} \text{ for } i = 1, 2, \dots, t.$$

Also recall the formula  $\text{Geq}(g, X, i)$  from Section 2 which is satisfiable iff  $X(i) \geq X(i^g)$ . Because of the regularity of  $G$  on the  $(t+1)$ -th orbit, either  $g(t+1) = 0$  (i.e., it fixes all points in the orbit) in which case  $l^g = l$  and so  $X(l) = X(l^g)$  or  $g(t+1) = j$  whence  $l^g = l + j$ , so  $X(l) \geq X(l + j)$ .

Hence  $\text{Geq}(g, X, l)$  is:

$$(g(t+1) = 0) \vee \bigvee_{1 \leq j \leq m_{t+1}-1} [\{g(t+1) = j\} \wedge \{X(l+j) \rightarrow X(l)\}]$$

Recall that  $\text{Pivot}(l)$  is the following formula:

$$\bigwedge_{g \in G} [\text{Fix}(g, X, l-1) \rightarrow \text{Geq}(g, X, l)].$$

Remark:

- (i) We need to add clauses to  $\text{Pivot}(l)$  which express encode that the variable  $e(X, i)$  is true iff  $X$  is invariant on  $\Delta_i$ . This can easily be done for each  $e(X, i)$  by using at most  $n$  new variables  $X(i, j)$  which is satisfiable iff  $X(i) = X(j)$ . Thus we add the following clauses to  $\text{Pivot}(l)$ :

$$\bigwedge_{1 \leq i \leq t} \bigwedge_{n_{i-1}+2 \leq j \leq n_i} [X(n_{i-1} + 1, j) \leftrightarrow (X(n_{i-1} + 1) = X(j))]$$

$$\bigwedge_{1 \leq i \leq n} \left[ e(X, i) \leftrightarrow \bigwedge_{n_{i-1}+2 \leq j \leq n_i} X(n_{i-1} + 1, j) \right]$$

Recall that  $n_{j-1} + 1$  (resp.  $n_j$ ) is the first (resp. last) point of orbit  $\Delta_j$ .

- (ii) The variables of  $\text{Pivot}(l)$  are  $X(i)$  where  $1 \leq i \leq n$  and  $X(i, j)$  for  $1 \leq i, j \leq n$ .
- (iii) Strictly speaking  $g(t+1) = j$  is not a boolean expression (since neither  $g(t+1)$ ,  $j$  are boolean variables). However, we can easily make it a boolean variable by having a new variable  $g_{t+1, j}$  which is 1 iff  $g(t+1) = j$ . Observe that there are  $O(n^2)$  such variables  $g_{t+1, j}$ . But for ease of presentation, we decide not to rewrite the equalities.

$$\begin{aligned} \text{Pivot}(l) &= \bigwedge_{g \in G} \neg \text{Fix}(g, X, l-1) \vee \text{Geq}(g, X, l) \\ &= \neg \exists_{g \in G} \text{Fix}(g, X, l-1) \wedge \neg \text{Geq}(g, X, l) \\ &= \neg \exists_{g \in G} \left\{ \bigwedge_{i \leq t} (g(i) = 0 \vee e(X, i)) \right\} \wedge (g(t+1) \neq 0) \wedge \\ &\quad \bigwedge_{1 \leq j \leq m_{t+1}-1} (g(t+1) \neq j \vee (X(j+l) \wedge \neg X(l))) \\ &= \neg \exists_{g \in G} \bigvee_{1 \leq j \leq m_{t+1}-1} \mathcal{E}(g, j, t+1) \wedge X(j+l) \wedge \neg X(l) \end{aligned}$$

Here  $\mathcal{E}(g, j, t+1)$  is a system of  $t+1$  equations (recall that we could write the formula  $\text{Fix}(g, X, l-1)$  as a set of equations):

$$\begin{aligned} \bigwedge_{1 \leq i \leq t} (1 - e(X, i)) g(i) &= 0 \pmod{m_i} \\ g(t+1) &= j \pmod{m_{t+1}} \end{aligned}$$

Note that each  $g \in G$  is a word in the generators  $S = \{s_1, s_2, \dots, s_k\}$ , i.e.,  $g = \sum_{1 \leq \alpha \leq k} s_\alpha x_\alpha$  where  $x_\alpha \in Z$  and  $g(i) = \sum_\alpha x_\alpha s_\alpha(i) \pmod{m_i}$ .

Thus each  $\mathcal{E}(g, j, t + 1)$  is equivalent to the following system of equations:

$$\begin{aligned} \sum_{\alpha=1}^k (1 - e(X, i)) s_{\alpha}(i) x_{\alpha} &= 0 \pmod{m_i}, \quad 1 \leq i \leq t \\ \sum_{\alpha=1}^k s_{\alpha}(t + 1) x_{\alpha} &= j \pmod{m_{t+1}} \end{aligned}$$

where the unknowns are  $x_{\alpha}$ ,  $1 \leq \alpha \leq k$ .

We need a boolean formula which is satisfiable iff  $\mathcal{E}(g, j, t + 1)$  is not solvable. In other words, if  $\mathcal{E}$  defined as the following system of equations:

$$\sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{m_i}, \quad 1 \leq i \leq t + 1.$$

then we need a boolean formula which expresses the non-solvability of this system.

Each equation  $\sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{m_i}$  is solvable iff  $\sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{p^{e_i}}$  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  $\mathcal{E}_p : \sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{p^{e_i}}, 1 \leq i \leq t + 1$  is solvable for each prime  $p \mid m_i$  for some  $1 \leq i \leq t + 1$ . Note that  $\mathcal{E}_p$  might contain fewer than  $t + 1$  equations, since it might be the case that  $e_i = 0$  for some  $i$  and so we can remove the trivial equation  $\sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{1}$  from  $\mathcal{E}_p$ . Observe that the number of systems  $\mathcal{E}_p$  is  $O(n/\log n)$  (by the Prime Number Theorem, [40, ch 10]) since for some  $i$ ,  $p \mid m_i$  and  $m_i \leq n$ .

We can rewrite  $\mathcal{E}_p$  as a system of equations, where each equation is defined modulo  $p^e$  where  $e = \max\{e_i \mid 1 \leq i \leq t + 1\}$ . To do this we multiply both sides of each equation  $\sum_{1 \leq i \leq k} A(i, j)x_j = b_i \pmod{p^{e_i}}$  (where we now can assume that  $e_i \neq 0$ )

by  $p^{e-e_i}$  to get the equation:

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

We get a new system of equations  $\mathcal{E}'_p$  defined over  $Z_{p^e}$  which is solvable iff  $\mathcal{E}_p$  is solvable.

We can write each  $\mathcal{E}'_p$  as a system of the form  $A_p x = b_p$  where  $A$  is an  $n \times n$  matrix (observe that the number of generators,  $k$ , is less than  $n$  and we can pad  $\mathcal{E}'_p$  with equations where coefficients are all 0). Thus we need a “short” boolean formula  $\bar{\phi}_p = \bar{\phi}(A_p, b_p)$  which is satisfiable iff  $\mathcal{E}'_p$  is not solvable, in effect, we need an analogue of Lemma 7.4 when the equations are defined over a ring of integers  $Z_{p^e}$ .

It is proved in Theorem 7.18 that we can write a boolean formula  $\bar{\phi}_p$  of size  $O(n^2 \log p^e \log \log p^e \log \log \log p^e)$  which is satisfiable iff  $A_p x = b_p$  is not solvable. Hence

$$\begin{aligned} \text{Pivot}(l) &= \bigwedge_{1 \leq j \leq m_{t+1}-1} \neg \exists_{g \in G} [\mathcal{E}(g, j, t+1) \wedge X(j+l) \wedge \neg X(l)] \\ &= \bigwedge_{1 \leq j \leq m_{t+1}-1} [\neg \exists_{g \in G} \mathcal{E}(g, j, t+1)] \vee (X(j+l) \rightarrow X(l)) \\ &= \bigwedge_{1 \leq j \leq m_{t+1}-1} \left( \bigvee_p \bar{\phi}_p \right) \vee (X(j+l) \rightarrow X(l)) \end{aligned}$$

There are  $m_{t+1} - 1 = O(n)$  values for  $j$ . Observe that  $p^e = O(n)$  and both  $\log p = O(\log n)$  and  $e = O(\log n)$ . Hence the total size of  $\text{Pivot}(l)$  is  $O(n^4 \log \log n \log \log \log n)$  (as we observed before there are at most  $O(n/\log n)$  prime factors we have to consider). An identical asymptotic bound for  $\text{Pivot}(l)$  when  $l$  is not the first point of the orbit can be proved in a similar fashion. Thus we have proved:

**Lemma 7.8.** Let  $G \leq \text{Sym}(\Omega)$  be an abelian group with cyclic projections in



each orbit. Then one can find a reordering of  $\Omega$  in polynomial time so that there is a lex-leader formula of size  $O(n^5 \log \log n \log \log \log n)$  (where  $|\Omega| = n$ ) defined over a polynomial (in  $n$ ) number of variables.

### 7.3: Non-solvability over $Z_m$

Our main reference for this subsection is [33]. The following theorem is classical.

**Theorem 7.9.** [Smith Normal Form, [33], pg 26] Let  $A$  be an  $n \times n$  matrix over  $Z_m$ . Then there exist invertible matrices  $U$  and  $V$  over  $Z_m$  such that  $UAV = D$  where  $D$  is a diagonal matrix of the form

$$\text{diag}(s_1, s_2, \dots, s_r, 0, 0, \dots, 0)$$

where  $r > 0$  and  $s_i \neq 0$  for all  $1 \leq i \leq r$

In [33], this theorem is actually stated for a principal ideal domain (such as  $Z$ ). But if a matrix  $A$  over  $Z$  can be diagonalized by invertible matrices  $U$  and  $V$  over  $Z$ , then  $\hat{A} = A \bmod m$  can be diagonalized by  $\hat{U}$  and  $\hat{V}$ . It is easy to see that  $\hat{U}$  and  $\hat{V}$  are also invertible: this is a simple consequence of the fact that a matrix is invertible over  $Z$  (actually over any commutative ring  $R$  with identity) iff its determinant is a unit in  $Z$  (or the ring  $R$ ), i.e., is  $+1$  or  $-1$ . Hence the determinant is a unit in  $Z_m$  as well.

We first make a simple observation.

**Lemma 7.10.** The linear congruence  $ax = b$  has a solution in  $Z_m$  iff  $(a, m) \mid b$ .

**Proof.** ( $\Rightarrow$ ) Suppose the equation  $ax = b$  has a solution in  $Z_m$ , then there is

an integer  $q > 0$  such that  $ax = b + qm$ . Since  $(a, m) \mid a$  and  $(a, m) \mid m$ , it must be the case that  $(a, m) \mid b$ .

( $\Leftarrow$ ) Suppose  $(a, m) \mid b$ . Set  $a_1 = a/(a, m)$ ,  $b_1 = b/(a, m)$ ,  $m_1 = m/(a, m)$ . The congruence  $a_1x = b_1$  has the solution  $x = a_1^{-1}b_1$  in  $Z_{m_1}$  (since  $(a_1, m_1) = 1$ , it is invertible in  $Z_{m_1}$ ). It is easy to see that  $a_1^{-1}b_1 + q'm_1 \pmod{m}$  is a solution of  $ax = b \pmod{m}$ .  $\square$

In the following discussion, let  $A, b$  be  $n \times n$ ,  $n \times 1$  matrices over  $Z_m$ . The following theorem is classical, we give an alternate algorithmic proof.

**Theorem 7.11.** The system of equations  $\mathcal{E} : Ax = b$  over  $Z_m$  has a solution iff the  $Z_m$ -module spanned by the row vectors of  $[A \ b]$  does not contain a vector of the form  $(0, 0, \dots, 0, \alpha) \in Z_m^{n+1}$  where  $\alpha \neq 0$ .

**Proof.** ( $\Rightarrow$ ) If  $\mathcal{E}$  has a solution, then clearly none of the above vectors can be in the row space of  $[A \ b]$ .

( $\Leftarrow$ ) Suppose  $\mathcal{E}$  does not have a solution. Let  $U, V$  be invertible matrices such that  $UAV = D$  is in Smith Normal Form (as stated in Theorem 7.9). Let  $V^{-1}x = y$  and  $Ub = d$ . Then  $Ax = b$  has a solution iff  $UAVV^{-1}x = UB$  has a solution, i.e., the system  $Dy = d$  has a solution. If the matrix  $[D \ d]$  has a row of the form  $(0, 0, \dots, 0, \alpha)$  where  $\alpha \neq 0$  then we are done as  $[UA \ d]$  must then also contain the row  $(0, 0, \dots, 0, \alpha)$ . So suppose that  $[D \ d]$  does not have such a row. The system of equations  $Dy = d$  has a solution iff  $s_i y_i = d_i$ ,  $1 \leq i \leq r$  has a solution in  $Z_m$ , i.e., iff  $(s_i, m) \mid d_i$  for each  $1 \leq i \leq r$  (Lemma 7.10). Thus if  $\mathcal{E}$  does not have a solution, there is some  $i$  such that  $(s_i, m) \nmid d_i$ . In particular, this means that  $(s_i, m) \neq 1$  and  $s_i \neq d_i$ . Thus there is some row of the adjoint matrix  $[UA \ d]$  of the form  $(c_1 s_i, c_2 s_i, \dots, c_n s_i, d_i)$  where  $c_i \in Z_m$ .

Since  $(s_i, n) \neq 1$ , there is a  $t_i \in Z_m$  such that  $s_i t_i = 0$ . Multiplying this row by  $t_i$  we get  $(0, 0, \dots, 0, d_i t_i)$  (observe that  $d_i t_i \neq 0$ ).  $\square$

**Corollary 7.12.** The system of equations  $\mathcal{E} : Ax = b$  over  $Z_m$  has a solution iff none of the vectors

$$\{(0, 0, \dots, 0, m/p) \in Z_m^{n+1} \mid \text{where } p \mid m, p \text{ prime} \}$$

are in the  $Z_m$  module spanned by the row vectors of  $[A \ b]$ .

**Proof.** ( $\Rightarrow$ ) If  $\mathcal{E}$  has a solution, then clearly none of the above vectors can be in the row space of  $[A \ b]$ .

( $\Leftarrow$ ) Suppose  $\mathcal{E}$  does not have a solution. Then from Theorem 7.11 we know that there is a vector  $(0, 0, \dots, 0, \alpha) \in Z_m^{n+1}$  where  $\alpha \neq 0$  which is in the row space of  $[A \ b]$ . Let us suppose that some prime power  $p^i$  divides  $\alpha$  where  $p \nmid m$ . Then  $p^i$  has an inverse  $p^{-i}$  in  $Z_m$ , we multiply the entire row by this inverse to remove the  $p$ -factor from  $\alpha$ . Thus we may assume that we have a row of the form  $(0, 0, \dots, \beta)$  where  $\beta = \prod_{1 \leq i \leq k} p_i^{e_i}$  where  $p_i, 1 \leq i \leq k$  are all the prime divisors of  $m$ . Now multiplying by appropriate prime powers we can change  $\beta$  to  $m/p$  for some prime divisor of  $m$ . This means that the vector  $(0, 0, \dots, 0, m/p)$  is in the linear span of  $[A \ b]$ .  $\square$

We thus have the following useful corollary which expresses non-solvability in terms of solvability (recall we proved a similar theorem in the vector space situation, Lemma 7.3).

**Corollary 7.13.** Let  $\mathcal{E} : Ax = b$  be a  $n \times n$ -system of equations over  $Z_{p^e}$ . Then

there is a  $(n + 1) \times n$ -system of equations  $Cx = d$  over  $Z_{p^e}$  which is solvable iff  $\mathcal{E}$  is not solvable.

Proof. Corollary 7.12 says that  $\mathcal{E}$  is not solvable iff the vector  $(0, 0, \dots, 0, p^{e-1})$  is in the row space of  $[A \ b]$ . This can be expressed as a system of equations  $Cx = d$  module  $p^e$  where  $C$  is a  $(n + 1) \times n$  matrix.  $\square$

Thus the non-solvability of a system of equations over  $Z_{p^e}$  is equivalent to the *solvability* of another system of equations over  $Z_{p^e}$ .

We now prove an analogue of Lemma 7.2 and Lemma 7.1 for equations defined over  $Z_{p^e}$ .

Let  $m = p^e$  in the following discussion. We need some additional results about computation in  $Z_m$ , specifically about boolean circuits computing additions and multiplications.

Recall that a boolean circuit  $C$  is a directed acyclic graph (DAG) whose vertices are labelled with the names of Boolean connectives  $\wedge, \vee, \neg$  (the logic gates) or variables (inputs). Each boolean circuit computes a boolean function  $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$  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 easy to prove:

Lemma 7.14. Let  $C$  be a circuit computing a boolean function  $f(x_1, x_2, \dots, x_m) = (y_1, y_2, \dots, y_n)$ . Then there is a boolean  $\mathcal{F}(C)$  formula of size  $O(s(C))$  defined over  $x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n$  and a linear (in  $s(C)$ ) number of extra variables whose mod-

els are such that the value of  $(y_1, y_2, \dots, y_m)$  is  $f(\alpha_1, \alpha_2, \dots, \alpha_m)$  where  $\alpha_i$  is the value of  $x_i$  in the model.

Proof. We label the edges of the circuit in stages. Initially the edges from input variables and the edges to the output variables get the corresponding variable label. If an output variable is directly connected to an input variable, we deal with this edge separately. For each gate  $v$  in  $C$ , let  $x, y$  be the labels of the inputs and let  $z$  be the output label (each outgoing edge gets the label  $z$ ). A clause in  $\mathcal{F}(C)$  will be  $x \circ y = z$  where  $\circ$  is  $\vee$  or  $\wedge$  depending on the gate. Observe that each output edge of the gate will get the label  $z$ . If the gate is a NOT( $\neg$ )-gate then we add a clause  $z = \neg x$  where  $z$  is output label and  $x$  is the input label. For each edge  $(x_i, y_j)$  where  $x_i$  is an input variable of  $C$  and  $y_j$  is an output variable, we add the clause  $x_i = y_j$ . Thus the size of the formula is  $O(s(C) + m + n) = O(s(C))$ .  $\square$

We need the following lemma which we quote from [36] (chapter 2):

Lemma 7.15.

1. The addition function  $f_{\text{add}}^{(n)} : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{n+1}$  for  $n$ -bit binary numbers can be computed by a circuit  $C$  with  $s(C) = O(n)$ .
2. The binary multiplication function  $f_{\text{mult}}^{(n)} : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$  for  $n$ -bit binary numbers can be computed by a circuit  $C$  with  $s(C) = O(n \log n \log \log n)$ . (Also see [38])
3. The reciprocal function  $f_{\text{recip}}^n : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  for  $n$ -bit binary numbers can be computed by a circuit of size  $O(n \log n \log \log n)$ .

