2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Exponential Lower Bounds for Monotone Span Programs

Robert Robere, Toniann Pitassi, Benjamin Rossman, Stephen A. Cook
Department of Computer Science
University of Toronto
Toronto, Canada
$\{$*robere, toni, rossman, sacook*$\}$*@cs.toronto.edu*

***Abstract*—Monotone span programs are a linear-algebraic model of computation which were introduced by Karchmer and Wigderson in 1993 [1]. They are known to be equivalent to linear secret sharing schemes, and have various applications in complexity theory and cryptography. Lower bounds for monotone span programs have been difficult to obtain because they use non-monotone operations to compute monotone functions; in fact, the best known lower bounds are quasipolynomial for a function in (nonmonotone) P [2]. A fundamental open problem is to prove exponential lower bounds on monotone span program size for *any* explicit function.**

We resolve this open problem by giving exponential lower bounds on monotone span program size for a function in monotone P. This also implies the first exponential lower bounds for linear secret sharing schemes. Our result is obtained by proving exponential lower bounds using Razborov's rank method [3], a measure that is strong enough to prove lower bounds for many monotone models. As corollaries we obtain new proofs of exponential lower bounds for monotone formula size, monotone switching network size, and the first lower bounds for monotone comparator circuit size for a function in monotone P. We also obtain new polynomial degree lower bounds for Nullstellensatz refutations using an interpolation theorem of Pudlak and Sgall [4]. Finally, we obtain quasipolynomial lower bounds on the rank measure for the st-connectivity function, implying tight bounds for st-connectivity in all of the computational models mentioned above.

***Keywords*-Comparator Circuits, Lower Bounds, Monotone, Nullstellensatz, Secret Sharing, Span Programs, Switching Networks**

I. Introduction

Razborov [3] introduced a simple matrix-theoretic technique (which we will call the *rank method*) to study lower bounds on formula size for boolean functions, and using this method he was able to give a simple proof that any monotone formula computing a certain monotone function in NP must have size at least $n^{\Omega(\log n)}$. While not the strongest lower bound known against monotone formula size — similar bounds were already known for st-connectivity, and stronger lower bounds are known for other functions — Razborov's method is exceptionally elegant, and applies to models of computation that seem to be out of reach of standard techniques. Two examples of such models are *monotone span programs* and *monotone switching networks* [1], [5]: monotone span programs use non-monotone (algebraic) operations to compute monotone functions, which makes them remarkably powerful and technically difficult to lower bound [1], [2], [5]–[9]; monotone switching networks are a classic model which resisted strong lower bounds for directed st-connectivity until Potechin [10] gave an ingenious Fourier-analytic argument.

Despite its elegance and connections with other models, very little is known about Razborov's rank method. In fact, Razborov's original argument is the only known lower bound for the rank measure, giving a quasipolynomial lower bound for a function in NP. This suggests several natural questions: First, is it possible to use Razborov's rank method to give nontrivial lower bounds for a function in P, or even monotone P? Secondly, can the rank method be used to prove exponential size lower bounds?

In this paper we resolve both of these problems. First, we prove $n^{\Omega(\log n)}$ lower bounds for directed st-connectivity using the rank method. Directed st-connectivity is one of the most basic functions: it is the canonical NL-complete problem and can be computed by polynomial-size, $O(\log^2 n)$-depth monotone circuits. Thus, our proof gives new (and arguably simpler) proofs of some celebrated results: it implies both Potechin's lower bound for monotone switching networks [10], as well as the classic Karchmer-Wigderson lower bound for monotone formulas [11].

Second, we prove exponential size lower bounds using the rank method against the GEN function, which is computable in monotone P. As well as being the first exponential lower bounds using the rank method, this implies both exponential lower bounds on monotone span program size for a function in monotone P (solving a well-known open problem), as well as the first exponential lower bounds for linear secret sharing schemes.

In addition, we show how to apply the rank method to *monotone comparator circuits*, which allows us to prove the first nontrivial lower bounds for any family of these circuits computing a function in monotone P. Before this, no size lower bounds were known for monotone comparator circuits except those implied by the classic lower bounds for clique and perfect matching [12].

Robert Robere, Toniann Pitassi, and Stephen A. Cook supported by NSERC. Benjamin Rossman supported by the JST ERATO Kawarabayashi Large Graph Project.

A. Monotone Span Programs and Related Models

Let $\mathbb{F}$ be any field. A span program over $\mathbb{F}$ is a model of computation which uses linear algebraic operations over $\mathbb{F}$ to compute boolean functions. Span programs were introduced by Karchmer and Wigderson [1], where they showed that non-monotone span programs capture logspace counting classes such as $\oplus\mathsf{L}$ and $\mathsf{Mod}_p\mathsf{L}$; monotone span programs are also known to characterize a subclass of secret sharing schemes known as *linear secret sharing schemes* [13], [14].

There is a fairly long history of lower bounds for monotone span programs. The first lower bounds for monotone span programs, due to Karchmer and Wigderson [1], showed that all threshold functions over $\mathrm{GF}(2)$ require monotone span programs of size $\Omega(n \log n)$. The first superpolynomial lower bounds, on the order of $n^{\Omega(\log n/\log\log n)}$, were obtained by Babai et al. [7] against a function in NP. These bounds were simplified and improved by Gál to $n^{\Omega(\log n)}$, who also observed the connection between monotone span programs and the rank method [5]. Beimel and Weinreb [2] later gave $n^{\Omega(\sqrt{\log n})}$ lower bounds for a function in uniform NC^2 (therefore for a function in P), proving that monotone span programs can be weaker than polynomial time.

An interesting feature of monotone span programs is that they are not "really" monotone – monotone span programs use non-monotone operations to compute monotone functions. Babai et al. [7] exploited this to give a function with linear size monotone span programs that requires superpolynomial-size monotone circuits and exponential-size monotone formulas. This immediately implies that the size and depth lower bound methods for monotone circuits cannot be used to prove lower bounds for monotone span programs

Due to this fact there are several basic open problems concerning monotone span programs. First, it is open to prove exponential lower bounds on monotone span program size for any explicit function. Second, it is open to show whether there are functions in monotone P that require monotone span programs of superpolynomial size. Third, it is open to give an example of a function with small (non-monotone) span programs but requiring large monotone span programs.

Our main result for span programs is the following theorem.

Theorem I.1. *The st-connectivity function requires $n^{\Omega(\log n)}$ size monotone span programs over $\mathbb{R}$. The GEN function requires $\exp(n^{\Omega(1)})$ size monotone span programs over $\mathbb{R}$.*

This resolves all of the open problems mentioned above. First, our lower bound for the GEN function is the first lower bound on monotone span program size greater than $n^{\Omega(\log n)}$ for any explicit function. Since GEN is in monotone P, this implies an exponential separation between monotone P and monotone span program size, resolving the second open problem mentioned above. Furthermore, our lower bound for st-connectivity implies a quasipolynomial separation between mNC^2 and monotone span programs, since st-connectivity is well-known to be computable by polynomial-size, $O(\log^2 n)$ depth monotone circuits. Finally, Karchmer and Wigderson showed that non-uniform polynomial-size span programs over $\mathrm{GF}(2)$ compute exactly those functions in $\oplus\mathsf{L}/\mathsf{poly}$. Wigderson [15] showed that $\mathsf{NL}/\mathsf{poly} \subseteq \oplus\mathsf{L}/\mathsf{poly}$, and since directed st-connectivity is in NL it follows it is computable by (non-uniform) polynomial-size span programs. Thus, we exhibit a function with small non-monotone span programs but requiring large monotone span program size.

