A characterization of span program size and improved lower bounds for monotone span programs

Anna Gál*

Abstract

We give a characterization of span program size by a combinatorial-algebraic measure defined on covers of pairs of 0's and 1's of the function computed. The measure we consider is a generalization of a measure on covers which has been used to prove lower bounds on formula size [K, Ry, Ra], and has also been studied with respect to communication complexity.

In the monotone case our new methods yield $n^{\Omega(\log n)}$ lower bounds for the monotone span program complexity of explicit Boolean functions in n variables over arbitrary fields, improving the previous lower bounds on monotone span program size. Our characterization of span program size implies that any matrix with superpolynomial separation between its rank and cover number can be used to obtain superpolynomial lower bounds on monotone span program size. We also identify a property of bipartite graphs that is sufficient for constructing Boolean functions with large monotone span program complexity.

1 Introduction

The model of span programs was introduced by Karchmer and Wigderson in 1993 [KW].

A span program for a Boolean function is presented as a matrix over some field, with rows labeled by variables or negated variables. The span program accepts an input assignment if and only if a fixed nonzero vector can be obtained as a linear combination of the rows whose labels are satisfied by the input. The size of the span program is the number of rows in the matrix. A span program is monotone if only positive literals are used as labels of the rows, i.e. negated variables are not allowed. More detailed definitions are given in section 2.1.

Span programs are related to several other models of computation. The class of functions computable by polynomial size span programs over $GF(2)$ is equivalent to $\oplus L/poly$, i.e. the class of functions computable by polynomial size parity branching programs [KW, BDHM]. Span programs over other fields are related to other logspace classes [KW, BDHM, ABO].

Monotone span programs are strongly related to the cryptographic problem of secret-sharing. A *secret-sharing scheme* is a cryptographic tool where a dealer shares a secret (from a finite set of possible secrets) among a set of parties such that only the pre-defined authorized subsets of parties are able to reconstruct the secret. The authorized subsets correspond to a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ where n is the number of parties and the authorized subsets are the subsets with their characteristic vectors in $f^{-1}(1)$. (Note that the function f has to be monotone. If a set of parties can reconstruct a secret then every superset of it has to be authorized as well.) Monotone span programs are equivalent to a subclass of secret sharing schemes called *"linear secret sharing schemes"*. See [KW, BGP] for references and more details on secret sharing.

There is also a connection between monotone span programs and certain algebraic proof systems [PS].

One of the main motivations to study span programs is that lower bounds for span program size imply lower bounds for formula size and other interesting complexity measures including branching program size.

So far the largest known lower bound for span program size is $\Omega(n^{3/2}/\log n)$ for the "element distinctness" function [KW, BGP] (n denotes the number of variables). Proving larger than $\Omega(n^3)$ lower bounds for the span program complexity of explicit functions in n variables would improve the largest known lower bound for formula size, which is $\Omega(n^{3-o(1)})$ proved by Hastad [H]. Proving superpolynomial lower bounds for the span program complexity of an explicit

function would imply that the function does not belong to NC^1 or NL.

Lower bounds for the monotone case are important for understanding the model and finding new lower bound methods. They are also of independent interest since they give lower bounds for linear secret-sharing schemes.

Unlike for Boolean circuits it is not known how much monotone span programs are weaker than non-monotone span programs. For Boolean circuits, Razborov's lower bound for the perfect matching function [Ra2] gives a superpolynomial separation between monotone and non-monotone circuits, and a result by É. Tardos [T] shows an exponential gap. No superpolynomial separation is known between monotone and non-monotone span programs.

Monotone span programs can be much more powerful than monotone circuits. A function which is computable by linear size monotone span programs but requires superpolynomial size monotone circuits and exponential size monotone formulae is exhibited in [B+, BGW]. This shows that size and depth lower bound methods for the monotone circuit model (e.g. [Ra1, Ra2, Ra3, An, AB, Ha] for circuit size, [KW1, RW, GH] for circuit depth) cannot be directly applied to monotone span programs.