Observe that as a consequence of Lemma 7.15, all computation in  $Z_{p^e}$  can be done by circuits of size  $O(\log p^e \log \log p^e \log \log \log p^e)$  (including division which is a reciprocal followed by a multiplication).

Let  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  and  $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_n)$  be 2  $n$ -bit vectors from  $Z_m^n$ . Also, let  $b \in Z_m$ . Let  $E (= E(\mathbf{x}, \epsilon, b))$  denote the equation  $\sum_{i=1}^n \epsilon_i x_i = b$  over  $Z_m$ .

Lemma 7.16. There exists a boolean formula  $\phi$  of size

$$O(n \log m \log \log m \log \log \log m)$$

defined over  $O(n \log m \log \log m \log \log \log m)$  variables which is satisfiable iff  $E$  is solvable.

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

$$\mu_i = \epsilon_i x_i, 1 \leq i \leq n$$

$$\gamma_1 = \mu_1,$$

$$\gamma_i = \gamma_{i-1} + \mu_i, 2 \leq i \leq n - 1$$

$$b = \gamma_{n-1} + \mu_n$$

For each equation above, let the rhs represent a function computed by a circuit  $C$  (this circuit does computation modulo  $m$ ) and assume  $y_1, y_2, \dots, y_{\lceil \log m \rceil}$  are the output bits for  $C$ . Let the variable on the lhs be  $x$ , represented by bits  $x_1, x_2, \dots, x_{\lceil \log m \rceil}$ . 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 7.14. The conjunction of formulas for each equation is our desired  $\phi$ . Correctness and size estimates are easy to prove.  $\square$

The next lemma now follows.

**Lemma 7.17.** Let  $Ax = b$  be a system of equations over  $Z_{p^e}$  where  $A$  is an  $n \times n$  matrix. Then there is a boolean formula  $\phi(A, b)$  of size

$$O(n^2 \log p^e \log \log p^e \log \log \log p^e)$$

defined over  $O(n^2 \log p^e \log \log p^e \log \log \log p^e)$  variables which is satisfiable iff  $Ax = b$  is solvable.

We can now write a boolean formula expressing non-solvability over  $Z_m$ .

**Theorem 7.18.** Let  $\mathcal{E} : Ax = b$  be a system of equations over  $Z_{p^e}$ ,  $A$  is a  $n \times n$  matrix. There exists a boolean formula  $\bar{\phi}(A, b)$  of size

$$O(n^2 \log p^e \log \log p^e \log \log \log p^e)$$

defined over  $O(n^2 \log p^e \log \log p^e \log \log \log p^e)$  variables which is satisfiable iff  $\mathcal{E}$  is not solvable.

**Proof.** Corollary 7.13 reduces non-solvability of  $\mathcal{E}$  to solvability of another system  $Cx = d$  modulo  $p^e$ . Hence  $\mathcal{E}$  is solvable iff  $\bar{\phi}(A, b) = \phi(C, d)$  is satisfiable. The bound now follows.  $\square$

#### 7.4: Abelian Groups: General Case

In the general case, the projection of abelian  $G \leq \text{Sym}(\Omega)$  in each orbit is isomorphic to  $Z_{n_1} \oplus Z_{n_2} \oplus \dots \oplus Z_{n_k}$ . In this subsection, we consider this general case.

Assume (as before) that  $G = \langle s_1, s_2, \dots, s_k \rangle \leq \text{Sym}(\Omega)$  (where  $\Omega = \{1, 2, \dots, n\}$ ) and the orbits are (after reordering  $\Omega$  if needed)  $\Delta_i = \{n_{i-1} + 1, n_{i-1} + 2, \dots, n_i\}$  for  $1 \leq i \leq r$ , where  $n_0 = 1, n_r = n$ . Let  $m_i = |\Delta_i|$ , so  $n = \sum_i m_i$ . We also assume that  $G^{\Delta_i} \cong Z_{r_i,1} \oplus Z_{r_i,2} \oplus \dots \oplus Z_{r_i,f_i}$  so that  $\prod_{j=1}^{f_i} r_{i,j} = m_i$ . Here  $f_i$  is the number of direct summands in  $G^{\Delta_i}$ .

Any  $g \in G$  can be written as an  $\sum_{i=1}^r f_i$ -tuple

$$(g(1,1), g(1,2), \dots, g(1,f_1), g(2,1), \dots, g(2,f_2), \dots, g(r,1), \dots, g(r,f_r))$$

where  $g(i,j)$  is the projection of  $g$  into the  $j$ -th direct summand of  $G^{\Delta_i}$ .

Since  $G$  is regular on  $\Delta_i$ , we can identify elements from  $G^{\Delta_i}$  with integers from  $\{0, 1, \dots, m_i - 1\}$ , i.e., we identify the group element  $x$  which maps  $l^x$  to  $l + j$  with the integer  $j$  in  $\{0, 1, \dots, m_i - 1\}$ . Thus every element of  $G^{\Delta_i}$  corresponds to an integer from  $\{0, 1, \dots, m_i - 1\}$  and because  $G^{\Delta_i}$  is also a direct product, each such integer  $j$  corresponds to a unique  $f_i$ -tuple  $(j(1), j(2), \dots, j(f_i))$ , where each coordinate  $j(k)$  is in the ring  $Z_{r_i,k}$ .

The arguments that write  $\text{Pivot}(l)$  in terms of a system of equations  $\mathcal{E}(g, j, t+1)$  do not depend on the structure of the orbit constituents of  $G$ , where  $l$  is the first point of orbit  $t+1$ . So as before:

$$\begin{aligned} \text{Pivot}(l) &= \bigwedge_{g \in G} \neg \text{Fix}(g, X, l-1) \vee \text{Geq}(g, X, l) \\ &= \neg \exists_{g \in G} \text{Fix}(g, X, l-1) \wedge \neg \text{Geq}(g, X, l) \\ &= \neg \exists_{g \in G} \bigvee_{1 \leq j \leq m_{t+1}-1} \mathcal{E}(g, j, t+1) \wedge X(j+l) \wedge \neg X(l) \end{aligned}$$



where  $\mathcal{E}(g, j, t + 1)$  is a system of  $t + 1$  equations:

$$\begin{aligned} (1 - e(X, i)) g(i, j) &= 0 \pmod{r_{i,j}} \text{ for } 1 \leq j \leq f_i, 1 \leq i \leq t \\ g(t + 1, i) &= j(i) \pmod{r_{t+1,i}} \text{ for } 1 \leq i \leq f_{t+1} \end{aligned}$$

Each  $g \in G$  is a word in the generators  $S = \{s_1, s_2, \dots, s_k\}$ , i.e.,  $g = \sum_{1 \leq \alpha \leq k} s_\alpha x_\alpha$  where  $x_\alpha \in Z$  and  $g(i, j) = \sum_\alpha x_\alpha s_\alpha(i, j) \pmod{r_{i,j}}$ .

Thus the system of equations  $\mathcal{E}(g, j, t + 1)$  breaks down into a system of equations, each equation modulo  $r_{i,j}$  where  $1 \leq j \leq f_i, 1 \leq i \leq t + 1$ . As before, we can write a formula which is satisfiable iff  $\mathcal{E}(g, j, t + 1)$  is not solvable. The rest of the arguments are similar. Thus we can improve Lemma 7.8 to the following theorem:

**Theorem 7.19.** Let  $G \leq \text{Sym}(\Omega)$  be an abelian group. Then there is a canonical ordering of  $\Omega$  such that there is a lex-leader formula of size  $O(n^5 \log \log n \log \log \log n)$  ( $|\Omega| = n$ ) defined over a polynomial (in  $n$ ) number of variables. Furthermore, such an ordering can be found in polynomial time.

## 8. Lex-Leader Formulas for $\mathcal{P}_d$ Groups

Let  $G = \langle R \rangle \leq \text{Sym}(\Omega)$  be a permutation group on  $n$  points  $\Omega = \{1, 2, \dots, n\}$  where the orbits of  $\Omega$  are  $\Delta_1, \Delta_2, \dots, \Delta_m$  where  $\Delta_1 = \{1, 2, \dots, i_1\}$  and  $\Delta_j = \{i_{j-1} + 1, i_{j-1} + 2, \dots, i_j\}$  (we reorder  $\Omega$  if necessary to ensure this). Also assume that the size of the projection  $|G^{\Delta_i}| \leq n^d (= \gamma \text{ say})$ , so that  $G$  is a  $\mathcal{P}_d$  group.

We assume that the set  $R$  is a special set of strong generators of  $G$  – these are the coset representatives  $R_i$  of  $G_i$  in  $G_{i-1}$  where  $G_i$  is the pointwise stabilizer of the first  $i$  points. That is  $R = \cup_{i=1}^{n-1} R_i$  and  $R_i$  has size  $\leq \delta$  where  $\delta$  is the size of the largest orbit of  $G$  in  $\Omega$ . Note that  $|R| \in O(n^2)$ .

We also assume that we have at hand a small set  $U$  of generators of  $G$  where  $|U| = O(\log |G|) = O(\log(\gamma^m)) = O(m \log \gamma)$ .

Let  $X \in 2^\Omega$  and let  $X \upharpoonright_{\Delta_j}$  denote the projection of  $X$  into  $\Delta_j$ . The permutation  $g$  fixes a string  $X \upharpoonright_{\Delta_i}$  if  $X(j) = X(j^g)$  for all  $j \in \Delta_i$ .

For  $X \in 2^{[n]}$ , let  $G_i^X = \{g \in G \mid g \text{ fixes } X \upharpoonright_{\Delta_1}, X \upharpoonright_{\Delta_2}, \dots, X \upharpoonright_{\Delta_i}\}$ .

Let  $\mathcal{C}(X)$  denote the chain of subgroups

$$\begin{aligned} G &\geq G_1^X \geq G_2^X \cdots \geq G_m^X \geq (G_m^X \cap G_1) \\ &\geq (G_m^X \cap G_2) \cdots \geq (G_m^X \cap G_{n-1} = 1) \end{aligned} \quad (\text{II.9})$$

Let  $S_i$  for  $1 \leq i \leq m$  ( $T_i$  for  $1 \leq i \leq n-1$ ) be a complete set of coset representatives of  $G_i^X$  in  $G_{i-1}^X$  for  $1 \leq i \leq m$  ( $G_m^X \cap G_i$  in  $G_m^X \cap G_{i-1}$  for  $1 \leq i \leq n-1$ , resp) where  $G_0 = G$ . Each  $S_i$  ( $T_i$ ) has size at most  $\gamma$  ( $\delta$  resp).

We now define the individual pieces of the lex-leader formula which we paste together to build a lex-leader formula. These formulas and their sizes are summarized in the next table.

Perm( $\pi$ )	$\{\pi(i, j) \mid 1 \leq i, j, \leq n\}$ represents a permutation	$O(n^2)$
Equal( $y, x$ )	$x = y$	$O(n^2)$
Product( $x, y, z$ )	$z = x y$	$O(n^3)$
Member( $L, \pi$ )	$\pi \in L$	$O( L  n^2)$

Table 1: Boolean Formulas for Permutation Groups

Perm( $\{\pi(i, j) \mid 1 \leq i, j \leq n\}$ ) is a boolean formula of size  $O(n^2)$  whose satisfying assignments correspond to permutations in  $\text{Sym}(\Omega)$ . To simplify notation, we write the permutation  $\pi$  as a shorthand for the set of variables  $\{\pi(i, j) \mid 1 \leq i, j \leq n\}$ .

Let  $x_1, x_2, \dots, x_n$  be  $n$  boolean variables. Let  $E_1(x_1, x_2, \dots, x_n)$  be true exactly when only one of the  $n$  variables is set true. We write a boolean formula  $\phi_1(x_1, x_2, \dots, x_n, \mu_1, \mu_2, \dots, \mu_n)$  of size  $O(n^2)$  which is satisfiable iff  $E_1(x_1, x_2, \dots, x_n)$  is true. The formula is:

$$\begin{aligned} & \bigvee_{1 \leq i \leq n} x_i \\ & \mu_1 = x_1 \\ & \mu_j = \mu_{j-1} \oplus x_j \\ & \mu_{j-1} \wedge x_j \rightarrow \mu_j \end{aligned}$$

$\text{Perm}(\pi)$  is a formula of size  $O(n^2)$  and is the conjunction of the following lines:

$$\begin{aligned} & \bigwedge_{1 \leq i \leq n} \phi(\pi(i, 1), \pi(i, 2), \dots, \pi(i, n), \mu(i, 1), \mu(i, 2), \dots, \mu(i, n)) \\ & \bigwedge_{1 \leq j \leq n} \phi(\pi(1, j), \pi(2, j), \dots, \pi(n, j), \tau(1, j), \tau(2, j), \dots, \tau(n, j)) \end{aligned}$$

Define  $\text{Equal}(y, x)$  to be the formula  $\bigwedge_i \bigwedge_j (x(i, j) = y(i, j))$ .

We also define  $\text{Product}(x, y, z)$  which is satisfiable iff  $z = xy$ :

$$\bigwedge_{i,j,k} x(i, j) \wedge y(j, k) \rightarrow z(i, k).$$

Let  $\text{Member}(L, \pi)$  be a boolean formula which is satisfiable when  $\pi \in L$ . Let  $|L| = l$  and let  $L = \{\pi_1, \pi_2, \dots, \pi_l\}$ , then the desired formula is:

$$\bigvee_{1 \leq i \leq l} \text{Equal}(\pi, \pi_i).$$

$\text{Sift}(\pi, X, S \cup T)$  is satisfiable iff  $\pi$  sifts through the chain of coset representatives  $S = \cup S_i$  and  $T = \cup T_i$  in  $\mathcal{C}(X)$ ,

The formula for  $\text{sift}(\pi, X, S, T)$  is:

$$\begin{aligned} & \bigwedge_{2 \leq i \leq m+n-1} \pi_i = \text{Product}(\pi_{i-1}, e_i) \\ & \text{Member}(S_1, \pi_1) \\ & \bigwedge_{2 \leq i \leq m} \text{Member}(S_i, e_i) \\ & \bigwedge_{m+1 \leq i \leq m+n-1} \text{Member}(T_{i-m}, e_i) \\ & \bigwedge_{1 \leq i \leq m+n-1} \text{Perm}(e_i) \\ & \text{Equal}(\pi, \pi_{m+n-1}) \end{aligned}$$

The size of sift is  $O(m\gamma n^2 + mn^3 + \delta n^3)$ .

We can similarly define a formula  $\text{sift-chain}(\pi, R)$  of size  $O(n^4)$  which is satisfiable if the permutation  $\pi$  sifts through the strong generators  $R$ .

Now we write a formula  $\text{STGEN}(R, S, T, X)$  which expresses the fact that the set of generators  $S \cup T$  are strong generators for  $G$ .

$$\begin{aligned} & \bigwedge_{x \in S \cup T} \text{sift-chain}(x, R) \\ & \bigwedge_{u \in U} \bigwedge_{y \in S \cup T} \text{Product}(y, u, z) \wedge \text{sift}(z, X, S, T) \end{aligned}$$

The first line ensures that  $\langle S \cup T \rangle \leq G$ . The second line ensures that  $G \leq \langle S \cup T \rangle$  (observe that we took elements from  $U$  for sift in this line, because  $U$  is of potentially smaller size than  $R$ ).

Clearly the size of the last line will dominate the size of STGEN. STGEN is of size  $O((\gamma m + \delta n)(m \log \gamma) \times (\text{size of sift})) = O((\gamma m + \delta n)(m \log \gamma)(m\gamma n^2 + mn^3 + \delta n^3))$ .

We define a formula  $\text{StringLeader}(X, \Delta, s)$  which expresses the fact that

$X \geq (^sX)$ . We make the assumption that  $s$  stabilizes  $\Delta$ . For  $i, j \in \Delta$ , let  $X_{i,j}$  be 1 iff  $X(i) = X(j)$ . StringLeader is the following formula of size  $O(n^4)$ .

$$\bigwedge_{i,j \in \Delta} (X_{i,j} \Leftrightarrow (X(i) = X(j))) \quad \wedge \\ \bigwedge_{i \in \Delta} \left[ \bigwedge_{j,k < i} (X_{j,k} \vee \neg s(j,k)) \right] \rightarrow [s(i,i) = 1 \vee \bigvee_{l \in \Delta} \{s(i,l) \wedge (X(l) \rightarrow X(i))\}]$$

It is easy to write a formula  $LL(X, \Delta, S)$  which expresses the fact that  $\forall s \in S, X(\Delta) \geq X(\Delta)^s$ . We assume that  $\Delta$  is stabilized by the set of permutations  $S$ . This formula of size  $O(|S|n^4)$  is

$$\bigwedge_{s \in S} \text{StringLeader}(X, \Delta, s)$$

We can now write a formula  $LLG(X, G)$  which expresses the fact that  $X \geq X^g$  for all  $g \in G$ :

$$\text{STGEN}(R, S, T, X) \wedge \bigwedge_{1 \leq i \leq m} LL(X, \Delta_i, S_i)$$

Hence the total length of the formula is  $O((\gamma m + \delta n)(m \log \gamma)(m\gamma n^2 + mn^3 + \delta n^3) + \gamma mn^4)$ .

Theorem 8.1.

- (i) Let  $G \leq \text{Sym}(\Omega)$  be a  $\mathcal{P}_d$  group where  $d > 1$ . Then there is a canonical ordering of  $\Omega$  such that there exists a lex-leader formula for  $G$  of size  $O(d n^{2d+5} \log n)$ , where  $|\Omega| = n$ . Furthermore, such an ordering can be found in polynomial time.
- (ii) Let  $G \leq \text{Sym}(\Omega)$  be a group whose largest orbit has size  $c$  (a fixed constant).

Then there is a canonical reordering of  $\Omega$  such that there is a lex-leader formula for  $G$  of size  $O(n^6)$ . Such a reordering can also be found in polynomial time.

Remark: The test for “strong generators” that we encode in the lex-leader formula is expensive, contributing a factor of  $O(n^{2d})$  to the size. It is conceivable that through cheaper tests for strong generators, e.g., Sims’s “verify” test (see [20]), the factor can be reduced to  $O(n^d)$ .

It is no surprise that solving the “bounded orbit” case in such generality in part (ii) of the above theorem gives us much worse results than Theorem 7.5: where we got a formula of size  $O(n^3)$  and now we get a formula of size  $O(n^6)$ . Another point to note: we need only consider  $\mathcal{P}_d$  groups where  $d > 1$  otherwise the entire group is polynomially bounded and an exhaustive enumeration would give all the group elements. Any lex-leader formula (for example  $LL(G)$ ) would work in that situation.