Secret Sharing Schemes. A *secret sharing scheme* is a cryptographic tool where a dealer shares a secret among a set of participants such that only the "authorized" subsets of participants are able to reconstruct the secret [13]. The subsets correspond to a monotone boolean function f on n bits, where n is the number of participants. Monotone span program size measures the amount of information that has to be given to the participants in so-called *linear* secret sharing schemes [1], [16]. Our result for GEN gives the first exponential lower bounds on the size of linear secret sharing schemes. For s-t connectivity, our quasipolynomial lower bound is especially striking due to the known polynomial upper bounds for the access structure corresponding to *undirected* s-t connectivity.

Switching Networks. Switching networks are a non-uniform model of computation used to study space complexity which are closely related to monotone circuits. It was long conjectured that *directed* st-connectivity required quasi-polynomial size monotone switching networks, which was resolved affirmatively by Potechin [10]. Extending this, Chan and Potechin [17] proved asymptotically tight $n^{\Omega(h)}$ lower bounds for the GEN function on pyramids of height h. (Weaker size lower bounds were previously obtained by Raz and McKenzie [18].)

It is known that if a function can be computed by a switching network of size S, then it can also be computed by a span program of size S over any field, and the same holds for their monotone versions (see [1] for a proof). By this simulation we get alternative proofs of the results of Potechin and Chan-Potechin.

Nullstellensatz. Nullstellensatz (NS) refutations are a natural algebraic proof system for proving unsolvability of systems of polynomial equations based on Hilbert's Nullstellensatz [19]. Given a set of polynomial equations $p_1 = 0, \ldots, p_m = 0$, an NS refutation is given by a sequence of polynomials $q_1, \ldots, q_m$ such that $\sum p_i q_i = 1$. The degree of the refutation is $d = \max_i \deg(q_i p_i)$, and the size is the total number of monomials in all of the polynomials.

Pudlak and Sgall [4] proved a strong connection between Nullstellsatz refutations and span programs. In particular, they proved that interpolants for degree-d refutations are exactly characterized by size $n^{O(d)}$ span programs, and the characterization also holds in the monotone setting. By this characterization, our lower bounds for monotone span programs imply strong (i.e. polynomial) lower bounds on the degree of NS refutations.

B. Monotone Comparator Circuits

A *sorting network* is a model of a sorting algorithm which is input-oblivious. The model is quite simple: the network receives as input n integers on n parallel wires travelling from left to right, with a sequence of *comparator gates* connecting pairs of wires that map $(x, y) \mapsto (\min\{x, y\}, \max\{x, y\})$. The goal is to sort all n integer inputs using the fewest number of gates (or, alternatively, with the smallest depth). Shallow sorting networks have many applications in theoretical computer science, and explicit constructions have been extensively studied in the literature (see [20] for an extensive survey).

When the inputs are restricted to be boolean a sorting network is called a *comparator circuit*. The class of problems computable by polynomial-size uniform comparator circuits is called CC. There are many interesting complete problems for CC such as the stable marriage problem [21], [22] and the telephone connection problem [23]. The structure of CC was initially studied by Subramanian [21], and further studied by Cook, Filmus and Le [22].

Monotone comparator circuits are a natural restriction of comparator circuits where the input bits are either constants or positive literals; we let mCC denote the class of languages computable by polynomial-size monotone comparator circuits. Essentially nothing is known on the complexity of monotone comparator circuits computing arbitrary monotone boolean functions. It is easy to see that monotone comparator circuits can simulate monotone formulas, and in turn can be simulated by (unrestricted) monotone circuits, but the relationship with mP and mNC was open.

We are able to show that the rank method applies to monotone comparator circuits, and so we obtain analogous results for monotone comparator circuits — separating mCC from mP, mNC^2 and CC.

C. Overview of Proof

We will explain the main ideas in the context of the st-connectivity function, although the argument easily generalizes to any GEN function. Consider the *layered st-connectivity function* $\mathrm{STCONN} = \mathrm{STCONN}_{h,w}$, for which the input is a list of edges encoding a subgraph of the layered, directed graph with w layers and h nodes per layer. Let $U \subseteq \mathrm{STCONN}^{-1}(1)$ and $V \subseteq \mathrm{STCONN}^{-1}(0)$, and let A be a $|U| \times |V|$ matrix over $\mathbb{R}$ with rows labelled by $u \in U$ and columns labelled by $v \in V$. For each underlying input variable x_e of STCONN, define the rectangle R_e to be the set of pairs $(u, v) \in U \times V$ such that $u_e = 1$ and $v_e = 0$. Let $\mathcal{R}_{\mathrm{STCONN}}(U, V)$ denote the collection of all of these rectangles.

The *rank measure* of A is defined to be the ratio of the rank of A and the maximum rank of the submatrix of A indexed by any of these rectangles

$$\mu_A(\mathrm{STCONN}) = \frac{\operatorname{rank} A}{\max_{R \in \mathcal{R}_{\mathrm{STCONN}}(U,V)} \operatorname{rank} A\restriction_R}.$$

This measure was originally introduced by Razborov [3], and for any A the measure $\mu_A(\mathrm{STCONN})$ is a lower bound on each of the monotone computation models we have discussed above. Thus, our overall goal is to find a family of matrices $\{A_n\}$ for the STCONN function for which the rank measure is $n^{\Omega(\log n)}$.

At a high-level our argument is a reduction from the rank measure to reversible pebbling number which proceeds in two steps. First, we prove a "lifting theorem" connecting the rank measure to a new algebraic complexity measure on boolean functions that we call the *algebraic gap complexity*, which may be of independent interest. The second step is to actually prove a lower bound on the gap complexity.

Step 1: The Pattern Matrix Lift. Sherstov [24] gave a general method to construct a "pattern matrix" A_p from a boolean function $p : \{0, 1\}^m \to \mathbb{R}$ such that the properties of A_p are related to the Fourier spectrum of p.

The main idea is to use a pattern matrix A_p (for a suitably chosen p) to certify a lower bound on the rank measure, using a theorem of Sherstov [24] showing that the rank of pattern matrices can be directly calculated from the Fourier spectrum of the function p used to generate the matrix. With this in mind, we show that the rows of the pattern matrix A_p can be interpreted as rejecting instances of the st-connectivity function (specifically a collection of s-t cuts) and the columns of A_p can be interpreted as accepting instances of the st-connectivity function (specifically a collection of s-t paths with length $m+1$). Using Sherstov's rank theorem we then calculate the rank of A_p directly from p, as well as the rank of each "rectangle submatrix" of A_p from $p\restriction_e$, where $p\restriction_e$ is a restriction of the function p obtained naturally from the edge e underlying the rectangle R_e. This implies that the matrix A_p will certify a large rank measure if the function p exhibits a large *algebraic gap*, in that the Fourier degree of p is large, but the Fourier degree of each of the restrictions $p\restriction_e$ is small.

Step 2: Exhibiting Large Algebraic Gaps. The second step of our argument is to actually construct a function p exhibiting large algebraic gaps. We first show that for each positive integer m, the problem of constructing a boolean function $p : \{0, 1\}^m \to \mathbb{R}$ with gap k is equivalent to the satisfiability of an (exponentially large) system of linear equations $\mathcal{E}(k)$. For st-connectivity, we show that the system $\mathcal{E}(k)$ is satisfiable if resolution cannot refute a corresponding unsatisfiable CNF formula within depth k. Since resolution-depth is equivalent to the decision tree-complexity of the corresponding search problem, our lower bound follows from the known $\Omega(\log m)$ lower bound on the reversible pebbling number of the m-node path graph. More generally, we prove that if we start with the a GEN function with accepting instances isomorphic to some template graph G, then algebraic gaps for the associated search problem can be obtained from lower bounds on the reversible pebbling number of G.