The following lower bounds are known for the monotone span program complexity of explicit Boolean functions in n variables. Karchmer and Wigderson [KW] proved an $\Omega(n \log n)$ lower bound for the size of monotone span programs over $GF(2)$ computing threshold functions. A lower bound of Csirmaz [Cs] for general secret sharing schemes implies $\Omega(n^2 / \log n)$ lower bounds for the size of monotone span programs over arbitrary fields computing explicit functions. Beimel, Gál and Paterson [BGP] developed a lower bound technique which allows one to prove lower bounds on monotone span program size by a combinatorial criterion on the family of minterms of the function. As a first application of this technique [BGP] proved a $\Omega(n^{5/2})$ lower bound for monotone span programs over arbitrary fields computing the 6-clique function. [B+, BGW] obtained $n^{\Omega(\log n/ \log \log n)}$ lower bounds for monotone span programs over arbitrary fields computing explicit functions based on the combinatorial criterion from [BGP].

In this paper we give a characterization of span program size by a combinatorial-algebraic measure defined on covers of pairs of 0's and 1's of the function computed. The measure we consider is a generalization of a measure on covers which in communication complexity terms can be described as the partition number of the relation defined by all triples (u, v, i) such that for a given Boolean function f, $f(u) = 0$, $f(v) = 1$ and $u_i \neq v_i$ (see section 2.3). This measure has been used to prove lower bounds on formula size by Rychkov [Ry] and Razborov [Ra], and first appeared implicitly in the method of Khrapchenko [K]. We define a generalization of the above measure over arbitrary fields and prove that the new measure is exactly equal to the span program complexity of the Boolean function over the given field.

In the monotone case our new methods yield $n^{\Omega(\log n)}$ lower bounds for the monotone span program complexity of explicit Boolean functions over arbitrary fields, improving the previous lower bounds on monotone span program size. We prove that any matrix whose rank is significantly larger than its cover number can be used to prove lower bounds on monotone span program size. We also identify a property of bipartite graphs that is sufficient for constructing Boolean functions with large monotone span program complexity. Based on this property, we derive $n^{\Omega(\log n)}$ lower bounds on monotone span program size by two different methods: using our new characterization of span program size, and the lower bound criterion from [BGP].

The $n^{\Omega(\log n/ \log \log n)}$ lower bounds in [B+, BGW] are proved for explicit Boolean functions defined by bipartite graphs with certain properties. The functions we work with in this paper are defined by bipartite graphs similarly. However, we are able to make use of a different property of the underlying bipartite graphs, and this gives the improvements in the lower bounds. Noga Alon [A] observed that the three constructions given in [AGHP] can be easily modified to have the same properties sufficient for the $n^{\Omega(\log n/ \log \log n)}$ lower bounds as the Paley-type bipartite graphs used in [B+, BGW], but while the proof for the Paley graph construction is based on the Weil character sum estimates [AGHP, AMN, B+, BGW], the proofs for the other two constructions in [AGHP] are purely combinatorial. In our case, Paley graphs and all the constructions in [AGHP] as well as the construction of [NN] satisfy the sufficient condition for obtaining $n^{\Omega(\log n)}$ lower bounds for explicit Boolean functions defined based on these constructions.

Finally, our lower bounds imply $n^{\Omega(\log n)}$ lower bounds for the size of monotone span programs computing the clique function, deciding whether an input graph on n vertices contains a clique of size $n/2$.

2 Definitions

2.1 Span programs

We describe the formal definition of the model of span programs introduced in [KW].

Let $\mathcal{F}$ be a field. For a matrix M over $\mathcal{F}$ $span(M)$ denotes the linear subspace generated by the rows of M, that is the set of vectors which are linear combinations of the rows of M.

A *span program* over $\mathcal{F}$ is given by a matrix M over $\mathcal{F}$ with its rows labeled by literals $\{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ and a fixed nonzero vector $\vec{t}$. (Sometimes $\vec{t}$ is called the *target vector*.) For an input $a = (a_1, \ldots, a_n) \in \{0, 1\}^n$ let M_a denote the submatrix of M obtained by keeping those rows whose labels are satisfied by a. That is, M_a contains rows labeled by x_i such that $a_i = 1$ and rows labeled by $\bar{x}_i$ such that $a_i = 0$. The span program accepts the input a if the fixed nonzero vector $\vec{t}$ belongs to $span(M_a)$.