## CHAPTER III

## FAULT TOLERANCE IN BOOLEAN SATISFIABILITY

1. Definitions and Notations

The intended objects of study are strings over  $\{0, 1\}$  of some specified length  $n$ . The basic operations on these strings are bit flips (negations): changing a specified bit from a one to a zero or the reverse. However, for the purposes of this paper we have found it convenient to denote these objects as sets. A string  $\alpha$  of length  $n$  represents a subset  $X$  of  $\{1, 2, \dots, n\}$  as its *incidence vector* (or characteristic sequence): the  $i$ th bit of  $\alpha$  (denoted by  $\alpha(i)$ ) is 1 iff  $i \in X$ . In this context, flipping a bit of  $\alpha$  means taking the symmetric difference of  $X$  with a singleton set,  $X \Delta \{i\}$ . In this manner we are able to describe a series of bit flips themselves as a set, simplifying the descriptions of our proofs.

Let  $[n]$  refer to a set of  $n$  elements  $\{1, 2, \dots, n\}$  and let  $2^{[n]}$  refer to its power set. Let  $\binom{[n]}{k}$  ( $\binom{[n]}{\leq k}$ ) denote the family of  $k$ -element ( $\leq k$  element) subsets of  $[n]$ . For  $S \subseteq [n]$ , we define the operator  $\delta_S : 2^{[n]} \rightarrow 2^{[n]}$  as follows:  $\delta_S(X) = X \Delta S$ . When  $S = \{i\}$  or  $S = \{i, j\}$ , we write  $\delta_i$  (resp.  $\delta_{ij}$ ) instead of  $\delta_{\{i\}}$  (resp.  $\delta_{\{i, j\}}$ ). Note that  $\delta_j(\delta_i(X)) (= \delta_i(\delta_j(X)))$  and if  $\delta_{ij}(X) = Y$  then  $\delta_{ij}(Y) = X$ .

Given a family of subsets  $\mathcal{F}$  of  $[n]$  let  $\mathcal{F}_k$  ( $\mathcal{F}_{\leq k}$ ) denote the number of sets in  $\mathcal{F}$  with  $k$  elements ( $\leq k$  elements).

Let  $\phi$  be a boolean formula of  $n$  variables  $[n]$ . An assignment  $X : [n] \rightarrow \{0, 1\}$

is called a model if  $X$  makes  $\phi$  true. We shall, as discussed, also interpret  $X$  as an incidence vector of a subset in  $2^{[n]}$ . In all our discussions on boolean formulas, we make the assumption that every variable appears in both positive and negative literals.

Definition. A model  $X$  of  $\phi$  is called a  $\delta(r, s)$ -model if  $\forall R \in \binom{[n]}{\leq r}$ , there exists  $S \in \binom{[n]}{\leq s}$ , such that  $R \cap S = \emptyset$ , and  $\delta_{R \cup S}(X)$  is a model of  $\phi$ . We view  $r$  and  $s$  as fixed constants unless otherwise mentioned.

In other words,  $X$  is a  $\delta(r, s)$ -model iff for every bit flip (called a "break") of up to  $r$  coordinates in the incidence vector of  $X$ , either no repair is needed (i.e.  $\delta_R(X)$  is a model) or there is a disjoint set of up to  $s$  bits that can be flipped to get a model of  $\phi$ .

In this paper we shall be primarily concerned with  $\delta(1, 1)$ -models, which we shall simply call  $\delta$ models. Define  $\Phi(r, s)$  to be the set of boolean formulas which have  $\delta(r, s)$ -models.

We can define degrees of fault tolerance by requiring that  $\delta$ models are themselves repairable under further breaks. This allows us to view  $\delta$ models as a generalized notion of models of boolean formulas.

Let  $\phi$  be a boolean formula. Then  $\delta^0(r, s)$ -models are just models of  $\phi$ :

$$\delta^0(r, s) = \{X \mid X \text{ is a model of } \phi\}$$



and we define  $\delta^k(r, s)$ -models inductively:

$$\delta^k(r, s) = \{X \in \delta^{k-1}(r, s) \mid \forall R \in \binom{[n]}{\leq r}, \exists S \in \binom{[n]}{\leq s}, \\ R \cap S = \emptyset \text{ and } \delta_{R \cup S}(X) \in \delta^{k-1}(r, s)\}$$

We define  $\Phi^k(a, b)$  to be the family of boolean formulas which have a  $\delta^k(a, b)$ -model. We define  $\Phi^*(r, s) = \bigcap_{i=0}^{\infty} \Phi^i(r, s)$  and we denote the corresponding models  $\delta^*(r, s) = \bigcap_i \delta^i(r, s)$ ,  $\delta^*$ -models when  $r = s = 1$ . These are models of a boolean formula that remain models under *any* sequence of breaks and repairs.

Observe that if  $\phi$  has a  $\delta^*$ -model then there is a family  $\mathcal{M}$  of models of  $\phi$  such that for each model in  $\mathcal{M}$ , every break is repairable in such a way that the repaired string is in  $\mathcal{M}$ . In other words,  $\mathcal{M}$  consists of  $\delta^*$ -models related by breaks and repairs. We shall call  $\mathcal{M}$  a weak stable set of models. It is clear that  $\phi \in \Phi^*$  iff it has a stable set of models.

Define the Hamming distance  $d(x, y)$  between two  $n$ -bit vectors to be the number of coordinates where they differ. Observe that our definition of  $\delta^*$ -models allows models in a weak stable set to be at (Hamming) distance 1 from each other. We shall call stable families which do not have any two vectors at distance one from each other stable families (see Section 4 for formal definitions).

In Section 2 and Section 3, we look at the computational complexity of finding  $\delta(r, s)$ -models. In Sections 4, 5, 6, we study the structure of stable families.

## 2. Complexity of Finding $\delta$ Models

We now study the computational complexity of finding  $\delta$ -models for general boolean formulas.

Consider the following decision question:

**Problem:**  $\Phi(r, s)$

INSTANCE: Boolean formula  $\phi$ .

QUESTION: Does  $F$  have a  $(r, s)$   $\delta$ model ?

The notion of  $\delta$ models was defined in [18], where  $\Phi(r, s)$  was also proved NP-complete. We include the proof below.

**Theorem 2.1.** [18]  $\Phi(r, s)$  is NP-complete

**Proof.** Reduction from SAT. Let  $\phi$  be an instance of SAT, where  $\phi$  is a boolean formula over  $n$  variables  $[n]$ . Construct the formula  $\phi' = \phi \vee (n + 1)$  where  $(n + 1)$  is a new variable. We claim that  $\phi$  is satisfiable iff  $\phi'$  has a  $\delta(r, s)$ -model. Suppose  $\phi$  is satisfiable: let  $X$  be a model. Extend  $X$  to a model  $X'$  of  $\phi'$  by setting variable  $(n + 1)$  to 0. We claim  $X'$  is a  $\delta(r, s)$ -model. Let us break any set of up to  $r$  bits in  $X'$ . If that break set includes the  $(n + 1)$ -th coordinate, we do not need any repairs. If it doesn't, we can repair by flipping  $X'(n + 1)$ : hence  $X'$  is a  $\delta(r, s)$ -model. Now suppose that  $\phi'$  has a  $\delta(r, s)$ -model  $X'$ . Then  $\phi'$  must have a model with  $X(n + 1) = 0$ : if  $X'$  has  $X'(n + 1) = 0$  then that is the desired model, otherwise flip  $X(n + 1)$ , we are guaranteed a repair to another model now with  $X(n + 1) = 0$ . The restriction of that model to  $[n]$  gives us a model for  $\phi$ . Hence  $\Phi(r, s)$  is NP-hard.

Observe that  $\Phi(r, s)$  is in NP : a NDTM needs to guess an assignment and check that it is indeed a  $\delta(r, s)$ -model. Since  $r$  and  $s$  are fixed, there are  $O(n^r)$  possible break sets and  $O(n^s)$  possible repair sets, thus checking whether the guessed assignment is a  $\delta(r, s)$ -model is in polynomial time.  $\square$

We can also identify  $\Phi^k(1, 1)$  with the natural decision question: given a boolean

formula, does it belong to the family  $\Phi^k(1, 1)$ ? Since an NDTM can guess a stable set of models (which could be of exponential size) and check that it is indeed a stable set of models, we have the following:

Lemma 2.2.  $\Phi^*(1, 1) \in \text{NEXP}$ .

If the stable set had a polynomial description, then the NDTM would just use polynomial space. We wonder whether  $\Phi^*(1, 1)$  is complete for NEXP. We can prove the following (weaker) result:

Theorem 2.3.  $\Phi^*(1, 1)$  is NP-hard.

Proof. We use the same reduction as in Theorem 2.1. Given an instance of SAT, a boolean formula  $\phi$  over  $n$  variables  $[n]$ , we construct  $\phi' = \phi \vee (n + 1)$  with  $n + 1$  being a new variable. Suppose  $\phi$  has a model  $X$ . We construct a stable set of models  $\mathcal{M}$  of  $\phi'$ . If  $Y \subseteq [n + 1]$ , let  $Y_n$  be the projection of the incidence vector of  $Y$  into the first  $n$  coordinates  $[n]$ .

$$\mathcal{M} = \{Y \subseteq [n + 1] \mid d(X, Y_n) = 1 \text{ iff } Y(n + 1) = 1\}$$

It is trivial to see that  $\mathcal{M}$  is indeed a stable set. Also observe that if  $\phi'$  had a  $\delta^*$ model, then it has a model  $X$  with  $X(n + 1) = 0$ . Then  $X_n$  is a model of  $F$ .  $\square$

As  $\delta^*$ models are automatically  $\delta^k$ models, for all  $k \geq 0$ , the proof of theorem 2.3 shows that  $\Phi^k(1, 1)$  is NP hard. Hence we have the following:

Corollary 2.4.  $\Phi^k(1, 1)$  is NP-complete.

Proof. We just need to show that  $\Phi^k(1, 1)$  is in NP. This is because an NDTM can guess an assignment  $X$  and check that it is a  $\delta^k(1, 1)$ -model. To check whether  $X$  is a  $\delta^k(1, 1)$ -model, it suffices to consider all possible  $n^k$  break sets, and check that a repair exists for each break applied in sequence from the break set. Since  $k$  is fixed, this can be done in polynomial time.  $\square$

### 3. Finding $\delta$ Models for Restricted Boolean Formulas

While the general problem of finding whether an input boolean formula has a  $\delta$ model is NP-complete, this question might have efficient answers for restricted classes of satisfiability. We consider the three polynomial-time instances (from Shaefer's dichotomy theorem [37]) of SAT which have polynomial-time satisfiability checkers: 2-SAT, Horn-SAT and Affine SAT. We observe an interesting phenomenon: these restricted classes have different complexity of testing fault tolerance. For example, 2-SAT and Affine SAT have polynomial time tests for the existence of  $\delta$ models (see Subsection 3.1 and 3.3) whereas the same problem is NP-complete for Horn SAT (Subsection 3.2). In Subsection 3.4, we give a summary of the complexity status of finding  $\delta$ models and  $\delta^*$ models for general and restricted boolean formulas.

#### 3.1: Finding $\delta$ models for 2-SAT

We now prove that finding  $\delta$ models for 2-SAT formulas is in polynomial time. 2-SAT formulas are in conjunctive normal form with 2 literals per clause. Recall that one can find models for 2-SAT formulas in polynomial time (see, e.g. [34]).

Let  $\phi$  be an instance of 2-SAT. Following the notation in [34], we define the graph  $G(\phi)$  as follows: the vertices of the graph are the literals of  $\phi$  and for each clause

$\alpha \rightarrow \beta$  (where  $\alpha, \beta$  are literals) we add two directed edges  $(\alpha, \beta)$  and  $(\neg\beta, \neg\alpha)$ . Thus the edges of  $G(\phi)$  capture the implications of  $\phi$ . The following lemma is easy to prove (see [34]).

**Lemma 3.1.** [34]  $\phi$  is unsatisfiable iff there is a variable  $x$  such that there is a path from  $x$  to  $\neg x$  and a path from  $\neg x$  to  $x$  in  $G(\phi)$ .

We will assume that each clause of  $G(\phi)$  is a disjunction of distinct literals, which means that  $G(\phi)$  has no isolated vertices.

If  $\phi$  has a  $\delta$ model, then  $G(\phi)$  has a further restriction.

**Lemma 3.2.** If  $\phi$  has a  $\delta$ model, then there is no path from  $u$  to  $\neg u$ , where  $u$  is a literal in  $\phi$ .

**Proof.** Clearly if such a path existed, the value of  $u$  in any model of  $\phi$  has to be set to false. Such a model can never be a  $\delta$ model, since a break to the value of  $u$  does not have a repair.  $\square$

Define a simple path in  $G(\phi)$  to be an ordered sequence  $\mathcal{P} = (u_1, u_2, \dots, u_m)$  where  $u_1, u_2, \dots, u_m$  are all distinct vertices. The length of  $\mathcal{P}$ , denoted by  $l(\mathcal{P})$ , is  $m - 1$ . By Lemma 3.2, we know that if  $\phi$  has a  $\delta$ model, then a simple path cannot include a variable and its negation. If  $X$  is a 0-1 assignment to the variables of  $\phi$  let  $X(u)$  denote the value of the literal  $u$  under  $X$ . One can easily show the following properties of  $G(\phi)$ .

**Lemma 3.3.** Let  $\phi$  have a  $\delta$ model  $X$ . Then

1. If  $\mathcal{P}$  is a simple path in  $G(\phi)$ , then  $l(\mathcal{P}) \leq 3$ .

2. Let  $\mathcal{P} = (u_1, u_2, u_3, u_4)$  be a simple path in  $G(\phi)$  of length 3. Then  $X(u_1) = X(u_2) = 0$  and  $X(u_3) = X(u_4) = 1$ .
3. Let  $\mathcal{P} = (u_1, u_2, u_3)$  be a simple path in  $G(\phi)$  of length 2. Then  $X(u_1) = 0$  and  $X(u_3) = 1$ .
4. Let  $\mathcal{P} = (u_1, u_2, \dots, u_{m+1} = u_1)$  be a simple cycle in  $G(\phi)$  of length  $m$ . Then  $m \leq 2$ .
5. If  $(u, v)$  and  $(u, w)$  are edges in  $G(\phi)$ , then  $X(u), X(v), X(w)$  cannot all be set to false.
6. If  $(v, u)$  and  $(w, u)$  are edges in  $G(\phi)$ , then  $X(u), X(v), X(w)$  cannot all be set to true.

Proof. We prove (i) – the others easily follow from similar arguments. Suppose  $l(\mathcal{P}) \geq 4$ . Then there is a simple path  $\mathcal{P}' = (u_1, u_2, u_3, u_4, u_5)$  where  $l(\mathcal{P}') = 4$  (take the initial 5 vertices of  $\mathcal{P}$ ). Let  $X$  be a  $\delta$ -model of  $\phi$ . Let  $F = \{u_i \mid X(u_i) = 0\}$  and  $T = \{u_1, u_2, \dots, u_5\} \setminus F$ . It is easy to see that  $F$  has to be the initial segment of  $\mathcal{P}'$ , and  $T$  has to be the remaining suffix. We claim that  $|F| \leq 2$ . Suppose not: then  $X(u_1) = X(u_2) = X(u_3) = 0$ . If we now break  $u_1$ , we will need 2 repairs, hence  $X$  cannot be a  $\delta$ model. Similarly  $|T| \leq 2$ . However  $|F| + |T| = 5$ , a contradiction.  $\square$

Let  $X$  be a partial assignment of the variables in  $\phi$ . We now show an algorithm that takes  $X$  and makes forced choices (but only with regard to vertices that take part in cycles) and checks to see whether  $X$  can be extended to a  $\delta$ model.

Extend( $\phi, X$ )

1. For each cycle  $(u_1, u_2), (u_2, u_1)$  in  $G(\phi)$ , such that exactly one of  $X(u_1)$  and  $X(u_2)$  is defined, set  $X(u_1) = X(u_2)$ . If there is a conflict, because of a vertex taking part in more than 1 cycle, then abort. Let  $X'$  be the new (partial) assignment.
2. For each edge  $(u_1, u_2)$  in  $G(\phi)$  such that both  $X(u_1)$  and  $X(u_2)$  are defined, check to see whether the implication  $u_1 \rightarrow u_2$  is satisfied by  $X$ . If not, abort.
3. For each triple of assigned vertices  $u, v, w$ , such that  $X(u) = X(v) = X(w) = 0$ , check if  $(u, v), (u, w)$  are edges in  $G(\phi)$ . If so, abort.
4. For each triple of assigned vertices  $u, v, w$ , if  $X(u) = X(v) = X(w) = 1$ , check if  $(v, u), (w, u)$  are edges in  $G(\phi)$ . If so, abort.
5. Return  $X'$ .

It is not difficult to see that if  $X$  can be extended to a  $\delta$ model for  $\phi$ , then  $\text{Extend}(X, \phi)$  returns  $X'$  which can also be extended to a  $\delta$ model.

Now we are ready to describe our algorithm  $\delta\text{Model}(\phi)$  to find  $\delta$ models for 2-SAT theories where the input instance is the 2-SAT formula  $\phi$ .

 $\delta\text{Model}(\phi)$ 

1. Construct  $G(\phi)$ . Set initial partial assignment  $X = \emptyset$ .
2. Check to see whether there is any vertex  $u$  such that there is a directed path from  $u$  to  $\neg u$ . If so, abort.

3. Check to see whether there is any simple path of length 5. If so, abort.
4. For every simple path of length 3 and every simple path of length 2, construct a partial assignment  $X$  as prescribed by Lemma 3.3. If there is a conflict in assigning a value to a vertex, abort.
5. Run  $\text{Extend}(\phi, X)$  which either aborts or returns a (possibly new) partial assignment  $X'$ .
6. For each isolated cycle  $(u, v), (v, u)$  (where both  $u, v$  have both in-degree and out-degree 1) such that both  $X'(u)$  and  $X'(v)$  are undefined, set both  $X'(u) = X'(v) = 1$ .
7. Let  $U$  be the set of literals left unassigned by  $X'$ . Construct a 2-SAT formula  $\beta$  as follows:
  - (a) Initially set  $\beta$  to the trivial (empty) formula.
  - (b) for each pair of unassigned literals  $u \in U$  and  $v \in U$  such that there is a vertex  $w$  in  $G(\phi)$  with  $X'(w) = 0$ , and  $(w, u)$  and  $(w, v)$  are edges in  $G(\phi)$ , set  $\beta = \beta \wedge (u \vee v)$ .
  - (c) for each pair of unassigned literals  $u \in U$  and  $v \in U$  such that there is a vertex  $w$  in  $G(\phi)$  with  $X'(w) = 1$ , and  $(u, w)$  and  $(v, w)$  are edges in  $G(\phi)$ , set  $\beta = \beta \wedge (\neg u \vee \neg v)$ .
  - (d) For each pair of literals  $u$  and  $v \in U$  such that there is a vertex  $w$  in  $G(\phi)$  with  $X'(w) = X'(u) = 0$ , and  $X(v)$  unassigned with  $(w, u)$  and  $(w, v)$  as edges in  $G(\phi)$ , set  $\beta = \beta \wedge (v)$ .