D. Related Work

Razborov [3] introduced the rank measure and proved $n^{\Omega(\log n)}$ lower bounds for a function in NP by using the disjointness matrix; in a later work [25] he showed that the rank measure cannot give superlinear lower bounds in non-monotone models of computation. Razborov's lower bound on the rank measure was studied by Gál and Pudlak [6], where it was shown to be related to the method of avoiding families used in monotone span program lower bounds [5], [7].

Karchmer and Wigderson [1] showed that monotone span program size upper bounds the size of *linear secret sharing schemes*; Beimel showed that they give an exact characterization [16]. See the comprehensive survey of Beimel for more on secret sharing schemes [14]. Span programs have also been connected to quantum algorithms [26].

The idea of "lifting" lower bounds on a simple complexity measure from weak to strong computation models has appeared in many forms, and has been enormously successful for proving lower bounds for a variety of models. The basic idea is to start with an "outer" function f for which we have given a lower bound in a weak model of computation, and "lift" f by composing f with an "inner" function g to get a new function, $f \circ g^n$ that is provably hard in a stronger model of computation. This method has led to many lower bounds in communication complexity, in classical, quantum, and number-on-forehead models [24], [27]–[29]. In circuit complexity similar approaches have led to strong lower bounds on monotone circuit depth [18], [30], and similarly for lower bounds in proof complexity [30], [31]. Other lifting techniques have given strong lower bounds against extended formulations of linear programs [32], [33].

II. Definitions

A real-valued boolean function is any function $p : \{0,1\}^n \to \mathbb{R}$. If A is any set and $x \in A^n$ we let x_i denote the ith component of x. If $x, y \in \{0,1\}^n$ we let $x \oplus y \in \{0,1\}^n$ denote the string obtained by taking the bitwise XOR of x and y.

For any n, the collection of all n-ary real-valued boolean functions $\{p : \{0,1\}^n \to \mathbb{R}\}$ forms a vector space under pointwise addition and scalar multiplication. For any $C \subseteq [n]$, the *Fourier character* at C is the function $\chi_C : \{0,1\}^n \to \{-1,1\}$ defined by $\chi_C(x) = (-1)^{\sum_{i \in C} x_i}$. The collection of characters $\{\chi_C\}_{C \subseteq [n]}$ form an orthonormal basis for the vector space of real-valued boolean functions known as the *Fourier basis*, where the vector space is equipped with the inner product $\langle p, q \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} p(x)q(x)$. Since this basis is orthonormal, given any function $p : \{0,1\}^n \to \mathbb{R}$, we can represent p in the Fourier basis as $p(x) = \sum_{C \subseteq [n]} \langle p, \chi_C \rangle \chi_C(x)$. This representation is called the *Fourier transform* of p.

We let $\hat{p}(C) = \langle p, \chi_C \rangle$ denote the coefficient of χ_C of p in the Fourier basis — this is the *Fourier coefficient* of p at C. The collection of non-zero Fourier coefficients of p is called the *Fourier spectrum* of p. The *Fourier degree* is the size of the largest non-zero Fourier coefficient of p: $\deg p = \max_{S \subseteq [m]} \{|S| \mid \hat{p}(S) \neq 0\}$, which, equivalently, is the degree of the unique representation of p as a multilinear polynomial over the real numbers.

If $x, y \in \{0,1\}^n$ then we write $x \leq y$ if $x_i \leq y_i$ for all i. A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if $f(x) \leq f(y)$ whenever $x \leq y$. If $f(x) = 1$ we call x an *accepting instance* or a *yes instance*, while if $f(x) = 0$ then we call x a *rejecting instance* or a *no instance*. If x is any yes instance of f and y is any no instance of f then there exists an index $i \in [n]$ such that $x_i = 1, y_i = 0$.

Suppose that $U, V \subseteq \{0,1\}^n$ are any sets satisfying $f(U) = 1, f(V) = 0$. A set $R \subseteq U \times V$ is called a *rectangle* if there are sets $U_0 \subseteq U, V_0 \subseteq V$ such that $R = U_0 \times V_0$. For each $i \in [n]$ let

$$X_i = \{x \in \{0,1\}^n \mid x_i = 1\} \times \{x \in \{0,1\}^n \mid x_i = 0\},$$

and let $R_i = X_i \cap (U \times V)$. Let $\mathcal{R}_f(U,V) = \{R_i \mid i = 1, 2, \ldots, n\}$. Since f is a monotone function there is an index i such that $u_i = 1, v_i = 0$ for all $u \in U, v \in V$, and so every entry of $U \times V$ is covered by some rectangle in $R_f(U,V)$. Let A be any $|U| \times |V|$ matrix with rows labelled by entries of U and columns labelled by entries of V, and if $S \subseteq U \times V$ is any subset of $U \times V$ let $A\restriction_S$ be the submatrix indexed by S.

Definition II.1. Let $f : \{0,1\}^n \to \{0,1\}$ and let $U \subseteq f^{-1}(1), V \subseteq f^{-1}(0)$. Let A be any $|U| \times |V|$ matrix over $\mathbb{R}$[1]. The *rank measure* of f with respect to A is

$$\mu_A(f) := \frac{\operatorname{rank}(A)}{\max\limits_{R \in \mathcal{R}_f(U,V)} \operatorname{rank}(A\restriction_R)}.$$

The rank measure was introduced by Razborov [3] to give simple superpolynomial lower bounds on the size of monotone boolean formulas. In this paper we give a similar result for variants of the GEN problem.

Definition II.2. Let n be a positive integer, and let $L \subseteq [n]^3$ be a collection of triples on $[n]$. For any subset $S \subseteq [n]$, the set of points *generated* from S by L is defined recursively as follows: every point in S is generated from S, and if i, j are generated from S and $(i, j, k) \in L$, then k is also

[1]This definition makes sense with respect to any field, but we will work exclusively in the reals.

generated from S. The GEN problem is as follows: given a collection of triples of vertices L and two distinguished points $s, t \in [n]$, decide if t generated from $\{s\}$.

Formally, an instance of GEN is given by two nodes $s, t \in [n]$ and n^3 boolean values coding the set $L \subseteq [n]^3$. For definiteness, in the remainder of the paper assume s, t are arbitrary fixed points in $[n]$, and we let GEN denote the corresponding monotone function.

We can naturally some graphs with GEN instances.

Definition II.3. A DAG $G = (V, E)$ is *good* if it is connected, has maximum in-degree 2, and has a unique sink node.

If G is a good DAG then we can form an instance of GEN from G by (1) Adding triples connecting the source point s to the sources of G (2) Adding a triple connecting the sink node of G to the target t and (3) For each internal node z, if z has in-degree 2 with distinct in-edges $(x, z), (y, z)$, then add a triple (x, y, z); Otherwise, if z has in-degree 1 with an in-edge (x, z), then add the triple (x, x, z). We say input triples obtained in this way are *legal*. The lower bounds in this paper are proven for sub-problems of GEN obtained by "lifting" good graphs G.