- (e) For each pair of literals  $u$  and  $v \in U$  such that there is a vertex  $w$  in  $G(\phi)$  with  $X'(w) = X'(u) = 1$ , and  $X(v)$  unassigned with  $(u, w)$  and  $(v, w)$  as edges in  $G(\phi)$ , set  $\beta = \beta \wedge (\neg v)$ .

If  $\beta$  is unsatisfiable, then abort else find a model for  $\beta$  and combine with  $X'$  to get an assignment  $M$ .

It is easy to see that if  $\delta\text{Model}(\phi)$  does not abort, the returned assignment  $M$  is a  $\delta$ model.

Each step in  $\delta\text{model}(\phi)$  is easily seen to be in polynomial time. Hence

Theorem 3.4. In polynomial time, one can determine if a 2-SAT theory has a  $\delta$ model and find one if it exists.

In fact, it is easy to see that each step can be done in NL.

Theorem 3.5.  $\Phi(1, 1) \cap 2\text{-SAT} \in \text{NL}$ .

Surprisingly, the situation completely alters when we consider  $\delta(1, b)$ -models for  $b > 1$ .

Theorem 3.6. Finding whether a 2-SAT theory has a  $\delta(1, b)$ -model is NP-complete for  $b > 1$ .

Proof. Clearly  $\Phi(1, b) \cap 2\text{-SAT} \in \text{NP}$ . We prove NP-completeness via a reduction from  $(b+1)$ -SAT. Let  $T = C_1 \wedge C_2 \dots \wedge C_m$  be an instance of  $(b+1)$ -SAT where each clause  $C_i$  is a disjunction of  $b+1$  literals:  $l_i(1) \vee l_i(2) \dots \vee l_i(b+1)$ . We

construct an instance  $T'$  of  $\Phi(1, b) \cap 2\text{-SAT}$  as follows:

$$T' = \bigwedge_{1 \leq i \leq m} F(i)$$

where  $F(i)$  is a 2-SAT theory defined for each clause  $C_i$  as follows:

$$F(i) = \bigwedge_{1 \leq j \leq (b+1)} (c_i \Rightarrow l_i(j)) \\ \bigwedge_{1 \leq j \leq (b+1)} (l_i(j) \Rightarrow a_i(1, j)) \\ \bigwedge_{1 \leq j \leq b-1} \bigwedge_{1 \leq k \leq (b+1)} (a_i(j, k) \Rightarrow a_i(j+1, k))$$

where we have introduced  $1 + b(b+1)$  new variables  $c_i$  and  $a_i(j, k)$  for  $1 \leq j \leq b$ ,  $1 \leq k \leq (b+1)$  to define the gadget  $F(i)$ . The gadget  $F(i)$  is best represented pictorially as follows:

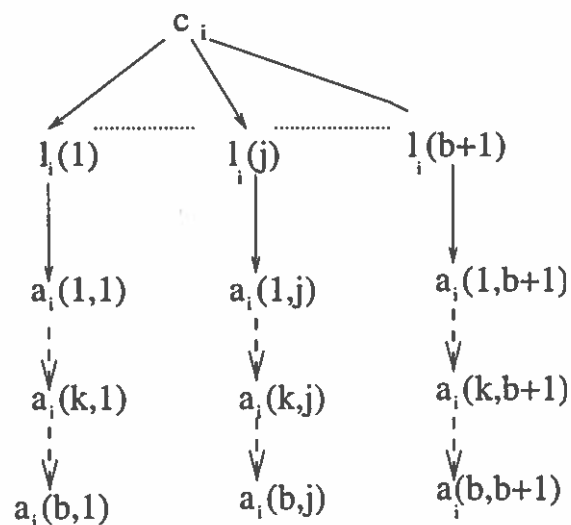


Figure 1: Gadget for 2-SAT

Let  $T$  have a model  $X$ . Extend that to a model of  $T'$  by setting  $c_i = 0$  for all  $1 \leq i \leq m$  and  $a_i(j, k) = 1$  for all  $1 \leq i \leq m, 1 \leq j \leq b, 1 \leq k \leq (b + 1)$ . We claim that this is a  $\delta(1, b)$ -model of  $T'$ . Flip any variable  $v$ . Now we do a case analysis of how many repairs are needed:

- $[v = c_i]$  Since  $l_i(1) \vee l_i(2) \dots \vee l_i(b + 1) = 1$  (since  $X$  is a model, there is at least one true literal in  $\{l_i(1), l_i(2), \dots, l_i(b + 1)\}$ ) so we need to flip at most  $b$  literals in  $\{l_i(1), \dots, l_i(b + 1)\}$ . Observe that no more repairs are necessary.
- $[v = a_i(j, k)]$  Need to flip  $a_i(t, k)$  where  $1 \leq t < j$  and we might need to flip the variable corresponding to  $l_i(k)$  if  $l_i(k)$  was set to true by  $X$ . This repair does not affect the truth value of other clauses of  $T'$ . Hence we flip at most  $j \leq b - 1 + 1 = b$  variables.
- $[v = l_i(j)]$  No repairs are necessary.

Now suppose  $T'$  has a  $\delta(1, b)$ -model. Note that in such a model  $c_i = 0$  for all  $i$  (otherwise we will need more than  $b$  repairs when we flip the value of  $a_i(1, 1)$ ). Now all literals  $\{l_i(1), l_i(2), \dots, l_i(b + 1)\}$  cannot be set to 0, since a break to  $c_i$  would again necessitate  $b + 1$  repairs. Hence at least one of the literals in  $\{l_i(1), l_i(2), \dots, l_i(b + 1)\}$  is set to 1. In other words, the clause  $C_i$  in  $T$  is satisfied. Since  $c_i = 0$  for all  $i$ ,  $T$  must have a model.  $\square$

We can also show that finding  $\delta^*$ -models for 2-SAT is in polynomial time.

Theorem 3.7.  $\Phi^*(1, 1) \cap 2\text{-SAT} \in P$ .

Proof. Let  $\phi$  be the input 2-SAT formula over  $n$  variables  $[n]$ . We construct the graph  $G(\phi)$  as described before.

Since a  $\delta^*$ model is by definition also a  $\delta$ model, we must have the same path restrictions set forth by Lemma 3.3 and Lemma 3.2. However, if  $\phi$  has a  $\delta^*$ model then we can show that any simple path in  $G(\phi)$  can have length at most 1. Suppose not: let  $(u, v, w)$  be a simple path of length 2. Let  $X$  be a  $\delta^*$ model of  $\phi$ . Because of Lemma 3.3, we know that  $X(u) = 0, X(w) = 1$  and this has to be the case for all  $\delta^*$ models, which means that a break to  $X(u)$  cannot be repaired to get another  $\delta^*$ model. Hence the length of a simple path in  $G(\phi)$  can have length at most 1. Note  $G(\phi)$  may have cycles  $(u, v), (v, u)$ , however in that situation  $\{u, v\}$  must form one connected component. We can assign either 0 or 1 to both  $u, v$  and remove them from consideration. So wlog, assume that  $G(\phi)$  has no cycles. In that case, the simple path length restriction means that  $G(\phi)$  is a bipartite graph.

Let  $G(\phi) = R \cup B$  where  $R, B$  are disjoint vertex sets and all edges in  $G(\phi)$  are between vertices in  $R$  and vertices in  $B$ . Let  $R$  be the vertices with in-degree 0 and  $B$  be the vertices with out-degree 0. Observe that a vertex cannot have positive in-degree and positive out-degree. Note that if  $(u, v)$  is an edge in  $G(\phi)$ , then the out-degree of  $\neg u$  is 0: otherwise, there would be a path of length 2 or a cycle, both of which we have excluded.

Hence if  $u \in R$  iff  $\neg u \in B$ . We also observe that there are no isolated points in  $G(\phi)$  since every clause is a disjunction of distinct literals. This a complete graph theoretic characterization of the structure of  $G(\phi)$  when  $\phi$  has a  $\delta^*$ model.

Now let  $X$  be an assignment that sets every literal in  $R$  false (0) and (that automatically sets) every literal in  $B$  true (recall our assumption from Section 1 that every variable appears in both positive and negative literals in a boolean formula).

Claim 3.8.  $X$  is a  $\delta^*$ model.

Proof. We exhibit a stable set  $\mathcal{C}$  of models of  $\phi$  and that contains  $X$ . Let  $Y_B$  ( $Y_R$ ) denote the restriction of any assignment  $Y$  onto the literals in  $B$  ( $R$ ), i.e.,  $Y_B = \{Y(u) \mid u \in B\}$ . Let

$$\mathcal{C} = \{Y \mid Y_B \text{ contains at most one literal set false}\}.$$

Note that if  $B$  contains at most one false literal under  $Y$ , then  $Y_R$  contains at most one true literal.

We now show that  $\mathcal{C}$  is a stable set. Suppose  $Y \in \mathcal{C}$  is such that  $Y_R$  ( $Y_B$ ) contains only false (true) literals. Let us break the value of a variable  $v$ : then there is a new false literal  $\neg u$  in  $B$  and a new true literal  $u$  in  $R$ . Since there is no directed edge  $(u, \neg u) \in G(\phi)$ , this break does not need a repair, the new assignment being already a member of  $\mathcal{C}$ . Now suppose that  $Y \in \mathcal{C}$  induces one true literal  $u$  in  $R$  (and hence induces the false literal  $\neg u$  in  $B$ ). Now let us break the value of a variable: if the variable corresponded to the literal  $u$ , then no repairs are needed, because the new assignment will induce only true (false) literals in  $B$  ( $R$ ). If not, then this break induces a second new true literal  $v$  in  $R$  and the new assignment is not in  $\mathcal{C}$  (since it has two literals  $u, v$  in  $R$  set true). We claim that flipping the variable corresponding to literal  $u$  is a repair: setting  $u$  to false still satisfies all implications  $(u, w)$  where  $w \in B$ : observe that  $\neg u$  is now set to true and there is no edge  $(u, \neg u)$  in  $G(\phi)$ . Clearly the repaired assignment satisfies the requirements for membership in  $\mathcal{C}$ .  $\square$

Since  $G(\phi)$  can be constructed in polynomial time and one can check whether it satisfies all the conditions needed for  $\phi$  to have a  $\delta^*$  model in polynomial time, we have a polynomial-time algorithm for  $\Phi^*(1, 1) \cap 2\text{-SAT}$ .  $\square$

### 3.2: Finding $\delta$ models for Horn-SAT

An instance of Horn-SAT is a boolean formula in CNF where each clause contains at most 1 positive literal. As in 2-SAT, there is a polynomial time algorithm to find a model of a Horn formula (see, e.g., [34]). However, *unlike* the situation in 2-SAT, finding  $\delta$ models for Horn formulas is NP complete.

**Theorem 3.9.**  $\Phi(1, 1) \cap \text{Horn-SAT}$  is NP-complete.

**Proof.**  $\Phi(1, 1) \cap \text{Horn-SAT}$  is clearly in NP. To prove that it is NP-hard, we reduce from 3-SAT. Let  $T = C_1 \wedge C_2 \cdots \wedge C_m$  be an instance of 3-SAT. We assume without loss of generality, that there are no pure literals in  $T$ .

For ease of description, we first apply an intermediate transformation to  $T$  by replacing any positive literal (say  $x$ ) in  $C_i$  by a new negative literal ( $\neg a_x$ ). But then we add clauses to  $T$  to signify that  $\neg a_x \Leftrightarrow x \equiv (((\neg a_x) \vee x) \wedge (a_x \vee x))$ . Thus we obtain

$$T' = \bigwedge_{1 \leq i \leq m} C'_i \bigwedge_x (\neg a_x \Leftrightarrow x)$$

where  $x$  refers to a variable (positive literal) in  $T$  and  $C'_i$  refers to the pure Horn clause (no positive literal) by replacing all the positive literals in  $C_i$  as described above. Note that since we assume that there are no pure literals in  $T$ , we add clauses  $((\neg a_x) \Leftrightarrow x)$  for all variables  $x$  that appear in  $T$ . Observe that  $T'$  is *almost* Horn, the bad clauses are only the clauses of the form  $(a_x \vee x)$ .

Clearly  $T'$  has a model iff  $T$  has a model. Now we produce an instance of  $\Phi(1, 1) \cap \text{Horn-SAT}$  from  $T'$ . We first introduce two global new variables  $A, B$ . For each clause  $C'_i = \neg v_i(1) \vee \neg v_i(2) \vee \neg v_i(3)$  of  $T'$  (note that  $v_i(1), v_i(2), v_i(3)$  are

pure variables, not literals) define the clause

$$\begin{aligned} C_i'' &= (\neg c_i \vee \neg v_i'(1) \vee \neg v_i'(2) \vee \neg v_i'(3)) \\ &\wedge (c_i \Rightarrow A) \wedge (A \Rightarrow B) \\ &\wedge (v_i(1) \Rightarrow v_i'(1)) \wedge (v_i(2) \Rightarrow v_i'(2)) \wedge (v_i(3) \Rightarrow v_i'(3)) \end{aligned}$$

where  $c_i, v_i'(1), v_i'(2), v_i'(3)$  are newly introduced variables for each clause  $C_i'$ . Note that  $C''$  is Horn. For the pair of clauses that represent  $(\neg a_x \Leftrightarrow x)$  we construct the clause-gadget

$$D(x) = (h_x \Rightarrow a_x) \wedge (a_x \Rightarrow t_x) \wedge (h_x \Rightarrow x) \wedge (x \Rightarrow t_x)$$

where  $h_x$  and  $t_x$  are new variables introduced for each variable  $x$  in the original theory  $T$ . Note that each clause in  $D(x)$  is Horn. Our instance of  $\Phi(1, 1) \cap \text{Horn-SAT}$  is

$$T'' = \bigwedge_{1 \leq i \leq m} C_i'' \bigwedge_x D(x)$$

Observe that each  $v_i'(j)$  where  $1 \leq j \leq 3$  is at the end of a chain of implications of length 2 ( $h_{v_i(j)} \Rightarrow v_i(j) \Rightarrow v_i'(j)$ ).

Suppose  $T'$  had a model  $X'$ . Extend that to a model  $X''$  of  $T''$  by setting  $c_i = 0$  for all  $i \in \{1, \dots, m\}$ ,  $A = 1, B = 1$  and  $v_i'(j) = 1$  for all  $1 \leq i \leq m$  and for all  $1 \leq j \leq 3$  and setting  $h_x = 0, t_x = 1$  for all variables  $x$ .

We now show that  $X''$  is a  $\delta$ model. Suppose some variable which appears in  $T'$  is flipped, then no repairs are needed. If  $A$  is flipped, we don't need repairs since

$c_i = 0$  for all  $i$  and  $B = 1$ . If  $B$  is flipped, we need to repair  $A$ . If  $h_x$  or  $t_x$  are flipped, we need one repair: either  $x$  or  $a_x$  (because of  $T'$ , both of them cannot be 1 or 0 as  $(x \vee a_x) \wedge (\neg x \vee \neg a_x)$ ). If  $c_i$  is flipped, we do not need to repair  $A$  or  $B$ . But we do need to repair either  $v'_i(1), v'_i(2)$  or  $v'_i(3)$  to make the clause  $\neg c_i \vee \neg v'_i(1) \vee \neg v'_i(2) \vee \neg v'_i(3)$  true. To be able to set one of these to 0 (all are 1) we require that one of  $v_i(1), v_i(2), v_i(3)$  to be set to 0, which is guaranteed by the hypothesis that  $X'$  is a model of  $T'$ . If  $v'_i(j)$  is flipped, at most one repair ( $v_i(j)$ ) is required. Hence  $X''$  is a  $\delta$ model.

Now suppose  $T''$  has a  $\delta$ model  $X''$ . We claim that the restriction of  $X''$  to the variables of  $T'$  is a model of  $T'$ . Observe that in  $X''$ , it must be the case that  $v'_i(j) = 1$  for all  $i, j$ : because each  $v'_i(j)$  is at the end of chain of implications of length 2 in a 2-SAT sub-formula of  $T''$  (see Lemma 3.3) and similarly  $h_x = 0, t_x = 1$ . Also,  $A = 1$ : suppose not, this implies that  $c_i = 0$  for all  $i$ . If  $c_i$  is now flipped, we need to repair at least one of the literals  $v'_i(j)$  appearing in  $C''_i$  and repair  $A$ . So 2 repairs will be needed. Hence  $A = 1$ . Since  $h_x = 0, t_x = 1$ , both  $a_x$  and  $x$  cannot be set to the same value. Hence  $a_x \Leftrightarrow x$  is true. We now need to show that restriction of  $X''$  to the variables of  $T'$  makes  $C'_i = \neg v_i(1) \vee \neg v_i(2) \vee \neg v_i(3)$  true. If  $c_i$  is flipped, one needs to repair with one of  $v'_i(1), v'_i(2), v'_i(3)$ . But this is possible only if one of  $v_i(1), v_i(2), v_i(3)$  is set to 0 which means that  $C'_i$  is indeed satisfied by  $X''$ .  $\square$

Using a similar construction, one can prove that  $\Phi(1, b) \cap \text{Horn-SAT}$  is NP-complete when  $b$  is a fixed integer.



### 3.3: Finding $\delta$ models for Affine-SAT

Another class of boolean formulas that have polynomial time satisfiability checkers is Affine-SAT: these are formulas which are a conjunction of clauses, where each clause is an exclusive-or (denoted by  $\oplus$ ) of distinct literals. One can find a satisfying assignment for a formula in affine form by a variant of Gaussian elimination. We now prove that finding  $\delta$ models for affine formulas is also in polynomial time.

**Theorem 3.10.**  $\Phi(1, 1) \cap \text{Affine-SAT} \in \text{P}$ .

**Proof.** Let  $\phi = C_1 \wedge C_2 \dots \wedge C_m$  be a boolean formula in affine form over the set of variables  $X = \{x_1, x_2, \dots, x_n\}$ .

For each variable  $x$ , define  $I(x) = \{i \mid 1 \leq i \leq m, x \text{ appears in } C_i\}$ . For  $i \in I(x)$ , let  $N_i(x) = \{y \in X \mid y \text{ appears in } C_i, y \neq x\}$  denote the set of variables that appear with  $x$  in clause  $C_i$ . For  $1 \leq i \leq m$  and  $Y \subseteq X$ , let  $Y \cap C_i$  denote the set of variables in  $Y$  that appear in clause  $C_i$ . With a slight abuse of notation, let  $I(Y) = \bigcup_{y \in Y} I(y)$  denote the set of clauses where any variable in  $Y$  appears.

**Lemma 3.11.**  $\phi$  has a  $\delta$ model iff  $\phi$  is satisfiable and for all  $x \in X$ , there exists  $y = y(x) \in X$ , such that  $y \in \bigcap_{i \in I(x)} N_i(x)$  and  $x \in \bigcap_{i \in I(y)} N_i(y)$ .

**Proof.** It is easy to see that  $\{x, y(x)\}$  are a break-repair pair. □

Since the conditions in Lemma 3.11 are easily checkable in polynomial time, we have a polynomial time algorithm for  $\Phi(1, 1) \cap \text{Affine-SAT}$ . □

We can in fact, prove the following stronger theorem:

**Theorem 3.12.**  $\Phi(r, s) \cap \text{Affine-SAT} \in \text{P}$ .

Proof. For each of the possible  $O(n^r)$  break sets, there are  $O(n^s)$  repair sets possible. The following lemma characterizes when a set  $S$  can be a repair set.

Lemma 3.13. Let  $R \in \binom{X}{(\leq r)}$ . Then the break  $\delta_R(X)$  is repaired by  $\delta_S$  for  $S \in \binom{X}{(\leq s)}$ ,  $S \cap R = \emptyset$  in a  $\delta(r, s)$ -model  $\mathcal{M}$  iff the following hold:

- (i) for each  $i \in I(R)$ , if  $|R \cap C_i|$  is odd, then  $|S \cap C_i|$  is odd.
- (ii) for each  $i \notin I(R)$ ,  $|S \cap C_i|$  is even.

Since  $r$  and  $s$  are fixed constants, the conditions in Lemma 3.13 can be checked in polynomial time.

Hence  $\Phi(r, s) \cap \text{Affine-SAT}$  is in polynomial time.  $\square$

Observe that Lemma 3.11 implies that any  $\delta$ model of an Affine-SAT formula is automatically a  $\delta^*$ model, since the necessary and sufficient conditions are about the formula and not the  $\delta$ model. Thus an Affine-SAT formula has a  $\delta$ model iff it has a  $\delta^*$ model, hence  $\Phi^*(1, 1) \cap \text{Affine-SAT}$  is also in polynomial time.

### 3.4: Complexity status: a summary

The principal results of Sections 2 and 3 are summarized in the Table 2.

	SAT	$\Phi(1, 1)$	$\Phi(1, 2)$	$\Phi^*(1, 1)$
general	NP-complete	NP-complete	NP-complete	NEXP, NP-hard
2-SAT	P	P	NP-complete	P
Horn-SAT	P	NP-complete	NP-complete	open
Affine-SAT	P	P	P	P

Table 2: Complexity of Finding  $\delta$ models

The complexity of  $\Phi(r, s)$  where  $r$  and  $s$  are part of the input as opposed to being fixed constants remains an interesting open question. It is not hard to see that  $\Phi(r, s)$  is in  $\Sigma_3^P$ : it is not known whether it is complete for that class. The status of this problem for restricted cases such as 2-SAT is similarly open. Another interesting question is whether we can improve on  $\Phi^*(1, 1) \in \text{NEXP}$  (e.g., to PSPACE or even NP). This improvement seems to rely on finding suitable small certificates for stable sets. We take a first step in this direction in Subsections 5.1 and 5.2.

Finally, a practical modification of  $\delta$ models involves weakening the condition to allow only a high percentage of the breaks to be repaired. We wonder how this would affect the complexity issues.

#### 4. Stable Sets: Definitions and Notations

If a boolean formula  $\phi$  has a  $\delta^*$ -model then it contains a stable set of models  $\mathcal{M}$  with the property that for all  $X \in \mathcal{M}$  and for all breaks  $i \in [n]$ , either  $\delta_i(X) \in \mathcal{M}$  or there is some  $j \neq i \in [n]$  such that  $\delta_{ij}(X) \in \mathcal{M}$ . Observe that this definition allows models in  $\mathcal{M}$  to be at Hamming distance one from each other.

Since we are treating assignments (and models) as sets, we decide to study the concept of stability as a property of families of sets, independent of any reference to a boolean formula. Let  $\mathcal{F} \subseteq 2^{[n]}$  be a family of sets. We say that  $\mathcal{F}$  is stable if for all  $X \in \mathcal{F}$  and for all  $1 \leq i \leq n$ , there exists a  $1 \leq j \leq n$ ,  $j \neq i$ , such that  $\delta_{ij}(X) \in \mathcal{F}$ . Note that in a weak stable family there may be 2 sets at distance 1 from each other: in general stable families, we disallow this. All the theorems that we prove can be proved (with minor modifications) for weak stable families, but the combinatorics is a little cleaner when we consider stable families and hence in the subsequent sections, these are the objects we consider. Our goal will be to elucidate the combinatorial structure

of stable families under suitable restrictions. In particular, we will be interested in the sizes of minimal and maximal structures and explicit constructions.

The first restriction is that each break to an element in  $\mathcal{F}$  has exactly one repair. We shall call these families sparse stable families. Formally,  $\mathcal{F} \subseteq 2^{[n]}$  is sparse stable if for all  $X \in \mathcal{F}$ , for all breaks  $i \in [n]$ , there exists a unique  $j \in [n], j \neq i$  such that  $\delta_{ij}(X) \in \mathcal{F}$ . We shall refer to this unique repair  $j$  as  $r_i^X$ ; when  $X$  is obvious from the context, we drop it from the notation and write  $r_i$  for  $r_i^X$ .

To prove lower bounds on the sizes of stable families, we need to introduce a notion of partial stability. While this is an interesting concept in its own right, in this thesis we primarily use it as a tool to make our inductive proofs go through.

First we need some definitions. Let  $\mathcal{F} \subseteq 2^{[n]}$  and let  $X \in \mathcal{F}$ . For each  $i \in [n]$ , define  $r^X(i) = \{j \mid \delta_{ij}(X) \in \mathcal{F}, j \neq i\}$ . When  $X$  is clear from the concept, we drop it from the notation and simply write  $r(i)$ . Define  $R(X) = \{i \in [n] \mid r^X(i) \neq \emptyset\}$  and  $R_u(X) = \{i \in [n] \mid |r^X(i)| = 1\}$ . A coordinate  $i \in R_u(X)$  has a unique repair  $j \in r^X(i)$ ; we refer to that repair as  $r_i^X$  (or, simply as  $r_i$  when  $X$  is obvious from the context).

A family  $\mathcal{F} \subseteq 2^{[n]}$  is  $k$ -stable if each  $X \in \mathcal{F}$  satisfies the following properties:

- (i) For each  $i \in [n]$ ,  $\delta_i(X) \notin \mathcal{F}$ .
- (ii)  $|R(X)| \geq k$ .

**Remark:** Observe that definition of  $R(X)$  implies that the repair bit  $j$  is also in  $R(X)$  where  $j \in r(i)$ . Condition (i) implies that no two sets in a  $k$ -stable family can be at distance 1 from each other. Condition (ii) implies that a  $k$ -stable family is  $l$ -stable for all  $l \leq k$ .

We similarly define  $k$ -sparse stable families: a family  $\mathcal{F} \subset 2^{[n]}$  is  $k$ -sparse stable if each  $X \in \mathcal{F}$  satisfies the following properties:

- (i) For all  $i, j \in R_u(X)$  such that  $i \neq j$ ,  $r_i \neq r_j$ , i.e., repairs are distinct.
- (ii) For each  $i \in [n]$ ,  $\delta_i(X) \notin \mathcal{F}$ .
- (iii)  $|R_u(X)| \geq k$ .

Remark:

- (1) The definition of  $k$ -sparse stable has very subtle differences from that of  $k$ -stable families. Note that unlike in the definition of  $k$ -stable families, the repair  $r_i$  is not necessarily in  $R_u(X)$ : if it is not in  $R_u(X)$ , then it will have multiple repairs (it already has one repair, namely  $i$ ).
- (2) If  $Y = \delta_{ir_i}(X) \in \mathcal{F}$ , where  $i \in R_u(X)$  then we claim that  $r_i \in R_u(Y)$ . This is because a break to coordinate  $r_i$  in  $Y$  is uniquely repaired by flipping coordinate  $i$ : if it had another repair  $j \neq i$ , then a break to  $i$  in  $X$  could also be repaired by flipping that  $j$ , contradicting the fact that  $r_i$  was the unique such repair in  $X$ .

A dense  $k$ -sparse stable family is one whose members satisfy all of the conditions of a  $k$ -sparse stable family except possibly (i), i.e., there might be a coordinate that is the unique repair of more than one coordinate for a member of the family.

## 5. Extremal Properties of Stable Sets

We now study the combinatorics of stable sets. We first consider sparse stable sets in Subsection 5.1: we prove lower and upper bounds for the size of a sparse stable set. Then we consider general stable sets in Subsection 5.2.

## 5.1: Sparse Stable Sets

In the following discussion, let  $\mathcal{F} \subset 2^{[n]}$  be sparse stable and  $X \in \mathcal{F}$ .  $X$  defines a relation on  $[n]$  as follows:  $i \stackrel{X}{\equiv} j$  if  $j = r_i$ . It is obvious that  $j = r_i$  iff  $i = r_j$ . This implies that  $\stackrel{X}{\equiv}$  is an equivalence relation on  $[n]$  with each equivalence class having size 2. Lemma 5.1 records the fact that  $\stackrel{X}{\equiv}$  is an equivalence relation and since each equivalence class is of size 2, we see that  $n$  has to be even.

**Lemma 5.1.** If  $Y = \delta_{ij}(X)$  and  $Z = \delta_{kl}(X)$  where  $X, Y, Z$  are distinct sets in the sparse stable family  $\mathcal{F} \subseteq 2^{[n]}$ , then  $n$  is even and  $\{i, j\} \cap \{k, l\} = \emptyset$ .

**Remark:** Note that it is important to include  $X$  in the definition of the relation  $\stackrel{X}{\equiv}$ , different  $X$ 's might give rise to different relations. In fact, presence of different equivalence relations within the same  $\mathcal{F}$  gives rise to structures with interesting properties, see for example Figure 2.

If we have three distinct sets  $X, Y, Z$  with  $Y = \delta_{ij}(X), Z = \delta_{ik}(X)$ , then the incidence vectors of  $X, Y$  and  $Z$  form an equilateral triangle with sides of length 2, the metric being the Hamming distance between the incidence vectors. We shall refer to this as a 2-triangle. As a consequence of Lemma 5.1 a sparse stable family is 2-triangle free, a fact which we exploit to prove upper bounds, see Theorem 5.10 and Theorem 5.11.

Given a sparse stable family  $\mathcal{F}$  we can define the (undirected) graph  $G = G(\mathcal{F})$  as follows: the vertices of  $G$  are the elements of  $\mathcal{F}$  and the edges are  $\{u, v\}$  where  $v = \delta_{ij}(u)$  for some  $i \neq j \in [n]$ .

We say that  $\mathcal{F}$  is connected if the graph  $G(\mathcal{F})$  is connected.

**Lemma 5.2.** Let  $\mathcal{F}$  be a sparse stable set. Then  $\mathcal{F}$  is minimal (i.e., does not contain a proper subset which is also sparse stable) iff  $\mathcal{F}$  is connected.

**Proof.** Let  $H$  be a connected component of  $G(\mathcal{F})$ . Let  $u \in H$ . By definition, for each  $i \in [n]$ ,  $u$  in  $H$  is connected to  $v$  where  $v = \delta_{ij}(X)$  for some unique  $j \in [n]$ . Hence  $H$  is stable. It is clearly sparse. Thus  $\mathcal{F}$  is minimal iff  $H = \mathcal{F}$ .  $\square$

We now derive a lower bound on the size of a minimal sparse stable set.

We first prove the following:

**Theorem 5.3.** If  $\mathcal{F} \subseteq 2^{[n]}$  is a non-empty  $k$ -sparse stable family, then  $|\mathcal{F}| \geq 2^{\lceil k/2 \rceil}$ .

**Proof.** By induction on  $k$ . For  $k = 0, 1, 2$ , the inductive hypothesis is easily seen to be true. Then assume that it is true for all  $2 < k < l$ . We prove it true for  $k = l$ .

Let  $\mathcal{F}$  be  $l$ -sparse stable. Choose an  $x_0$  such that there is some set  $S \in \mathcal{F}$  such that  $x_0 \in S$  and some  $T \in \mathcal{F}$  such that  $x_0 \notin T$ . There must be such an  $x_0$  since  $l \geq 1$  so we can choose any  $i \in R_u(X)$  for some  $X \in \mathcal{F}$  as our  $x_0$ . Let  $\mathcal{F}_{x_0} = \{X \in \mathcal{F} \mid x_0 \in X\}$  and  $\overline{\mathcal{F}_{x_0}} = \{X \in \mathcal{F} \mid x_0 \notin X\}$ .

Observe that both  $\mathcal{F}_{x_0}$  and  $\overline{\mathcal{F}_{x_0}}$  are  $(l-2)$ -sparse stable (since  $x_0$  can be a repair of at most one  $i \in R_u(X)$  for each set  $X$  in either  $\mathcal{F}_{x_0}$  or  $\overline{\mathcal{F}_{x_0}}$  and it itself can be in  $R_u(X)$  for some  $X$ ). By the induction hypothesis, this means that  $|\mathcal{F}_{x_0}| \geq 2^{\lceil \frac{l-2}{2} \rceil}$  and  $|\overline{\mathcal{F}_{x_0}}| \geq 2^{\lceil \frac{l-2}{2} \rceil}$ .

Since  $\mathcal{F}_{x_0} \cap \overline{\mathcal{F}_{x_0}} = \emptyset$ ,  $|\mathcal{F}| \geq 2 \times 2^{\lceil \frac{l-2}{2} \rceil} = 2^{1 + \lceil \frac{l-2}{2} \rceil} = 2^{\lceil l/2 \rceil}$ .  $\square$

**Remark:** Obviously when  $\mathcal{F} \subseteq 2^{[n]}$ , the largest  $k$  could be in the above theorem is  $n$ .

Thus we have a proof of the following theorem.

**Theorem 5.4.** If  $\mathcal{F} \subseteq 2^{[n]}$  is sparse stable, then  $|\mathcal{F}| \geq 2^{n/2}$ .

It is easy to see that the lower bound is tight: let  $\mathcal{B}$  denote the family of subsets of  $[n]$  (where  $n$  is even) whose incidence vectors satisfy the following boolean formula

$$\mathcal{B} = \bigwedge_{1 \leq i \leq n/2} (x_{2i-1} = x_{2i}) \quad (\text{III.10})$$

where  $x_j$  refers to the  $j$ th bit of the incidence vector. It is easy to see that  $\mathcal{B}$  is a sparse stable set (since it is connected) of size  $2^{n/2}$  and that it is minimal.

If  $\mathcal{F}$  is sparse stable, very often we will make the assumption that  $\emptyset \in \mathcal{F}$  – we can relabel 0's and 1's in a member of  $\mathcal{F}$  appropriately.

Let  $\mathcal{F}_k = \{A \in \mathcal{F} \mid |A| = k\}$ .

We investigate the structure of  $G(\mathcal{F})$  below.

**Lemma 5.5.** If  $\mathcal{F}$  is minimal and  $\emptyset \in \mathcal{F}$  then

1.  $\mathcal{F}_k = \emptyset$  for all odd  $k$ .
2. If  $\mathcal{F}_k$  is the highest non-empty level (i.e.,  $|\mathcal{F}_k| > 0$  and  $\mathcal{F}_i = \emptyset$  for all  $i > k$ ), then  $k \geq n/2$ .

**Proof.** (i) Observe that  $|X| + |\delta_{ij}(X)| \equiv 0 \pmod{2}$ . This means that  $X$  and  $\delta_{ij}(X)$  have the same parity. If  $u = \emptyset \in \mathcal{F}$ , any set  $v$  reachable by breaks and repairs has to have even parity. Since  $\mathcal{F}$  is minimal, it is connected (via Lemma 5.2) so every set is reachable from  $u$ . Thus  $\mathcal{F}$  cannot have any sets with odd parity.

(ii) Suppose  $k < n/2$ . Let  $\mathcal{F}_k$  be the highest non-empty level and let  $u \in \mathcal{F}_k$ . Let  $S = \{i \in [n] \mid u(i) = 1\} \in \binom{[n]}{k}$ . Let  $S' = \cup_{i \in S} r_i$ . Since  $\mathcal{F}$  is sparse stable,  $|S'| = k$ .



Since  $|S| = k$ ,  $|S' \cup S| \leq 2k < n$ . This means that for some index  $j \in [n] \setminus (S \cup S')$  which is repaired in  $u$  by some  $k$  also in  $[n] \setminus (S \cup S')$ . Observe that  $u(j) = u(k) = 0$ , so  $\delta_{jk}(u) \in \mathcal{F}_{k+2}$  which contradicts the hypothesis that  $\mathcal{F}_k$  was the largest non-empty level.  $\square$

The idea behind Lemma 5.5 (ii) is that if there are more 0's than 1's in the incidence vector, there must be two 0's which form a break-repair pair.

Let  $u \in \mathcal{F}_k$ , define the parents of  $u$  to be the set  $P(u) = \{v \in \mathcal{F}_{k-2} \mid u = \delta_{ij}(v) \text{ for some } i \text{ and } j\}$  and the children of  $u$  to be the set  $C(u) = \{v \in \mathcal{F}_{k+2} \mid v = \delta_{ij}(u) \text{ for some } i \text{ and } j\}$ . We prove the following estimates on the sizes of  $C(u)$  and  $P(u)$ .

**Lemma 5.6.** Let  $u \in \mathcal{F}_k$ , where  $2 \leq k < n/2$  is even. Then  $|P(u)| \leq k/2$  and  $|C(u)| \geq (n - 2k)/2$ .

**Proof.** Let  $S = \{i \in [n] \mid u(i) = 1\} \in \binom{[n]}{k}$ . Consider the equivalence relation defined on  $[n]$  via  $i \stackrel{u}{\equiv} j$  if  $i = r_j$  (which also implies  $j = r_i$ ). Clearly  $S$  can properly contain at most  $k/2$  equivalence classes. Each such equivalence class  $\{i, j\}$  corresponds to an element in  $v = \delta_{ij}(u) \in P(u)$ . Thus  $|P(u)| \leq k/2$ .