Definition II.4. Let G be a good DAG, let t be the sink node of G, and let o be a positive integer. The *o-lifted graph* $G^{\uparrow o}$ is obtained by taking the tensor product of G with the complete directed graph on o vertices, and then adding a "super-source" node $\mathbf{s}$ and a "super-target" node $\mathbf{t}$. Explicitly, $G^{\uparrow o}$ is obtained from G as follows: we replace each node $u \in G$ with o copies $\{u^{(1)}, u^{(2)}, \ldots, u^{(o)}\}$. For each edge (u, v) add o^2 edges $(u^{(i)}, v^{(j)})$ for all $i, j \in [o]$. Finally, add a new source node $\mathbf{s}$ and a new target node $\mathbf{t}$ and add edges connecting $\mathbf{s}$ to the lifted source nodes $u^{(i)}$, as well as edges connecting the lifted sink nodes $t^{(i)}$ to the target node $\mathbf{t}$.

Given a node $u^{(i)} \in G^{\uparrow o}$ let $\pi(u^{(i)}) = u$ be the underlying node in the graph G. The $G^{\uparrow o}$-GEN *problem* is a subproblem of GEN obtained by restricting the allowed input triples to those triples of vertices $(u, v, w) \in G^{\uparrow o}$ such that $(\pi(u), \pi(v), \pi(w))$ is a legal triple of the underlying graph G.

The following proposition connects GEN and st-connectivity.

Proposition II.5. *Let m, o be positive integers. Let P_m be the directed path graph with m nodes and let* $\mathrm{STCONN}_{o,m}$ *be the st-connectivity function on the graph $P_m^{\uparrow o}$. Then* $\mathrm{STCONN}_{o,m} = P_m^{\uparrow o}$-GEN. *(See Figure 1.)*

We also need a variant of the well-known black pebbling game on DAGs [10], [17].

Definition II.6. Let $G = (V, E)$ be a good DAG with sources R and a unique sink t, and we define the *reversible*

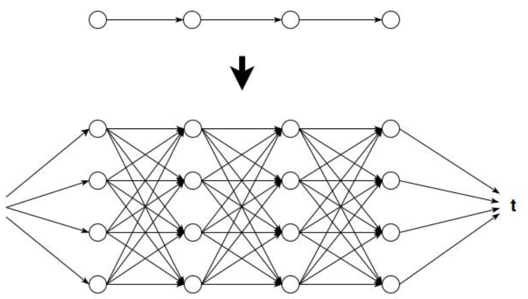

Figure 1. A path P_4 and the lifted graph $P_4^{\uparrow 4}$ (we have added a new source node $\mathbf{s}$ and a new target node $\mathbf{t}$ connected to the lifted source and target nodes). The function $P_4^{\uparrow 4}$-GEN is exactly the layered s-t connectivity problem $\mathrm{STCONN}_{4,4}$.

pebbling game as follows. A *pebble configuration* is a subset $S \subseteq V$ of "pebbled" vertices. For every $x \in V$ such that the in-neighbours of x are pebbled, a *legal pebbling move* consists of either pebbling or unpebbling x (i.e. updating $S = S \cup \{x\}$ or $S = S \setminus \{x\}$). Since the source nodes $s \in R$ do not have any in-neighbours they can always be pebbled or unpebbled.

The goal of the reversible pebbling game is as follows: starting with the empty configuration, place a pebble on t using only legal pebbling moves such that the maximum number of pebbles in any pebbling configuration is minimized. Formally, we want to find a sequence of pebbling configurations $\emptyset = S_0, S_1, \ldots, S_n$ such that $t \in S_n$, and for each $i \in \{0, \ldots, n-1\}$, the configuration S_{i+1} is reachable from configuration S_i by a legal pebbling move. We call such a sequence a *pebbling sequence* for G. The *cost* of the sequence is $\max_i |S_i|$. The *reversible pebbling number* of a DAG G, denoted $\mathsf{rpeb}(G)$, is the minimum cost of a reversible pebbling sequence for G.

III. Rank Measure Lower Bounds

The main result in this paper is a lower bound on the rank measure of $G^{\uparrow o}$-GEN in terms of the reversible pebbling number of the underlying graph G.

Theorem III.1. *Let G be any good DAG with m vertices. There is a real matrix A such that*

$$\mu_A(G^{\uparrow 2m^2}\text{-GEN}) \geq \Omega(m^{\mathsf{rpeb}(G)}).$$

This theorem implies a number of lower bounds in monotone complexity theory, both old and new. In the remainder of this section we use Theorem III.1 to give proofs of each of these lower bounds. We begin by introducing the graphs which are used to prove our exponential lower bounds.

Definition III.2. A *pyramid graph* with h levels is defined as follows. Introduce $h(h-1)/2$ vertices V, partitioned into h sets $V_1, V_2, \ldots, V_h$ where V_i has i vertices. Order V_i as $v_{i,1}, v_{i,2}, \ldots, v_{i,i}$; then for each $i = 2, 3, \ldots, h$, if $v_{i,j}$ and

$v_{i,j+1}$ are adjacent vertices in V_i add two edges $(v_{i,j}, v_{i-1,j})$ and $(v_{i,j+1}, v_{i-1,j})$.

Theorem III.3. *Let h be a positive integer, let Δ_h be the pyramid graph with h levels, and let $m = \binom{h}{2}$ be the number of nodes in Δ_h. Let $N = O(m^7)$ be the number of input triples to $\Delta_h^{\uparrow 2m^2}$-GEN. Let $\varepsilon = 1/14$. Then there is a real matrix A such that*

$$\mu_A(\Delta_h^{\uparrow 2m^2}\text{-GEN}) \geq 2^{\Omega(N^{\varepsilon} \log N)}.$$

Proof: It follows from [34] that $\mathsf{rpeb}(\Delta_h) \geq h$. Since $h = \Omega(\sqrt{m})$ and $m = \Omega(N^{1/7})$, setting $\varepsilon = 1/14$ and applying Theorem III.1 yields $\mu_A(\Delta_h^{\uparrow 2m^2}\text{-GEN}) \geq m^{\Omega(h)} \geq N^{\Omega(N^{\varepsilon})} \geq 2^{\Omega(N^{\varepsilon} \log N)}$. ■

We remark that Gilbert and Tarjan [35] constructed a family of good DAGs with reversible pebbling number $\Omega(n/\log n)$, which is known to be tight due to an upper bound by Dymond and Tompa [36]. One could use these graphs to obtain lower bounds, but essentially this would just improve the value of ε in the previous theorem from 1/14 to 1/7.

We also use a lower bound by Potechin [10] on the reversible pebbling number of path graphs to give a lower bound for st-connectivity.

Theorem III.4. *Let m be a positive integer, let P_m be the directed path with m nodes, and consider the lifted path graph $P_m^{\uparrow 2m^2}$. Let $N = O(m^5)$ be the number of input triples to $\mathrm{STCONN}_{2m^2,m}$. Then there is a real matrix A such that*

$$\mu_A(\mathrm{STCONN}_{2m^2,m}) \geq N^{\Omega(\log N)}.$$

Proof: Proposition II.5 shows that $P_m^{\uparrow 2m^2}$-GEN is exactly $\mathrm{STCONN}_{2m^2,m}$. Potechin [10] proved $\mathsf{rpeb}(P_m) \geq \Omega(\log m)$, so using the fact that $m = \Omega(N^{1/5})$ and applying Theorem III.1 implies $\mu_A(\mathrm{STCONN}_{2m^2,m}) \geq m^{\Omega(\log m)} \geq N^{\Omega(\log N)}$. ■

A. Span Program Lower Bounds and Corollaries

Let $\mathbb{F}$ be any field, and let $\vec{1}$ be the all-1s vector. A *monotone span program* over $\mathbb{F}$ is a matrix M with its rows labelled by boolean variables $x_1, \ldots, x_n$. On an input $x \in \{0,1\}^n$, let M_x denote the submatrix of M containing all rows labelled with variables set to 1 by x. The program *accepts* the input if $\vec{1}$ lies in the linear span of the rows of M_x. Note that any monotone span program computes a monotone function since the linear span of a set of vectors is monotone nondecreasing. Let $\mathsf{mSP}_{\mathbb{F}}$ denote the set of all monotone functions computable by polynomial-size monotone span programs over $\mathbb{F}$, and let $\mathsf{mSP}_{\mathbb{F}}(f)$ denote the size of the smallest monotone span program over $\mathbb{F}$ computing f.