For  $x \in \mathcal{F}_k$ , there exists at least  $n - 2k$  indices with 0's (using similar arguments as in Lemma 5.5) which must properly contain equivalence classes under  $\stackrel{x}{\equiv}$ , each equivalence class  $\{i, j\}$  (of size 2) corresponding to an element  $y = \delta_{ij}(u) \in \mathcal{F}_{k+2}$ . There are at least  $(n - 2k)/2$  such equivalence classes. Hence  $|C(u)| \geq (n - 2k)/2$ .  $\square$

**Lemma 5.7.** Let  $\mathcal{F}$  be minimal sparse,  $\emptyset \in \mathcal{F}$  and  $2 \leq k \leq n/2$ . Then  $|\mathcal{F}_{k+2}| \geq |\mathcal{F}_k|(n - 2k)/(k + 2)$ .

Proof. Count the set  $\mathcal{C} = \{(x, y) \mid x \in \mathcal{F}_k, y \in \mathcal{F}_{k+2}, x \in P(y)\}$  in two different ways. Counting the first coordinate first and using the fact that  $|C(x)| \geq (n - 2k)/2$  (Lemma 5.6) we have  $|\mathcal{C}| \geq |\mathcal{F}_k|(n - 2k)/2$ . Similarly, we get  $|\mathcal{C}| \leq |\mathcal{F}_{k+2}|(k + 2)/2$  from Lemma 5.6. This gives us the desired result.  $\square$

If  $\mathcal{F}$  is minimal sparse with  $\emptyset \in \mathcal{F}$ , then we know that  $|\mathcal{F}_2| = n/2$ . Hence by Lemma 5.5, (assume for simplicity that  $4 \mid n$ )

$$\begin{aligned}
|\mathcal{F}| &\geq \sum_{\substack{i < n/2 \\ i=0, i \text{ even}}} |\mathcal{F}_i| \\
&\geq 1 + \frac{n}{2} + \frac{n(n-4)}{2 \cdot 2 \cdot 2} + \frac{n(n-4)(n-8)}{2 \cdot 2 \cdot 2 \cdot 3 \cdot 2} + \dots \\
&\geq \sum_{k=0}^{n/4} \prod_{i=0}^{k-1} \frac{n-4i}{2i+2} = \sum_{k=0}^{n/4} \frac{1}{k!} \prod_{i=0}^{k-1} \left(\frac{n}{2} - 2i\right) \\
&= \sum_{k=0}^{n/4} \frac{2^k}{k!} \prod_{i=0}^{k-1} \left(\frac{n}{4} - i\right) = \sum_{k=0}^{n/4} 2^k \binom{\frac{n}{4}}{k} \\
&= (1+2)^{n/4} = 3^{n/4}
\end{aligned}$$

Let  $N_i(X)$  ( $N_{\leq i}(X)$ ) denote the number of sets in  $\mathcal{F}$  at Hamming distance exactly equal to  $i$  (resp.  $\leq i$ ) from  $X$ . We have thus proved:

Theorem 5.8. If  $\mathcal{F} \subseteq 2^{[n]}$  is sparse stable and  $X \in \mathcal{F}$ , then  $|N_{\leq n/2}(X)| \geq 3^{n/4}$ .

Theorem 5.8 implies that there are  $3^{n/4}$  sets at distance  $\leq n/2$  from any set in  $\mathcal{F}$ . We know from Theorem 5.4 that  $|\mathcal{F}| \geq 2^{n/2} = 4^{n/4}$ . This leads us to conjecture that  $3^{n/4}$  is probably not a tight lower bound.

Conjecture 5.9.  $|N_{\leq n/2}(X)| \geq 2^{n/2-1}$  for all  $X \in \mathcal{F}$  where  $\mathcal{F} \subseteq 2^{[n]}$  is a sparse stable family.

Observe that this would give us an alternate evidence of the asymptotic lower bound of the size of minimal sparse stable sets. The reason why Theorem 5.8 does not give us the best possible result is that it counts the number of sets in each level  $\mathcal{F}_k$  which have parents in  $\mathcal{F}_{k-2}$  for  $2 \leq k \leq n/2$ . It might be the case that for some  $k$ , the level  $\mathcal{F}_k$  contains sets which have no parents in  $\mathcal{F}_{k-2}$ , i.e., they are only connected to vertices in  $\mathcal{F}_k$  and  $\mathcal{F}_{k+2}$  in  $G(\mathcal{F})$ . An estimate of the number of such sets would enable us to improve the lower bound of  $3^{n/4}$ .

We now turn to the problem of estimating the largest size of sparse stable families. Since we know that sparse stable families are 2-triangle free, we derive an upper bound on 2-triangle free families. This proof is taken from [29].

In the following discussion, let  $d(x, y)$  denote the Hamming distance between the  $n$ -bit vectors  $x$  and  $y$ .

**Theorem 5.10.** If  $\mathcal{F} \subseteq 2^{[n]}$  is 2-triangle free, then  $|\mathcal{F}| \leq \frac{2^{n+1}}{n}$ .

**Proof.** Let  $\mathcal{C} = \{(x, y) | x \in \mathcal{F}, y \subseteq [n], d(x, y) = 1\}$ . We count  $|\mathcal{C}|$  in two ways: counting via first coordinate we get  $|\mathcal{C}| = |\mathcal{F}|n$ . Let  $y \subseteq [n]$ . Observe that  $|N_1(y)| \leq 2$  where  $N_1(y) = \{x \in \mathcal{F} | d(x, y) = 1\}$ : if there were three sets in  $N_1(y)$  then they would form a 2-triangle in  $\mathcal{F}$ . Thus  $|\mathcal{F}|n \leq 2^n \times 2$ , since there are  $2^n$  choices of  $y$ . This gives us the desired bound.  $\square$

**Corollary 5.11.** If  $\mathcal{F} \subseteq 2^{[n]}$  is sparse stable, then  $|\mathcal{F}| \leq \frac{2^{n+1}}{n+2}$ .

**Proof.** A slight modification of the proof of Theorem 5.10 gives us our desired bound. Let  $\mathcal{E} = \{y | d(x, y) = 1 \text{ for some } x \in \mathcal{F}\}$  denote the set of vectors at distance 1 from any vector in  $\mathcal{F}$ . Observe that because we assume that every break requires

a repair (see definition in Section 4),  $\mathcal{E} \cap \mathcal{F} = \emptyset$ . That is, there cannot be two sets in  $\mathcal{F}$  at distance 1 from each other. Counting the family  $\mathcal{C}$  defined in Theorem 5.10 gives us:  $|\mathcal{C}| \leq |\mathcal{E}| \times 2$  and  $|\mathcal{C}| = |\mathcal{F}| \times n$ , which implies that  $|\mathcal{E}| \geq |\mathcal{F}| \times n/2$ . So we have

$$\begin{aligned} |\mathcal{F}| + |\mathcal{E}| &\leq 2^n \\ \text{i.e., } |\mathcal{F}| + |\mathcal{F}| \times \frac{n}{2} &\leq 2^n \\ \text{which implies that } |\mathcal{F}| &\leq \frac{2^{n+1}}{n+2} \end{aligned}$$

□

When  $\mathcal{F}$  is *minimal* sparse stable, we can improve Corollary 5.11 by a constant factor,

**Corollary 5.12.** If  $\mathcal{F} \subseteq 2^{[n]}$  is a minimal sparse stable family, then  $|\mathcal{F}| \leq \frac{2^n}{n}$ .

**Proof.** Without loss of generality we can assume that  $\emptyset \in \mathcal{F}$ . Since  $\mathcal{F}$  is connected (by Lemma 5.2),  $\mathcal{F} \subseteq E[n]$  where  $E[n]$  is the family of subsets of  $[n]$  with even parity.  $O[n]$  is the family of odd-parity subsets. This means that  $\mathcal{E}$  defined in Corollary 5.11 is a subset of  $O[n]$ . Hence  $|\mathcal{E}| \leq 2^{n-1}$ . But since  $|\mathcal{E}| \geq |\mathcal{F}| \times \frac{n}{2}$ , we get the desired bound. □

The best construction of large sparse stable sets we can give is of size  $2^n/n^2$  which we describe below. Constructions of large sparse sets (not necessarily stable) are considerably easier: an easy probabilistic construction shows that there are sparse families of size  $2^n/n^{1.5}$  [29].

We now describe a construction of sparse stable sets of size  $2^n/n^2$ . We slightly modify the notation of Equation III.10 to define the following family of boolean functions. Let  $S \subseteq [n/2]$ . Define

$$\mathcal{B}_S = \bigwedge_{i \in S} (x_{2i-1} \neq x_{2i}) \bigwedge_{i \notin S} (x_{2i-1} = x_{2i}).$$

So  $\mathcal{B}_\emptyset = \mathcal{B}$  of Equation III.10. It is clear that each  $\mathcal{B}_S$  defines a sparse stable set: we shall refer to an element of this sparse stable set as some  $x \in \mathcal{B}_S$  to mean that  $x$  satisfies  $\mathcal{B}_S$ . The following lemma allows us to build large (non-minimal) sparse stable sets.

**Lemma 5.13.** If  $S, T \subseteq [n/2]$  such that  $d(S, T) \geq 3$  then  $\mathcal{B}_S \vee \mathcal{B}_T$  defines a sparse stable set.

**Proof.** We prove that if  $x \in \mathcal{B}_S$  and  $y \in \mathcal{B}_T$ , then  $d(x, y) \geq 3$ , which implies that  $\mathcal{B}_S \vee \mathcal{B}_T$  is 2-triangle free. Let  $D = S \Delta T$  thus  $|D| \geq 3$ . Observe that any vector that satisfies  $x_{2i-1} = x_{2i}$  is at least distance 1 away from any vector that satisfies  $x_{2i-1} \neq x_{2i}$  where  $i \in D$ . Hence  $\mathcal{B}_S \vee \mathcal{B}_T$  is 2-triangle free. Obviously  $\mathcal{B}_S \vee \mathcal{B}_T$  is stable.  $\square$

Thus if  $\mathcal{F}$  is a family of subsets of  $[n/2]$  such that  $S, T \in \mathcal{F}$ ,  $S \neq T \Rightarrow d(S, T) \geq 3$ , then  $\bigvee_{S \in \mathcal{F}} \mathcal{B}_S$  will define a sparse stable set. We quote the following classic result.

**Theorem 5.14.** [Gilbert-Varshamov Inequality [43, 35]] There exists a family  $\mathcal{F} \subseteq 2^{[n]}$  satisfying the condition  $S, T \in \mathcal{F}$ ,  $S \neq T \Rightarrow d(S, T) \geq 3$  such that  $|\mathcal{F}| \geq \frac{2^n}{V_2(n)}$  where  $V_2(n) = |\binom{[n]}{\leq 2}| = \binom{n}{0} + \binom{n}{1} + \binom{n}{2}$ .

Using a family of size  $\Omega(\frac{2^{n/2}}{n^2})$  guaranteed by Theorem 5.14 as our “index” set, we get that many disjoint balls each of size  $2^{n/2}$  with a 2-triangle free union.

Corollary 5.15. There exist (non-minimal) sparse stable sets of size  $\Omega(2^n/n^2)$ .

We suspect that minimal sparse stable sets cannot achieve this bound. A bound on the largest known one is as follows:

Theorem 5.16. There exists a minimal sparse subset of  $2^{[n]}$  of size  $80^{n/10} \approx 1.54^n$ .

The proof follows from the existence of a minimal sparse stable set of 80 subsets for  $n = 10$  (see Section 6 for an enumeration of this set). We can use this to construct a minimal sparse set of size  $80^{n/10}$  by taking direct products. We do not have succinct description of this set: for example, a short boolean formula whose models correspond to the members of this set. The smallest example where break-repair pairings change within the same sparse family is a minimal sparse stable set of size 10 consisting of subsets of  $[6]$ . We also include this example as Figure 2 in Section 6. In that section, we also include a diagram (Figure 3) of a sparse sub-family of  $2^{[8]}$  of size 32 which shares many of the structural features of the example of size 80 for  $n = 10$ .

## 5.2: Stable Sets

In this section, we study the extremal properties of general stable sets, i.e., with no restrictions on the number of repair indices for a break. The first theorem concerns the minimum size of a stable set. To complete the proof, once again we need to turn to the partially stable (i.e.,  $k$ -stable) families.

Theorem 5.17. If  $\mathcal{F} \subseteq 2^{[n]}$  is a non-empty  $k$ -stable family, then  $|\mathcal{F}| \geq k$ .

Proof. By induction on  $k$ . For the base case  $k = 0$ , clearly  $|\mathcal{F}| > 0$  since  $\mathcal{F}$  is non empty. Assume that the hypothesis is true for all  $1 \leq k < l$ . Let  $\mathcal{F}$  be  $l$ -stable. Let  $x_0 \in [n]$  be such that there is some set  $S \in \mathcal{F}$  such that  $x_0 \in S$  and some  $T \in \mathcal{F}$  such that  $x_0 \notin T$ . Clearly such an  $x_0$  has to exist: consider any element  $x_0 \in r(i)$  for  $i \in R(X)$  for any  $X \in \mathcal{F}$ .

Let  $\mathcal{F}_{x_0} = \{X \in \mathcal{F} \mid x_0 \in X\}$ . For each  $X \in \mathcal{F}_{x_0}$ , let  $i(X) = \{i \in R(X) \mid x_0 \in r(i)\}$  (note that  $r(i)$  is still defined with respect to the original family  $\mathcal{F}$ ). Let  $i_0 = \max\{|i(X)| \mid X \in \mathcal{F}_{x_0}\}$ .

It is easy to see that  $\mathcal{F}_{x_0}$  is  $\max\{0, l - i_0 - 1\}$  repairable ( $x_0$  could itself be in  $R(X)$  for some  $X \in \mathcal{F}_{x_0}$ ). If  $l \leq i_0 + 1$ , then consider the  $X \in \mathcal{F}_{x_0}$  such that  $i(X) = i_0$ . Then each  $\delta_{i, x_0}(X) \in \mathcal{F}$  for  $i \in R(X)$  such that  $x_0 \in r(i)$ . Since there are at least  $l - 1$  choices for  $i$ ,  $|\mathcal{F}| \geq l - 1 + 1 = l$  (we also include the string  $X$  in this count). If  $i_0 < l - 1$ , then by the induction hypothesis,  $|\mathcal{F}_{x_0}| \geq l - i_0 - 1$ . Again consider the  $X \in \mathcal{F}_{x_0}$  such that  $i(X) = i_0$ . For each of the  $i_0$  breaks  $\delta_{i, x_0}(X)$  we get a distinct element in  $\mathcal{F} \setminus \mathcal{F}_{x_0}$ . Hence this gives us at least  $l - i_0 - 1 + i_0 + 1 = l$  members (we include  $X$  in the count) in  $\mathcal{F}$ .  $\square$

If  $\mathcal{F} \subseteq 2^{[n]}$ , the largest that  $k$  could be in the above theorem is  $n$ .

Corollary 5.18. If  $\mathcal{F}$  is a non-empty stable family, then  $|\mathcal{F}| \geq n$ .

This lower bound is easily seen to be tight: the boolean formula  $E_1(x_1, x_2, \dots, x_n)$  defines a stable set of size  $n$ , where  $E_1(x_1, x_2, \dots, x_n)$  is true iff exactly one variable in  $\{x_1, x_2, \dots, x_n\}$  is true.

As in the previous section, we define the undirected graph  $G(\mathcal{F})$  with vertices as elements of  $\mathcal{F}$  and edges  $\{u, \delta_{ij}(u)\}$  for  $u \in \mathcal{F}$  and  $i, j \in [n]$ . We can also easily see

that a weaker version of the statement in Lemma 5.2 holds for general stable sets.

Lemma 5.19. Let  $\mathcal{F}$  be a minimal stable set. Then  $\mathcal{F}$  is connected.

Observe that the converse is not necessarily true. The family of sets with an even number of elements is clearly stable and connected but not minimal.

Because minimal stable sets are connected and  $\emptyset \in \mathcal{F}$  (wlog), all sets in  $\mathcal{F}$  have even parity (analogous to Lemma 5.5). It is thus clear that minimal stable sets can have size at most  $2^{n-1}$ . We can improve this result to a constant fraction of  $2^{n-1}$ . The key to the proof again are bounds on the sizes of partially stable sets.

We need to derive some easy lemmas.

The following lemma is trivial to prove:

Lemma 5.20. Let  $G = (V, E)$  be a graph such that each  $v \in V$  has degrees 1 or 2. Then  $|V| \leq 2|E|$ .

Proof. Count the family  $\mathcal{C} = \{\{v, e\} \mid v \in V \text{ is incident on } e \in E\}$  in two ways. Clearly  $|\mathcal{C}| \geq |V|$  since each vertex has degree at least 1 and  $|\mathcal{C}| = 2|E|$  since each edge is incident on exactly two vertices in  $V$ . This gives us  $|V|/2 \leq |E|$ .  $\square$

We first investigate the properties of dense  $k$ -sparse stable families and use this to prove upper bounds on the sizes of minimal stable families.

Let  $\mathcal{F}$  be a dense  $k$ -sparse stable family. Let  $X, Y \in \mathcal{F}$ : let  $i \in R(X)$  and  $r_i = j$ , such that  $\delta_{ij}(X) = Y$ . Since  $j = r_i$ , we write  $Y = \delta_{(i,j)}^*(X)$ . Observe that the order  $(i, j)$  is important: a break to coordinate  $j$  can have multiple repairs in  $X$ . For  $Z \subseteq [n]$ , we say that the pair  $\{X, Y\}$  *excludes*  $Z$  if  $Y = \delta_{(i,j)}^*(X)$ ,  $Z = \delta_{ik}(X)$  for some distinct  $k \in [n], k \neq i$  or  $j$ . Note that we can write  $\{X, Y\}$  as an unordered



pair because if  $Y = \delta_{(i,j)}^*(X)$  then  $X = \delta_{(j,i)}^*(Y)$  and  $Z$  is still excluded. Thus if  $Y = \delta_{(i,j)}^*(X)$ , then  $\{X, Y\}$  excludes  $n - 2$  elements  $Z = \delta_{(i,k)}(X)$  where  $k \neq j, i$ . We say that  $X \in \mathcal{F}$  excludes  $Z$  if there is a  $Y \in \mathcal{F}$  such that  $\{X, Y\}$  excludes  $Z$ .

**Lemma 5.21.** Let  $\mathcal{F} \subseteq 2^{[n]}$  be dense  $k$ -sparse stable and let  $X \in \mathcal{F}, Z \subseteq [n]$ .

Then

$$|\{Y \in \mathcal{F} \mid \{X, Y\} \text{ excludes } Z\}| \leq 2.$$

**Proof.** Wlog assume  $Z = \emptyset$ . Then  $X = \{i, j\}$  for some  $i, j \in [n]$ . Any  $Y \neq X$  such that  $\{X, Y\}$  excludes  $Z$  has to have  $|Y| = 2$  and  $Y \cap X \neq \emptyset$ . We claim that  $Y \cap X$  has to be distinct for each such  $Y$ : otherwise suppose  $Y_1 = \{i, k\}, Y_2 = \{i, l\}$  such that both  $\{X, Y_1\}$  and  $\{X, Y_2\}$  exclude  $Z$ . This means that  $\delta_{(j,k)}^*(X) = Y_1$  and  $\delta_{(j,l)}^*(X) = Y_2$ , a contradiction. Hence there can be at most 2 such sets  $Y$  with distinct intersection with  $X$ .  $\square$

Construct the graph  $G_{\mathcal{F}, Z} = (V, E)$  where  $Z \subseteq [n]$ , and  $V = \{X \in \mathcal{F} \mid X \text{ excludes } Z\}$  and

$$E = \{\{X, Y\} \mid \{X, Y\} \text{ excludes } Z, \text{ where } X, Y \in \mathcal{F}\}.$$

Then Lemma 5.21 implies that the maximum degree of  $G(\mathcal{F})$  is at most 2 and since we only include those  $X$  which exclude  $Z$ , the minimum degree is at least 1.

**Lemma 5.22.** Let  $G = G_{\mathcal{F}, Z} = (V, E)$ . Then  $|V| \leq 2n$ .

**Proof.** Assume wlog  $Z = \emptyset$ . Then the vertices of  $G$  consist of unordered tuples of 2-element subsets of  $[n]$ . Consider the family  $\mathcal{C} = \{X \cap Y \mid \{X, Y\} \in E\}$ . It is clear that  $\mathcal{C}$  consists of one element subsets of  $[n]$ . We claim that the map

$E \rightarrow \mathcal{C}$  via  $\{X, Y\} \rightarrow X \cap Y$  is injective. Suppose not: then there are two pairs  $\{X, Y\}, \{R, S\} \in E$  such that  $X \cap Y = R \cap S$ . Let  $X = \{a, b\}, Y = \{a, c\}$  so that  $Y = \delta_{(b,c)}^*(X)$  and let  $R = \{a, d\}, S = \{a, e\}$  so that  $\delta_{(d,e)}^*(R) = S$ . Observe that  $b \neq d$ , otherwise  $S = \delta_{be}(X)$  and then  $Y = \delta_{(b,c)}^*(X)$  is violated. Similarly  $d \neq c$ . Since  $\delta_{(d,c)}(R) = Y$ , we cannot have  $\delta_{(d,e)}^*(R) = S$ , a contradiction.

Since  $|\mathcal{C}| \leq n, |E| \leq n$ . Now lemma 5.20 implies that  $|V| \leq 2|E|$ , so  $|V| \leq 2n$ .

□

Remark: We strongly suspect that Lemma 5.22 can be improved. In fact, if the upper bound of  $2n$  can be replaced by  $n + O(1)$  (explicit constructions seem to give at most that many vertices in the graph), this improvement will lead us to a better upper bound for the largest minimal sparse stable families (see Conjecture 5.29).

We need the following lemma to count the number of sets  $Z$  that a set  $X \in \mathcal{F}$  can exclude. For  $X \in \mathcal{F}$  denote

$$Z_X = \{ Z \mid X \in \mathcal{F} \text{ excludes } Z \}$$

and

$$\mathcal{Z} = \bigcup_{X \in \mathcal{F}} Z_X.$$

Lemma 5.23. Let  $\mathcal{F} \subset 2^{[n]}$  be a dense  $k$ -sparse family. Let  $X \in \mathcal{F}$  and  $Z \subseteq [n]$ .

Then

$$|Z_X| \geq \begin{cases} k(n - \frac{k+3}{2}) & \text{if } k \leq n - 3 \\ n(n - 3)/2 & \text{otherwise} \end{cases}$$

Proof. Wlog assume that  $X = \emptyset$ . Recall that  $R(X) \subseteq [n]$  is the set of

coordinates with a unique repair in  $\mathcal{F}$  (so  $|R(X)| = r \geq k$ ) and for each  $i \in R(X)$ ,  $r_i$  is the unique repair. Let  $Z_i = \{Z \subseteq [n] \mid Z = \delta_{ij}(X), j \neq r_i\}$ , i.e.,  $Z_i$  denotes the sets that  $X$  excludes because of its unique repair to the break at coordinate  $i$ . Also each  $Z_i \subseteq \binom{[n]}{2}$  and  $|Z_i| = n - 2$ . Obviously  $Z_X = \bigcup_{i \in R(X)} Z_i$ .

Let  $i, j, k$  be distinct elements in  $R(X)$  and let  $Z \in Z_i \cap Z_j \cap Z_k$ . Then  $Z = \delta_{ix_i}(X) = \delta_{jx_j}(X) = \delta_{kx_k}(X)$  (where  $x_l \neq r_l$  for  $l \in \{i, j, k\}$ ) or in other words  $Z = \{i, x_i\} = \{j, x_j\} = \{k, x_k\}$ . Since  $i, j, k$  are distinct and  $Z$  is a 2-element set, such a  $Z$  cannot exist. Hence  $Z_i \cap Z_j \cap Z_k = \emptyset$ . Similarly we conclude that  $|Z_i \cap Z_j| \leq 1$  for all  $i \neq j$ .

Now using the principle of exclusion-inclusion,

$$\begin{aligned} |Z_X| &= \sum_{i \in R(X)} |Z_i| - \sum_{i < j} |Z_i \cap Z_j| + \\ &\quad \cdots + (-1)^k \sum_{i_1 < i_2 < \cdots < i_k} \left| \bigcap_{s=1}^k Z_{i_s} \right| + \cdots + (-1)^{|R(X)|} \left| \bigcap_{i \in R(X)} Z_i \right| \\ &\geq r(n-2) - \binom{r}{2} \quad \text{since all higher terms are 0} \\ &= r\left(n - \frac{r+3}{2}\right) \end{aligned}$$

The function  $f(r) = r\left(n - \frac{r+3}{2}\right)$  can attain its minima only at its extreme points:  $r = k$  or  $n$ . A comparison gives us the required minimum values.

□

**Lemma 5.24.** Let  $\mathcal{F} \subseteq 2^{[n]}$  be a dense  $k$ -sparse family and let  $Z \subseteq [n]$ .

$$\frac{|Z|}{|\mathcal{F}|} \geq \begin{cases} \frac{k}{2n} \left(n - \frac{k+3}{2}\right) & \text{if } k \leq n - 3 \\ \frac{n-3}{4} & \text{otherwise} \end{cases}$$

Proof. Count the family

$$\mathcal{H} = \{\{X, Z\} \mid X \in \mathcal{F}, Z \in \mathcal{Z}_X\}$$

in two different ways. Lemma 5.23 implies that

$$|\mathcal{H}| \geq |\mathcal{F}| \times k \left( n - \frac{k+3}{2} \right) \text{ if } k \leq n-2.$$

Otherwise if  $k > n-2$ ,

$$|\mathcal{H}| \geq |\mathcal{F}| \times n(n-3)/2.$$

Since there are  $|\mathcal{Z}|$  choices of  $Z$  and each  $Z$  is excluded by at most  $2n$  elements in  $\mathcal{F}$  (via Lemma 5.22), we also have

$$|\mathcal{H}| \leq |\mathcal{Z}| \times 2n$$

which proves the result. □

If  $\mathcal{F}$  is dense  $k$ -sparse minimal, without loss of generality we may assume that both  $\mathcal{F}$  and  $\mathcal{Z}$  consist of sets of even parity and clearly  $\mathcal{F} \cap \mathcal{Z} = \emptyset$ . Then, for  $k \leq n-3$ , we have

$$\begin{aligned} |\mathcal{F}| + |\mathcal{Z}| &\leq 2^{n-1}, \text{ i.e.,} \\ |\mathcal{F}| + |\mathcal{F}| \frac{k(n - \frac{k+3}{2})}{2n} &\leq 2^{n-1}. \end{aligned}$$

If  $k > n - 3$ , then

$$|\mathcal{F}| + |\mathcal{F}|(n - 3)/4 \leq 2^{n-1}.$$

We record these facts in the following theorem:

Theorem 5.25. If  $\mathcal{F}$  is a minimal dense  $k$ -sparse stable family, then

$$|\mathcal{F}| \leq \begin{cases} \frac{2n}{2n+k(n-\frac{k+3}{2})} 2^{n-1} & \text{if } k \leq n - 3 \\ \frac{2^{n+1}}{n+1} & \text{otherwise} \end{cases}$$

Theorem 5.25 actually gives us essentially the same result for the largest size of a minimal sparse stable set, a bound of  $\frac{2^{n+1}}{n+1}$  (use  $k = n$  in the above expression), a result slightly worse than obtained in Theorem 5.12. However notice a subtle point: a dense  $k$ -sparse stable family need not be  $k$ -sparse stable because there might be a coordinate which uniquely repairs multiple coordinates for a set in the family. However, a dense  $n$ -sparse sub-family of  $2^{[n]}$  is automatically an  $n$ -sparse family, since it is sparse stable.

Similarly, a slight modification of the argument in Theorem 5.25 produces a slightly worse upper bound to the size of a sparse stable set (not necessarily minimal) of  $\frac{2^{n+2}}{n+1}$  (compared to Corollary 5.11).

Surprisingly, Theorem 5.25 also gives us an upper bound on the size of minimal stable families, even though its statement is about  $k$ -sparse families. To establish the connection, we need to make the following observation:

Lemma 5.26. If  $\mathcal{C}$  is a minimal stable set, then it is a dense 1-sparse stable family.

Proof. Suppose there is some  $X \in \mathcal{F}$  does not have a coordinate with a unique repair in  $\mathcal{F}$ . Since  $\mathcal{F}$  is a stable set, this means that every coordinate break in  $X$  has multiple repairs. We claim that  $\mathcal{F} \setminus \{X\}$  is a stable set. Suppose not. Then there is some  $Y \in \mathcal{F} \setminus \{X\}$  and an  $i \in [n]$  such that  $\delta_{(i,j)}^*(Y) = \{X\}$  for some  $j \neq i$ . However this means that  $\delta_{(j,i)}^*(X) = Y$  so  $X$  does a coordinate with a unique repair, a contradiction.  $\square$

Corollary 5.27. Let  $\mathcal{F}$  is a stable minimal family of subsets of  $[n]$ , then  $|\mathcal{F}| \leq (\frac{2}{3} + o(1)) 2^{n-1}$ .

Proof. Lemma 5.26 and Theorem 5.25 imply that

$$|\mathcal{F}| \leq \frac{2}{3 - \frac{2}{n}} 2^{n-1}$$

from which the result follows.  $\square$

As in the situation for sparse stable sets, we do not know of constructions of large minimal stable sets which achieve the above bound. However we have the following lower bound.

Theorem 5.28. There exists a minimal stable subset of  $2^{[n]}$  of size  $2^{2n/3}$ .

The proof of the theorem relies on the existence of a minimal stable set of size 16 for  $n = 6$  (displayed in Section 6), which was found by exhaustive search. A direct product of these yields a minimal set of size  $16^{n/6}$ , thus proving the theorem. Similar searches found the minimal set for  $n = 5$  is 8 and for  $n = 4$  is 4. This suggests the following conjecture.

Conjecture 5.29. The largest minimal stable subset of  $2^{[n]}$  is of size  $2^{n-2}$ .

We now turn to the following algorithmic question : given a stable set as input, is it minimal? Observe that a brute force algorithm that checks each subset of the stable set will run in exponential time in the size of the input. The size of the input could itself be exponential with respect to  $n$ , the length of the strings. Our goal is to find an algorithm that runs in polynomial time in the size of the input (not necessarily polynomial time in  $n$ ).

Theorem 5.30. In polynomial time, one can test if a stable set is minimal.

Proof. Let  $\mathcal{F}$  denote the input stable set. For each vertex  $u \in \mathcal{F}$  the algorithm runs the procedure  $\text{expand}(\{u\})$ . The procedure  $\text{expand}(X)$  executes the following steps in sequence:

1. If  $X = \mathcal{F}$  return true.
2. If there exists an  $u \in \mathcal{F} \setminus X$  and an  $i \in [n]$  such that for all  $j \in [n]$  ( $j \neq i$ ),  $\delta_{ij}(u) \in \mathcal{F} \Rightarrow \delta_{ij}(u) \in X$ , then set  $X = X \cup \{u\}$ .
3. If no such  $u$  exists and  $X \subset \mathcal{F}$  then return false ( $\mathcal{F}$  is not minimal). Else go to step 1.

If  $\text{expand}(\{u\})$  returns true for each  $u \in \mathcal{F}$  then  $\mathcal{F}$  is minimal. If  $\text{expand}(\{u\})$  is false for any  $u \in \mathcal{F}$  then  $\mathcal{F}$  is not minimal.

Now we prove correctness. It is obvious that if  $\text{expand}(\{u\})$  stops (i.e., returns false) with  $X \subsetneq \mathcal{F}$ , then  $\mathcal{F} \setminus X$  is a stable set and  $\mathcal{F}$  is not minimal.

We now prove that if  $\mathcal{F}$  was not minimal, then there is a  $u \in \mathcal{F}$  that will make  $\text{expand}(\{u\})$  return false. If  $\mathcal{F}$  was not minimal, then there is some stable set  $S \subsetneq \mathcal{F}$ .

Let  $u \in \mathcal{F} \setminus S$ . We claim that at any stage of  $\text{expand}(\{u\})$ , the set  $X$  that it has built so far, will be disjoint from  $S$ , i.e.,  $X \cap S = \emptyset$ . Suppose not, then let  $s \in S$  be the first element in  $S$  which is being added to a set  $X'$  built so far by  $\text{expand}(\{u\})$  in step 2. This means  $X' \cap S = \emptyset$ .  $s$  is added because all repairs to a break to particular coordinate lie in  $X'$ . However, this implies that all repairs to that particular break to  $s$  actually lie outside  $S$ , so  $S$  cannot be closed, a contradiction. Since  $\text{expand}(\{u\})$  can never include an element in  $S$ , it has to stop with a subset  $X \subseteq \mathcal{F}$  and return false.

Clearly this algorithm runs in polynomial time (polynomial in  $|\mathcal{F}|$  and  $n$ ).  $\square$

## 6. Examples of Stable Sets

We conclude with some examples of minimal sparse stable families and large minimal stable families. Figure 2 shows a sparse stable set of size 10 for  $n = 6$ . The break repair pairs are also shown for each string in the set.

Observe that the break-repair pairs in the above example is different for each set. In fact, for each  $1 \leq i \neq j \leq 6$ , there is a set with  $\{i, j\}$  as a break-repair pair. Such stable families seem to be rather rare: experimentally, we found that, in most cases, sparse stable sets are direct products or subsets of direct products.

We now show an instructive example of a minimal sparse stable set of size 32 for  $n = 8$ . This sparse stable subset is a proper subset of  $E[4] \times E[4]$  where  $E[4]$  is the set of 4-bit strings of even parity. The projection in the first 4 coordinates is  $E[4]$  and for each vector in the projection, the corresponding set of vectors in the last 4 coordinates is a sparse stable set of size 4 as shown in Figure 3.

We now include the sparse stable set of size 80 for  $n = 10$  used to prove The-