Gál [5] showed that the rank measure is a lower bound on monotone span program size.

Theorem III.5 (Lemma 3.2 and Theorem 3.4 in [5])**.** *Let $\mathbb{F}$ be any field, let f be any monotone boolean function. For any matrix A we have*

$$\mu_A(f) \leq \mathsf{mSP}_{\mathbb{F}}(f).$$

Theorems III.3 and III.4 therefore give exponential and superpolynomial lower bounds, respectively, on the size of *real* monotone span programs computing $\Delta_h^{\uparrow 2m^2}$-GEN and $\mathrm{STCONN}_{2m^2,m}$, showing $\mathsf{mP} \not\subseteq \mathsf{mSP}$ and $\mathsf{mNC}^2 \not\subseteq \mathsf{mSP}$.

Theorem III.6. *Let h be a positive integer, let m be the number of nodes in the height-h pyramid graph Δ_h, and let N be the number of input variables to the function $\Delta_h^{\uparrow 2m^2}$-GEN. Let $\varepsilon = 1/14$. Then $\mathsf{mSP}_{\mathbb{R}}(\Delta_h^{\uparrow 2m^2}\text{-GEN}) \geq 2^{\Omega(N^{\varepsilon} \log N)}$. Similarly, for any positive integer m, if N is the number of input variables to $\mathrm{STCONN}_{2m^2,m}$ then $\mathsf{mSP}_{\mathbb{R}}(\mathrm{STCONN}_{2m^2,m}) \geq N^{\Omega(\log N)}$.*

It is known that polynomial-size, non-monotone span programs can compute STCONN if the span programs are allowed to be non-uniform [15], and so the previous theorem also separates monotone span programs from non-monotone span programs. Furthermore, since monotone span programs can simulate monotone switching networks, the above two theorems give alternative proofs of the recent results by Potechin [10] and Chan-Potechin [17] that STCONN requires superpolynomial-size monotone switching networks and $\Delta_h^{\uparrow 2m^2}$-GEN requires exponential-size monotone switching networks.

Corollary III.7. *Any monotone switching network computing $\Delta_h^{\uparrow 2m^2}$-GEN requires $2^{\Omega(N^{\varepsilon} \log N)}$ states, where $\varepsilon = 1/14$. Any monotone switching network computing $\mathrm{STCONN}_{2m^2,m}$ requires $N^{\Omega(\log N)}$ states.*

A *secret sharing scheme* is a basic cryptographic tool roughly defined as follows (we follow the presentation in [14]; we refer the interested reader there for formal definitions and numerous applications). We have a *dealer* who has a "secret" (say, an element of some field $\mathbb{F}$), a collection of n parties, and a collection $\mathcal{A} \subseteq 2^{[n]}$ of subsets of the n parties which we call an *access structure*. A secret sharing scheme for $\mathcal{A}$ is a method of sharing information with the n parties such that any set of parties in $\mathcal{A}$ can reconstruct the dealer's secret, while any subset of parties not contained in $\mathcal{A}$ cannot reconstruct the dealer's secret.

Beimel [16] showed that the size of the smallest monotone span program tightly characterizes the amount of information required to be shared in *linear* secret sharing schemes, which are a particular subclass of sharing schemes often studied in the literature. Before our results, the best known lower bounds against any linear secret sharing scheme were quasipolynomial.

Corollary III.8. *There is an explicitly defined access structure $\mathcal{A}_{\Delta\text{-GEN}}$ such that any linear secret sharing scheme for*

$\mathcal{A}_{\Delta\text{-GEN}}$ *has information ratio* $2^{\Omega(N^{\varepsilon} \log N)}$ *for* $\varepsilon = 1/14$.

We also obtain a quasipolynomial lower bound for linear secret sharing schemes over the "st-connectivity access structure", and therefore bipartite matching by the known projection reduction from st-connectivity to bipartite matching[2] [37].

Recall from the introduction that a *Nullstellensatz refutation* of a set of polynomial equations $p_1 = 0, \ldots, p_m = 0$ is given by a system of polynomials $q_1, \ldots, q_m$ such that $\sum p_i q_i = 1$. The degree of the refutation is $d = \max_i \deg(q_i p_i)$, and the size is the total number of monomials in all of the polynomials. Using a known connection between Nullstellensatz refutations and span programs discovered by Pudlak and Sgall [4], we can use our rank measure lower bounds to obtain Nullstellensatz degree lower bounds. Due to space limitations, we refer the reader to the full version of the paper for details.

Corollary III.9. *There are unsatisfiable systems of constant-degree polynomial equations* $(P_{\mathrm{STCONN}_{2m^2,2m}}, Q_{\mathrm{STCONN}_{2m^2,2m}})$ *and* $(P_{\Delta_h^{\uparrow 2m^2}\text{-GEN}}, Q_{\Delta_h^{\uparrow 2m^2}\text{-GEN}})$ *in* N *variables such that the following holds. Every real Nullstellensatz refutation of the system* $(P_{\mathrm{STCONN}_{2m^2,2m}}, Q_{\mathrm{STCONN}_{2m^2,2m}})$ *has degree* $\Omega(\log N)$. *Every real Nullstellensatz refutation of the system* $(P_{\Delta_h^{\uparrow 2m^2}\text{-GEN}}, Q_{\Delta_h^{\uparrow 2m^2}\text{-GEN}})$ *requires Nullstellensatz refutations with degree* $\Omega(N^{\varepsilon})$ *for some* $\varepsilon > 0$.

B. Comparator Circuit Lower Bounds

We recall the definition of comparator circuits. A *comparator gate* is the function mapping a pair of input bits $(x, y) \mapsto (x \wedge y, x \vee y)$; it is natural to think of a comparator gate as "sorting" the input (x, y), since the smaller input goes to the first coordinate and the larger input goes to the second. A *comparator circuit* consists of m wires and a sequence $(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)$ of comparator gates, each connecting a pair of wires (in this notation, the $\wedge$ output of the comparator gate is attached to the first wire, and the $\vee$ output of the comparator gate is attaced to the second wire). Each of the m wires is labelled with either a constant $0, 1$, some input variable x or its negation $\overline{x}$ (the circuit is *monotone* if no wire is labelled with the negation of a variable). We will be interested in comparator circuits which compute boolean functions $f : \{0,1\}^n \to \{0,1\}$, where n is possibly less than the number of wires. To do this we designate one of the wires as the output wire and allow the labelling of distinct wires with the same input variable.

If C is a comparator circuit then the *size* of C is the number of wires[3] in C. Let mCC denote the class of languages computable by polynomial-size monotone comparator circuits, and if f is a monotone boolean function let $\mathsf{mCC}(f)$ denote the minimum size of any comparator circuit computing f.

Theorem III.10. *For any matrix* A *and any monotone boolean function* f, $\mu_A(f) \leq \mathsf{mCC}(f)$.

Proof: Use the fact that $\mu_A(\cdot)$ is a *submodular complexity measure* [25]; details are deferred to the full version of the paper. ■

By combining this theorem with Theorems III.3 and III.4 we get a number of new results on mCC. In particular, we separate mCC from mNC^2, mP, and exhibit a monotone function in CC but not mCC.

Corollary III.11. *Let* h *be a positive integer, let* m *be the number of nodes in the height-*h *pyramid graph* Δ_h, *and let* N *be the number of input variables to the function* $\Delta_h^{\uparrow 2m^2}$-GEN. *Let* $\varepsilon = 1/14$. *Then* $\mathsf{mCC}(\Delta_h^{\uparrow 2m^2}\text{-GEN}) \geq 2^{\Omega(N^{\varepsilon} \log N)}$. *Similarly, if* m *is a positive integer, and* N *be the number of input variables to the function* $\mathrm{STCONN}_{2m^2,m}$, *then* $\mathsf{mCC}(\mathrm{STCONN}_{2m^2,m}) \geq N^{\Omega(\log N)}$.

IV. From the Rank Measure to Algebraic Gaps

The rest of the paper is devoted to sketching the proof of Theorem III.1. In this section we show how to reduce the problem of constructing a matrix A witnessing Theorem III.1 to the construction of a boolean function satisfying certain Fourier-analytic properties. Our reduction is general, and naturally phrased using the canonical search problem associated with unsatisfiable CNFs.

Definition IV.1. Let k be a positive integer and let $\mathcal{C} = C_1 \wedge C_2 \wedge \cdots \wedge C_q$ be an unsatisfiable k-CNF on variables $z_1, z_2, \ldots, z_m$. Associated with $\mathcal{C}$ is the following search problem $\mathsf{Search}(\mathcal{C})$: given an assignment $z \in \{0,1\}^m$ to the variables of $\mathcal{C}$, output a falsified clause C_i.

Each clause C has a unique falsifying assignment, and given a boolean function $p : \{0,1\}^m \to \mathbb{R}$ on the same variables as $\mathcal{C}$ we let $p\restriction_C$ denote the restriction of the function p by this assignment.

The reduction roughly proceeds as follows. We start with the search problem $\mathsf{Search}(\mathcal{C})$ and introduce a new algebraic complexity measure on such search problems that we call the *algebraic gap complexity*. We give a generic method to convert the search problem $\mathsf{Search}(\mathcal{C})$ into a monotone boolean function $f_{\mathcal{C}}$ by a variant of the construction introduced by Raz and Mckenzie [18]. The main theorem of this section gives a method to lift lower bounds on the algebraic gap complexity of $\mathsf{Search}(\mathcal{C})$ to lower bounds on the rank measure of the function $f_{\mathcal{C}}$.

We now introduce our new algebraic complexity measure. Recall that $\deg p$ is the size of the largest non-zero Fourier

[2]We thank Ilan Komargodski for pointing us to this result.

[3]It is not hard to show that the number of wires and the number of gates in any comparator circuit without redundant gates are separated by at most a quadratic factor.

coefficient of p.

Definition IV.2. Let $\mathcal{C}$ be an unsatisfiable CNF on m variables. The *algebraic gap complexity* of $\mathsf{Search}(\mathcal{C})$ is the largest integer k for which there is a boolean function $p : \{0,1\}^m \to \mathbb{R}$ such that $\deg p = m$ and $\deg p\restriction_C \leq m-k$ for all clauses C in $\mathcal{C}$.

Next we review *pattern matrices*, which are a major component of our reduction. Let o, m be positive integers, let $n = om$, and let $V(o,m) = [o]^m \times \{0,1\}^m$. Given $x \in [o]^m$ and $y \in \{0,1\}^n$ construct a string $y\restriction_x$ as follows: first, partition $[n]$ into m blocks of size o and write $y = (y_{i,j})$ for $i \in [m]$, $j \in [o]$; then for each $i \in [m]$ set the ith value of $y\restriction_x$ to y_{i,x_i}.

Definition IV.3. Let m, o be positive integers, let $n = om$, and let $p : \{0,1\}^m \to \mathbb{R}$. The *$(m,o,p)$-pattern matrix* is the real matrix A with rows indexed by $y \in \{0,1\}^n$, columns indexed by $(x,w) \in V(o,m)$ such that $A[y,(x,w)] = p(y\restriction_x \oplus w)$.

Observe that pattern matrices convert real-valued boolean functions $p : \{0,1\}^m \to \mathbb{R}$ into matrices. The next lemma follows immediately from Theorem 4.3 in [24].

Lemma IV.4. *Let $p : \{0,1\}^m \to \mathbb{R}$ be given and let A be the (m,o,p)-pattern matrix. The rank of A is*

$$\operatorname{rank} A = \sum_{S : \hat{p}(S) \neq 0} o^{|S|}.$$

The Pattern Matrix Lift. Let m, o be positive integers, let $n = om$, and consider any unsatisfiable d-CNF $\mathcal{C}$ on m variables $z_1, z_2, \ldots, z_m$. We show how to use a pattern matrix to "lift" the search problem $\mathsf{Search}(\mathcal{C})$ to a monotone boolean function $f_{\mathcal{C}}$ such that algebraic gap lower bounds for $\mathsf{Search}(\mathcal{C})$ translate to rank measure lower bounds for $f_{\mathcal{C}}$. Our lifted function is a variant of the transformation first given by Raz and Mckenzie [18] and then further explored by Göös and Pitassi [30].

Definition IV.5. Let d, m, o be positive integers and let $n = om$. Let $\mathcal{C}$ be an unsatisfiable d-CNF on m variables $z_1, z_2, \ldots, z_m$. The *$(\mathcal{C}, o)$-lifted function* $f_{\mathcal{C}}$ is the $\{0,1\}$-valued monotone boolean function defined as follows. The variables of $f_{\mathcal{C}}$ are indexed by pairs $(C,(a,b))$, where C is a clause in $\mathcal{C}$ and $(a,b) \in [o]^{\mathsf{vars}(C)} \times \{0,1\}^{\mathsf{vars}(C)}$. Given a $\{0,1\}$-assignment to the variables of $f_{\mathcal{C}}$, the function outputs 1 if there is an $(x,w) \in V(o,m)$ such that for each clause C, the variable $(C,(x_{\mathsf{vars}(C)}, w_{\mathsf{vars}(C)}))$ is set to 1. Clearly the function is monotone, and observe that $f_{\mathcal{C}}$ has $|\mathcal{C}|(2o)^d$ input variables.

Define the following set of accepting instances $\mathcal{Y}$ and rejecting instances $\mathcal{N}$ of $f_{\mathcal{C}}$. For any $(x,w) \in V(o,m)$ let $Y(x,w)$ be the accepting instance obtained by setting all variables of the form $(C,(x_{\mathsf{vars}(C)}, w_{\mathsf{vars}(C)}))$ to 1 for each clause C in $\mathcal{C}$; let $\mathcal{Y}$ be the set of all such instances. For any $y \in \{0,1\}^n$ let $N(y)$ be the rejecting instance obtained by setting the variable $(C,(x_{\mathsf{vars}(C)}, w_{\mathsf{vars}(C)}))$ to 1 iff the clause C is satisfied when evaluated on the string $y\restriction_{x_{\mathsf{vars}(C)}} \oplus w_{\mathsf{vars}(C)}$ (observe that this is a 0-input of $f_{\mathcal{C}}$ since the formula $\mathcal{C}$ is unsatisfiable); let $\mathcal{N}$ be the set of all such instances.