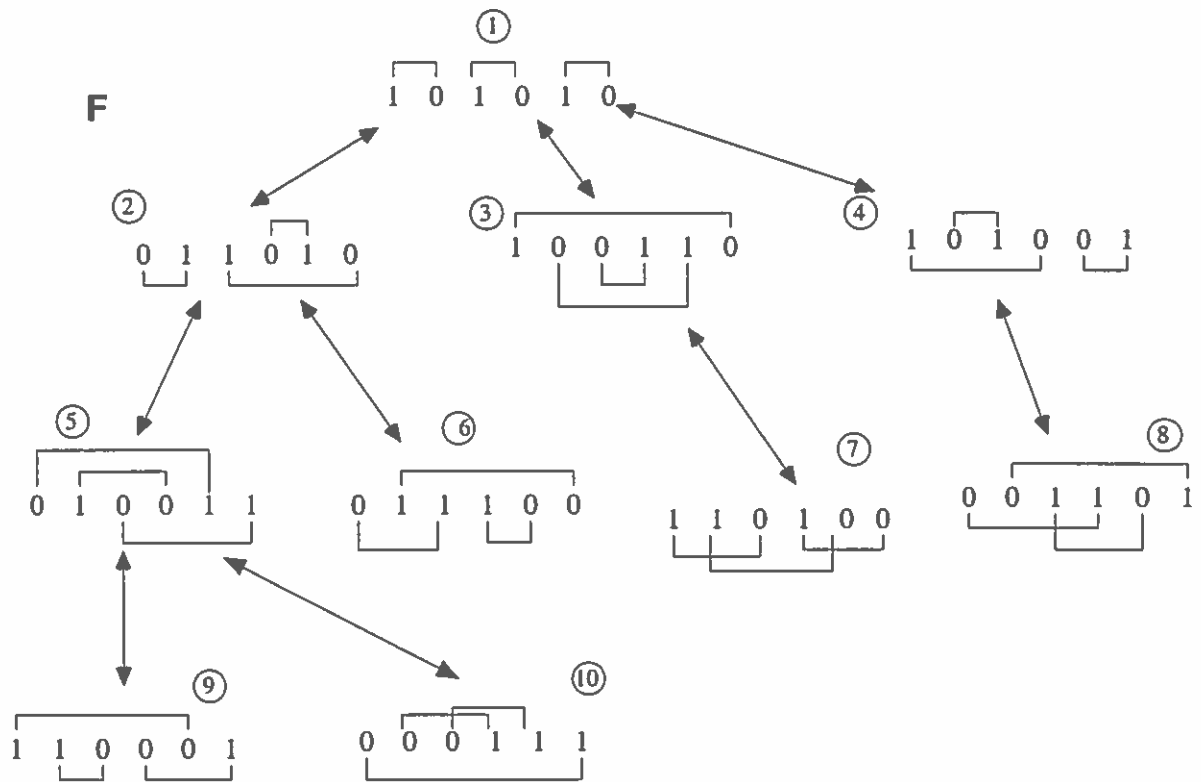
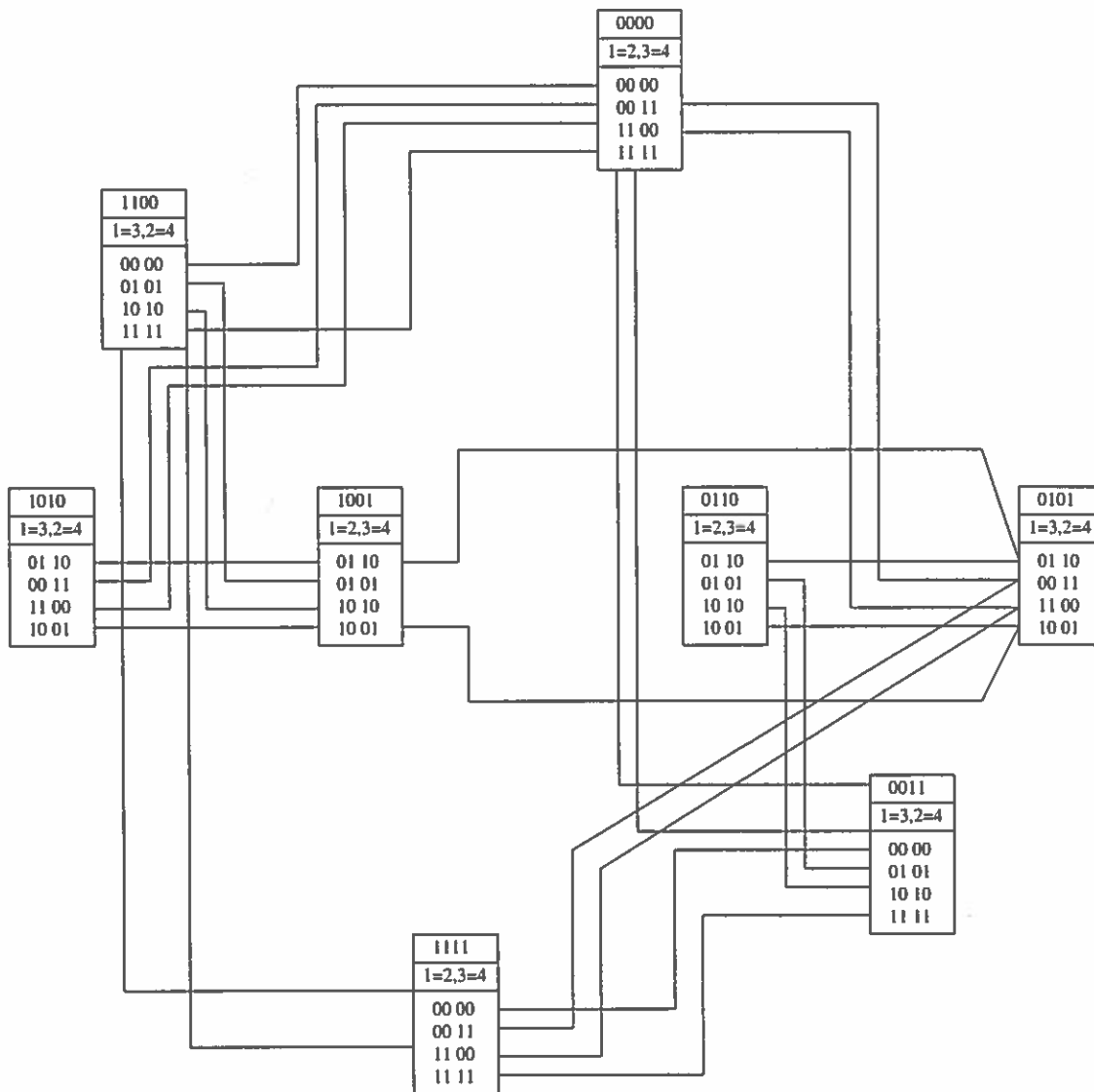


Figure 2: Sparse Stable Family of Size 10 for  $n = 6$

Figure 3: Sparse Stable Family of Size 32 for  $n = 8$

orem 5.16. We also include the break repair pairs for each incidence vector. The structure is remarkably similar to that shown in figure for the sparse set of size 32. The projection in the last 4 coordinates is  $E[4]$ , i.e., 4-bit strings of even parity. If we consider the set of strings with the same last 4 coordinates, then their projections into the first 6 coordinates will form a sparse stable family of size 10 (equivalent to that shown in Figure 2 with some relabeling of coordinates).

```

0000000000[12][34][56][78][910], 1110100000[14][26][35][78][910]
1101010000[15][23][46][78][910], 1100001010[13][24][56][79][810]
1100000101[13][24][56][79][810], 1100000000[12][35][46][79][810]
0011000000[16][25][34][78][910], 0000110000[13][24][56][79][810]
0000001100[16][25][34][78][910], 0000000011[16][25][34][78][910]
0111100000[14][25][36][79][810], 1010110000[13][26][45][78][910]
1110101100[13][26][45][78][910], 1110100011[13][26][45][78][910]
0101110000[15][24][36][78][910], 1011010000[16][23][45][79][810]
1101011100[15][24][36][78][910], 1101010011[15][24][36][78][910]
0110001010[13][26][45][78][910], 1001001010[15][24][36][78][910]
1100111010[12][34][56][78][910], 1100001111[12][35][46][79][810]
0110000101[13][26][45][78][910], 1001000101[15][24][36][78][910]
1100110101[12][34][56][78][910], 0011001100[12][34][56][78][910]
0011000011[12][34][56][78][910], 0000111010[12][35][46][79][810]
0000110101[12][35][46][79][810], 1000011100[16][24][35][79][810]
0100101100[13][25][46][79][810], 0000001111[12][34][56][78][910]
1111001100[12][36][45][79][810], 1100111001[16][25][34][78][910]
1100110110[16][25][34][78][910], 1010111100[14][26][35][78][910]

```

1000010110[13][25][46][79][810], 1000011001[13][25][46][79][810]  
 0101111100[15][23][46][78][910], 0100100110[16][24][35][79][810]  
 0100101001[16][24][35][79][810], 1111000110[14][23][56][79][810]  
 1111001001[14][23][56][79][810], 1111111001[12][34][56][78][910]  
 1111110110[12][34][56][78][910], 1000010011[16][24][35][79][810]  
 0100100011[13][25][46][79][810], 1111000011[12][36][45][79][810]  
 1111110101[16][25][34][78][910], 1111111010[16][25][34][78][910]  
 0011110110[12][36][45][79][810], 1010110011[14][26][35][78][910]  
 0101110011[15][23][46][78][910], 0011110011[14][23][56][79][810]  
 1010111111[13][26][45][78][910], 0101111111[15][24][36][78][910]  
 0011111001[12][36][45][79][810], 0011111100[14][23][56][79][810]  
 0111101010[16][23][45][79][810], 0111100101[16][23][45][79][810]  
 1110101111[14][26][35][78][910], 1011011010[14][25][36][79][810]  
 1011010101[14][25][36][79][810], 1101011111[15][23][46][78][910]  
 0010011010[14][26][35][78][910], 0110000110[14][26][35][78][910]  
 0110001001[14][26][35][78][910], 0001101010[15][23][46][78][910]  
 1001000110[15][23][46][78][910], 1001001001[15][23][46][78][910]  
 0010010101[14][26][35][78][910], 0001100101[15][23][46][78][910]  
 0011001111[16][25][34][78][910], 0000111111[13][24][56][79][810]  
 0010010110[13][26][45][78][910], 0010011001[13][26][45][78][910]  
 0001100110[15][24][36][78][910], 0001101001[15][24][36][78][910]  
 1011011111[16][23][45][79][810], 0111101111[14][25][36][79][810]

An example of a minimal set of size 16 for  $n = 6$  used to prove Theorem 5.28 is below:

$$\{100100, 010100, 001100, 111100 \\ 010010, 111010, 000110, 110110 \\ 101110, 011110, 100001, 001001 \\ 101101, 011101, 000011, 110011\}$$

An essential ingredient in building stable sets is  $E_j([n])$  which is true iff exactly  $j$  variables in  $[n]$  are set to true.  $E_j([n])$  is a stable set with interesting collection of minimal stable sets. In this thesis, we study the structure of  $E_2([n])$ .

Theorem 6.1. Any minimal stable set which is a subset of  $E_2([n])$  can be expressed as a direct product  $E_1(l) \times E_1(r)$ , where  $l$  and  $r$  partition  $[n]$ ,  $|l|, |r| \geq 2$ .

Proof. Let  $\mathcal{C}$  be a minimal closed subset of  $E_2([n])$  (it thus consists of two element subsets). Assume  $X = \{1, 2\} \in \mathcal{C}$ . This corresponds to the string  $X = 1100\dots 0$ , whose membership we can assume wlog by renumbering the bits.

We partition the rest of  $[n]$  into sets  $\alpha$ ,  $\beta$ , and  $\gamma$  as follows. The set  $\alpha$  consists of those indices whose breaks are repaired only by index 1 and not by 2. Formally,

$$\alpha = \{i \in [n] \mid \delta_{1i}(X) \in \mathcal{C} \text{ and } \delta_{2i}(X) \notin \mathcal{C}\}.$$

Similarly,  $\beta$  are those indices repaired by 2 and not 1, while  $\gamma$  will be those repaired by both 1 and 2.

First let us consider the case where  $\gamma = \emptyset$ . Note that we must have  $\alpha \neq \emptyset$ , or else there is no repair for a break to index 1 in  $X$ . Similarly,  $\beta \neq \emptyset$ . Let  $l = \{1\} \cup \alpha$  and  $r = \{2\} \cup \beta$ . We claim that  $E_1(l) \times E_1(r) \subseteq \mathcal{C}$ . From the definitions of  $\alpha$  and  $\beta$ , it follows that sets of the following form are already in  $\mathcal{C}$ :  $\{1, 2\}$ ,  $\{1\} \cup \{b | b \in \beta\}$ , and  $\{2\} \cup \{a | a \in \alpha\}$ . It remains to show that  $\{a, b\} \in \mathcal{C}$  for any  $a \in \alpha$  and  $b \in \beta$ . Consider a break to bit  $b$  of  $\{2, a\} \in \mathcal{C}$ . Since  $\mathcal{C}$  is stable, there exists a repair  $i$  with  $\delta_{ib}(\{2, a\}) \in \mathcal{C}$ . The only choices are  $i = 2$  or  $i = a$  since  $\mathcal{C} \subseteq E_2([n])$ . If  $i = a$ , then  $\{2, b\} = \delta_{ib}(\{2, a\}) \in \mathcal{C}$ , contradicting the definition of  $\beta$ . Therefore,  $i = 1$  and  $\{a, b\} = \delta_{ib}(\{2, a\}) \in \mathcal{C}$ .

Now suppose that  $\gamma \neq \emptyset$ . We can assume that one or both of  $\alpha$  and  $\beta$  are non-empty by the following argument: If  $\mathcal{C} = E_2([n])$  it would not be minimal, so  $\mathcal{C} \neq E_2([n])$ . This implies that there is a set  $\{x, y\} \in \mathcal{C}$  for which at least one of  $x$  or  $y$  is not repaired by *all* indices in  $[n] \setminus \{x, y\}$ . We can then renumber the elements so that  $\{x, y\}$  corresponds to  $X = \{1, 2\}$ .

Consider at this point the case where  $\gamma \neq \emptyset$  and  $\alpha \neq \emptyset$  ( $\beta$  may or may not be empty). Here we set  $l = \{1\} \cup \alpha$  and  $r = \{2\} \cup \beta \cup \gamma$ . As above, we claim that  $E_1(l) \times E_1(r) \subseteq \mathcal{C}$ . And as similar to the above, what we need to show is that  $\{a, b\}, \{a, c\} \in \mathcal{C}$  for any  $a \in \alpha$ ,  $b \in \beta$ , and  $c \in \gamma$ .

The argument that  $\{a, b\} \in \mathcal{C}$  is just as above. To see that  $\{a, c\} \in \mathcal{C}$ , consider the set  $\{1, c\} \in \mathcal{C}$ . When there is a break to index  $a$ , there must be a repair  $i \in [n] \setminus \{a\}$ :  $\delta_{ia}(\{1, c\}) \in \mathcal{C}$ . The only possibilities are  $i = 1$  or  $i = c$ . If  $i = c$ , then  $\{1, a\} = \delta_{ia}(\{1, c\}) \in \mathcal{C}$ . That would mean that an index in  $\alpha$  was a repair for bit 2 in  $X$ , contradicting the definition of  $\alpha$ . Hence, the only choice is  $i = 1$ , giving  $\{a, c\} = \delta_{ia}(\{1, c\}) \in \mathcal{C}$ .

The situation where  $\gamma \neq \emptyset$ ,  $\beta \neq \emptyset$ , and  $\alpha = \emptyset$  is symmetric to the previous case and can be handled in the same manner.

□

Note that while minimal sub-families of  $E_2([n])$  breaks up as direct products, that need not be the case for  $E_3([n])$ : as Figure 2 shows, there is a sub-family of  $E_3([6])$  which is not a direct product.

### 7. Summary and Future Work

The following table summarizes the results from Section 5.

	stable	stable minimal	sparse stable	sparse stable minimal
largest	$2^{n-1}$	$\leq (\frac{2}{3} + o(1)) 2^{n-1}$ $\geq 2^{2n/3}$	$\leq \frac{2^{n+1}}{n+2}$ , $\geq \Omega(2^n/n^2)$	$\leq 2^n/n$ $\geq 80^{n/10}$
smallest		$n$		$2^{n/2}$

Table 3: Upper and Lower Bounds for Stable Families

The most interesting questions here involve tightening the bounds in the table above and understanding the structure of the minimal sets. The sparse minimal sets in particular seem to have a rich combinatorial structure.

## BIBLIOGRAPHY

- [1] Ian Anderson. *Combinatorics of Finite Sets*. Clarendon Press, Oxford, 1987.
- [2] K. Appel and W. Haken. Every planar map is four-colorable. *Bull. Amer. math. Soc*, 82:711–712, 1976.
- [3] L. Babai and L. Kucera. Canonical labelling of graphs in linear average time. In *Proceedings of the Twentieth IEEE Conference on Foundations of Computer Science*, pages 44–49, 1979.
- [4] Laszlo Babai, D. Yu. Grigoryev, and David M. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 310–324, 1982.
- [5] Laszlo Babai, W.M. Kantor, and Eugene M. Luks. Computational complexity and the classification of finite simple groups. In *Proceedings of the 24th IEEE Symp. on the Foundations of Computer Science*, pages 162–171, 1983.
- [6] Laszlo Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proc. 15th ACM Symp. on Theory of Computing*, pages 171–183, 1983.
- [7] Olivier Bailleux and Pierre Marquis. Distance-sat: Complexity and algorithms. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 642–647, 1999.
- [8] Michael Ben-Or, Nathan Linial, and Michael Saks. Collective coin flipping and other models of imperfect randomness. *Combinatorics*, pages 75–112, 1987.
- [9] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In *Handbook of theoretical computer science, Vol. A*, pages 757–804. Elsevier, Amsterdam, 1990.
- [10] Cynthia A. Brown, Larry Finkelstein, and Paul 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.
- [11] Peter Cameron. *Combinatorics: Topics, Techniques, Algorithms*. Cambridge University Press, 1994.
- [12] James 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.



- [13] James Crawford, Matthew Ginsberg, Eugene M. Luks, and Amitabha 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.
- [14] John D. Dixon and Brian Mortimer. *Permutation groups*. Springer-Verlag, New York, 1996.
- [15] Konrad Engel. *Sperner Theory*, volume 65 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1997.
- [16] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, New York, 1979.
- [17] John Gaschnig. Performance measurement and analysis of some search algorithms. Technical report, Carnegie Mellon University, 1979. CMU-CS-79-124.
- [18] Matthew Ginsberg, Andrew Parkes, and Amitabha Roy. Supermodels and robustness. In *AAAI-98/IAAI-98 Proceedings, Madison, WI*, pages 334–339, 1998.
- [19] Matthew L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, 1:25–46, 1993.
- [20] Holgar Gollan. A new existence proof for  $Ly$ , the sporadic simple group of rank 3. *J. Symbolic Computation*, 31:203–209, 2001.
- [21] M. Hall. *The theory of groups*. Chelsea Publishing Company, New York, second edition, 1976.
- [22] David Joslin and Amitabha 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.
- [23] Gyula Katona and Jaya Srivastava. Minimal 2-coverings of a finite affine space based on  $GF(2)$ . *J. Statist. Plann. Inference*, 8(3):375–388, 1983.
- [24] Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Birkhäuser Boston Inc., Boston, MA, 1993.
- [25] C. W. H. Lam. Applications of group theory to combinatorial searches. In L. Finkelstein and W. M. Kantor, editors, *Groups and Computation, Workshop on Groups and Computation*, volume 11 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 133–138, 1993.

- [26] 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.
- [27] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. Assoc. Comput. Mach.*, 40(3):607–620, 1993.
- [28] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comp. Sys. Sci.*, 25:42–65, 1982.
- [29] Eugene M. Luks and Amitabha Roy. Triangle free subsets of the boolean lattice. unpublished manuscript, 2001.
- [30] Brendan McKay. nauty user's guide. Technical report, Department of Computer Science, Australian National University, 1990. TR-CS-90-02.
- [31] D. Miklós. Linear binary codes with intersection properties. *Discrete Appl. Math.*, 9(2):187–196, 1984.
- [32] Steve Minton, Mark D. Johnson, Andrew B. Phillips, and Phillip Laird. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 17–24, 1990.
- [33] Morris Newman. *Integral Matrices*. Academic Press, 1972.
- [34] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [35] Steven Roman. *Coding and Information Theory*. Springer-Verlag, 1992.
- [36] John Savage. *Models of Computation, Exploring the Power of Computing*. Addison-Wesley, 1998.
- [37] T. Schaefer. The complexity of satisfiability problems. In *Proc. of the 10th ACM STOC*, 1978.
- [38] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, 7:281–292, 1971.
- [39] Bart Selman, Hector Levesque, and David Mitchell. A new method to solve hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, 1992.
- [40] Harold N. Shapiro. *Introduction to the Theory of Numbers*. Wiley-Interscience, 1983.

- [41] Charles C. Sims. Computational methods in the study of permutation groups. In *Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967)*, pages 169–183. Pergamon, Oxford, 1970.
- [42] E. Sperner. Ein satz über untermengen einer endlichen menge. *Math. Z.*, 27:544–8, 1928.
- [43] J. H. van Lint. *Introduction to Coding Theory, Second Edition*. Springer-Verlag, 1992.
- [44] Douglas B. West. Extremal problems in partially ordered sets. In Ivan Rival, editor, *Ordered Sets, Proceedings of the NATO Advanced Study Institute held at Banff, Canada, Aug 28 – September 12 1981*.
- [45] H. Wielandt. *Finite Permutation Groups*. Academic Press, New York, 1964.