We study the rank measure $\mu_A(f_{\mathcal{C}})$ of $f_{\mathcal{C}}$ with respect to A. If we consider, for each input variable $(C,(a,b))$, the corresponding rectangle $R \subseteq \mathcal{Y} \times \mathcal{N}$ then in order to bound the rank measure $\mu_A(f_{\mathcal{C}})$ we must analyze the rank of the submatrices $A\restriction_R$ of A corresponding to each variable.

Lemma IV.6. *Let o, m be positive integers and let $n = om$. Let $\mathcal{C}$ be an unsatisfiable d-CNF defined on m variables $z_1, z_2, \ldots, z_m$. Let $p : \{0,1\}^m \to \mathbb{R}$, and let A be the (m,o,p)-pattern matrix. Let $(C,(a,b))$ be any input variable of $f_{\mathcal{C}}$, and let R be the rectangle corresponding to $(C,(a,b))$ in $\mathcal{R}_{f_{\mathcal{C}}}(\mathcal{Y},\mathcal{N})$. Then*

$$\operatorname{rank}(A\restriction_R) = \sum_{S : \widehat{p\restriction_C}(S) \neq 0} o^{|S|}.$$

Proof Sketch: Let d be the arity of the clause C. Let A' denote the $(o, m-d, p\restriction_C)$-pattern matrix. By examining the restrictions of A obtained by restricting each variable of $f_{\mathcal{C}}$, one can see that the matrix $A\restriction_R$ is row equivalent to the matrix consisting of a column of (some number of) copies of A'. This implies $\operatorname{rank} A\restriction_R = \operatorname{rank} A'$. Since A' is the $(o, m-d, p\restriction_C)$-pattern matrix, Lemma IV.4 implies that $\operatorname{rank} A\restriction_R = \operatorname{rank} A' = \sum_{S : \widehat{p\restriction_C}(S) \neq 0} o^{|S|}$, and the lemma follows.

■

Now we have that the rank of the pattern matrix A is related to the degree of p, while the rank of the submatrix $A\restriction_R$ is related to the degree of $p\restriction_C$, where $(C,(a,b))$ is the input variable of $f_{\mathcal{C}}$ corresponding to the rectangle R. It follows that to maximize the rank measure we need a function p which maximizes the difference between these two degrees (thus explaining the definition of the algebraic gap complexity). This is formalized in the next theorem, which is the main result of this section.

Theorem IV.7. *Let m, d be positive integers and let $\mathcal{C}$ be an unsatisfiable d-CNF on m variables. Let k be the algebraic gap complexity of $\mathsf{Search}(\mathcal{C})$, and let $f_{\mathcal{C}}$ be the $(\mathcal{C}, m^2)$-lifted function. There is a matrix A such that $\mu_A(f_{\mathcal{C}}) \geq cm^k$ for some universal constant c.*

Proof: Let $o = m^2$, let $p : \{0,1\}^m \to \mathbb{R}$ be the function witnessing the algebraic gap complexity of $\mathsf{Search}(\mathcal{C})$, and let A be the (m,o,p)-pattern matrix. We lower bound

$$\mu_A(f_{\mathcal{C}}) = \frac{\operatorname{rank} A}{\max\limits_{R \in \mathcal{R}_{f_{\mathcal{C}}}(\mathcal{Y},\mathcal{N})} \operatorname{rank} A\restriction_R}.$$

By Lemma IV.4 $\operatorname{rank} A = \sum_{S:\hat{p}(S)\neq 0} o^{|S|} \geq m^{2m}$ since $\hat{p}([m]) \neq 0$ and $o = m^2$. Let $R \in \mathcal{R}_{f_C}(\mathcal{Y}, \mathcal{N})$ be chosen arbitrarily, let $(C, (a, b))$ be the input variable of f_C corresponding to R. Note that we may assume that $\hat{p}(S) = 0$ for all $S \subseteq [m]$ with $|S| < m - k$ w.l.o.g. since this does not affect the algebraic gap exhibited by p. Since $\deg p\!\upharpoonright_C \leq m - k$, it follows that all of the non-zero Fourier coefficients $\widehat{p\!\upharpoonright_C}(S')$ are obtained as a linear combination of non-zero Fourier coefficients of $\hat{p}(S)$ where $|S| \leq |S'| + d$. Applying Lemma IV.6 and using these two facts, an elementary calculation using properties of binomial coefficients yields $\operatorname{rank} A\!\upharpoonright_R \leq \sum_{i=0}^{d} \binom{m}{k-i} m^{2(m-k-i)} \leq 6m^{2m-k}$. Putting it all together we get

$$\mu_A(f_C) = \frac{\operatorname{rank} A}{\max\limits_{R\in\mathcal{R}_{f_C}(\mathcal{Y},\mathcal{N})} \operatorname{rank} A\!\upharpoonright_R} \geq \frac{m^{2m}}{6m^{2m-k}} \geq cm^k$$

where $c = 1/6$. ■

V. From Algebraic Gaps to Reversible Pebbling

Recall that a DAG G is *good* if it is connected, has maximum in-degree 2, and has a unique sink node t. In this section we complete the proof of Theorem III.1 by using *pebbling contradictions*.

Definition V.1. Let $G = (V, E)$ be any good DAG with sources R and target t. Let Peb_G denote the following unsatisfiable CNF formula. There is one variable z_v for each vertex $v \in V$, and we add the following clauses:

1) The *target clause* $(\neg z_t)$.
2) For each source vertex $u \in R$ add the *source clause* (z_u).
3) For each internal vertex w with in-neighbours $W \subseteq V$ add the *edge clause* $\left(z_w \vee \bigvee_{v\in W} \neg z_v\right)$.

By Theorem IV.7 proved at the end of the previous section, proving lower bounds against the algebraic gap complexity of $\mathsf{Search}(\mathsf{Peb}_G)$ will imply rank measure lower bounds for f_{Peb_G}. The main theorem of this section is that the algebraic gap complexity of $\mathsf{Search}(\mathsf{Peb}_G)$ is at least the reversible pebbling number of G.

Theorem V.2. *For any good DAG G the algebraic gap complexity of* $\mathsf{Search}(\mathsf{Peb}_G)$ *is at least* $\mathsf{rpeb}(G)$.

We defer the proof of Theorem V.2 to the full version of the paper, and use it to prove Theorem III.1.

Proof of Theorem III.1: Let G be any good DAG with m vertices. By Theorem V.2 we have that the algebraic gap complexity of $\mathsf{Search}(\mathsf{Peb}_G)$ is at least $\mathsf{rpeb}(G)$. Let f_{Peb_G} be the (Peb_G, m^2)-lifted function. Theorem IV.7 implies that there is a matrix A such that

$$\mu_A(f_{\mathsf{Peb}_G}) \geq \Omega(m^{\mathsf{rpeb}(G)}).$$

It is not hard to see that f_{Peb_G} is a restriction of $G^{\uparrow 2m^2}$-GEN, and so $\mu_A(G^{\uparrow 2m^2}\text{-GEN}) = \mu_A(f_{\mathsf{Peb}_G}) \geq \Omega(m^{\mathsf{rpeb}(G)})$. ■

Acknowledgements: The authors thank Mika Göös and Aaron Potechin for helpful conversations, and Ilan Komargodski for pointing out the reduction to bipartite matching.

References

[1] M. Karchmer and A. Wigderson, "On span programs," in *Proceedings of the Eigth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, 1993, pp. 102–111. [Online]. Available: http://dx.doi.org/10.1109/SCT.1993.336536

[2] A. Beimel and E. Weinreb, "Separating the power of monotone span programs over different fields," *SIAM J. Comput.*, vol. 34, no. 5, pp. 1196–1215, 2005. [Online]. Available: http://dx.doi.org/10.1137/S0097539704444038

[3] A. A. Razborov, "Applications of matrix methods to the theory of lower bounds in computational complexity," *Combinatorica*, vol. 10, no. 1, pp. 81–93, 1990. [Online]. Available: http://dx.doi.org/10.1007/BF02122698

[4] P. Pudlák and J. Sgall, "Algebraic models of computation and interpolation for algebraic proof systems," in *Proof Complexity and Feasible Arithmetics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, April 21-24, 1996*, 1996, pp. 279–296.

[5] A. Gál, "A characterization of span program size and improved lower bounds for monotone span programs," *Computational Complexity*, vol. 10, no. 4, pp. 277–296, 2001. [Online]. Available: http://dx.doi.org/10.1007/s000370100001

[6] A. Gál and P. Pudlák, "A note on monotone complexity and the rank of matrices," *Inf. Process. Lett.*, vol. 87, no. 6, pp. 321–326, 2003. [Online]. Available: http://dx.doi.org/10.1016/S0020-0190(03)00334-X

[7] L. Babai, A. Gál, and A. Wigderson, "Superpolynomial lower bounds for monotone span programs," *Combinatorica*, vol. 19, no. 3, pp. 301–319, 1999. [Online]. Available: http://dx.doi.org/10.1007/s004930050058

[8] L. Babai, A. Gál, J. Kollár, L. Rónyai, T. Szabó, and A. Wigderson, "Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, 1996, pp. 603–611. [Online]. Available: http://doi.acm.org/10.1145/237814.238010

[9] A. Beimel, A. Gál, and M. Paterson, "Lower bounds for monotone span programs," *Computational Complexity*, vol. 6, no. 1, pp. 29–45, 1997. [Online]. Available: http://dx.doi.org/10.1007/BF01202040

[10] A. Potechin, "Bounds on monotone switching networks for directed connectivity," in *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, 2010, pp. 553–562. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2010.58

[11] M. Karchmer and A. Wigderson, "Monotone circuits for connectivity require super-logarithmic depth," *SIAM J. Discrete Math.*, vol. 3, no. 2, pp. 255–265, 1990. [Online]. Available: http://dx.doi.org/10.1137/0403021

[12] A. A. Razborov, "Lower bounds for the monotone complexity of some boolean functions," *Soviet Math. Dokl.*, vol. 31, pp. 354–357, 1985.

[13] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: http://doi.acm.org/10.1145/359168.359176

[14] A. Beimel, *Coding and Cryptology: Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Secret-Sharing Schemes: A Survey, pp. 11–46. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20901-7_2

[15] A. Wigderson, "$\oplus$ l/ poly $=$ nl / poly," http://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/W94/proc.pdf, accessed: 2016-03-18.

[16] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Technion, 1996.

[17] S. M. Chan and A. Potechin, "Tight bounds for monotone switching networks via fourier analysis," *Theory of Computing*, vol. 10, pp. 389–419, 2014. [Online]. Available: http://dx.doi.org/10.4086/toc.2014.v010a015

[18] R. Raz and P. McKenzie, "Separation of the monotone NC hierarchy," *Combinatorica*, vol. 19, no. 3, pp. 403–435, 1999. [Online]. Available: http://dx.doi.org/10.1007/s004930050062

[19] P. Beame, R. Impagliazzo, J. Krajícek, T. Pitassi, and P. Pudlák, "Lower bound on hilbert's nullstellensatz and propositional proofs," in *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, 1994, pp. 794–806. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1994.365714

[20] D. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1997.

[21] A. Subramanian, "The computational complexity of the circuit value and network stability problems," Ph.D. dissertation, Stanford University, 1990.

[22] S. A. Cook, Y. Filmus, and D. T. M. Le, "The complexity of the comparator circuit value problem," *TOCT*, vol. 6, no. 4, pp. 15:1–15:44, 2014. [Online]. Available: http://doi.acm.org/10.1145/2635822

[23] V. Ramachandran and L. Wang, "Parallel algorithm and complexity results for telephone link simulation," in *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing, SPDP 1991, 2-5 December 1991, Dallas, Texas, USA*, 1991, pp. 378–385. [Online]. Available: http://dx.doi.org/10.1109/SPDP.1991.218216

[24] A. A. Sherstov, "The pattern matrix method," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1969–2000, 2011. [Online]. Available: http://dx.doi.org/10.1137/080733644

[25] A. A. Razborov, "On submodular complexity measures," in *Proceedings of the London Mathematical Society Symposium on Boolean Function Complexity*. New York, NY, USA: Cambridge University Press, 1992, pp. 76–83. [Online]. Available: http://dl.acm.org/citation.cfm?id=167687.167709

[26] B. W. Reichardt, "Span programs are equivalent to quantum query algorithms," *SIAM J. Comput.*, vol. 43, no. 3, pp. 1206–1219, 2014. [Online]. Available: http://dx.doi.org/10.1137/100792640

[27] T. Lee and A. Shraibman, "Disjointness is hard in the multiparty number-on-the-forehead model," *Computational Complexity*, vol. 18, no. 2, pp. 309–336, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00037-009-0276-2

[28] A. Chattopadhyay and A. Ada, "Multiparty communication complexity of disjointness," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 15, no. 002, 2008. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2008/TR08-002/index.html

[29] M. Göös, "Lower bounds for clique vs. independent set," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 1066–1076. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2015.69

[30] M. Göös and T. Pitassi, "Communication lower bounds via critical block sensitivity," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 847–856. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591838

[31] T. Huynh and J. Nordström, "On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity," in *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, 2012, pp. 233–248. [Online]. Available: http://doi.acm.org/10.1145/2213977.2214000

[32] M. Göös, S. Lovett, R. Meka, T. Watson, and D. Zuckerman, "Rectangles are nonnegative juntas," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 2015, pp. 257–266. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746596

[33] J. R. Lee, P. Raghavendra, and D. Steurer, "Lower bounds on the size of semidefinite programming relaxations," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 2015, pp. 567–576. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746599

[34] S. A. Cook, "An observation on time-storage trade off," *J. Comput. Syst. Sci.*, vol. 9, no. 3, pp. 308–316, 1974. [Online]. Available: http://dx.doi.org/10.1016/S0022-0000(74)80046-2

[35] J. R. Gilbert and R. E. Tarjan, "Variations of a pebble game on graphs." Stanford University, Tech. Rep., 1978.

[36] P. W. Dymond and M. Tompa, "Speedups of deterministic machines by synchronous parallel machines," *J. Comput. Syst. Sci.*, vol. 30, no. 2, pp. 149–161, 1985. [Online]. Available: http://dx.doi.org/10.1016/0022-0000(85)90011-X

[37] A. K. Chandra, L. J. Stockmeyer, and U. Vishkin, "Constant depth reducibility," *SIAM J. Comput.*, vol. 13, no. 2, pp. 423–439, 1984. [Online]. Available: http://dx.doi.org/10.1137/0213028