A span program *computes* a Boolean function

$f : \{0,1\}^n \to \{0,1\}$ if it accepts exactly those inputs a where $f(a) = 1$.

The *size* of a span program is the number of its rows.

The number of columns is not counted as part of the size. It is always possible to restrict the matrix of a span program to a set of linearly independent columns without changing the function computed by the program, therefore it is not necessary to use more columns than rows. However, in some of our arguments it will be convenient to work with a very large number of columns.

The choice of the fixed nonzero vector $\vec{t}$ does not effect the size of the span program. It is always possible to replace $\vec{t}$ by another nonzero vector $\vec{t}'$ via a change of basis without changing the function computed and the size of the program. Most often $\vec{t}$ is chosen to be the $\bar{1}$ vector (with all entries equal 1).

A span program is called *monotone* if the labels of the rows are only positive literals $\{x_1, \ldots, x_n\}$. Monotone span programs compute only monotone functions, and every monotone Boolean function can be computed by a monotone span program.

We denote by $SP_{\mathcal{F}}(f)$ (respectively $mSP_{\mathcal{F}}(f)$) the size of the smallest span program (respectively monotone span program) over $\mathcal{F}$ that computes f.

2.2 A measure on covers

Our characterization of span program size is based on a generalization of a combinatorial measure which has been used to prove lower bounds on formula size by Rychkov [Ry] and Razborov [Ra], and first appeared implicitly in the method of Khrapchenko [K]. We describe this measure and its relation to formula size.

Let U and V be arbitrary finite sets, such that $U \cap V = \emptyset$. A *rectangle* (or *combinatorial rectangle*) is a set $U_0 \times V_0$, where $U_0 \subseteq U$ and $V_0 \subseteq V$. A *cover* of $U \times V$ is a set $\mathcal{R}$ of rectangles such that every pair $(u, v) \in U \times V$ belongs to at least one rectangle in $\mathcal{R}$. We say that a cover $\mathcal{R}'$ is *embedded* in $\mathcal{R}$ if every rectangle in R' is a subset of some rectangle in $\mathcal{R}$.

For a cover $\mathcal{R}$ of $U \times V$ we consider the following measure

$$\alpha(\mathcal{R}) := min\{|\mathcal{R}'| : \mathcal{R}' \text{ is a disjoint cover embedded in } \mathcal{R}\}.$$

Theorem 2.1 *[Ra] For any cover $\mathcal{R}$ of $U \times V$ and any nonzero matrix A on $U \times V$ over an arbitrary field $\mathcal{F}$ the following holds.*

$$\frac{rk_{\mathcal{F}}(A)}{max_{R \in \mathcal{R}} rk_{\mathcal{F}}(A_R)} \leq \alpha(\mathcal{R}),$$

where A_R is the submatrix of A corresponding to the rectangle R.

Let $U, V \subseteq \{0,1\}^n$ such that $U \cap V = \emptyset$. We consider the following rectangles, for $i = 1, \ldots, n$.

$$R_{0i} := \{(u, v) : u \in U, v \in V, u_i = 0, v_i = 1\},$$

$$R_{1i} := \{(u, v) : u \in U, v \in V, u_i = 1, v_i = 0\}.$$

Since $U \cap V = \emptyset$, every pair (u, v) differs in at least one coordinate, thus the above rectangles form a cover of $U \times V$. This cover is called the *canonical cover* of $U \times V$ and is denoted by $\mathcal{R}_{can}(U, V)$. Let f be a Boolean function $\{0,1\}^n \to \{0,1\}$. We use the notation $\mathcal{R}_{can}(f) = \mathcal{R}_{can}(f^{-1}(0), f^{-1}(1))$.

If f is a monotone Boolean function, then the pairs (u, v) for $u \in U \subseteq f^{-1}(0)$ and $v \in V \subseteq f^{-1}(1)$ must differ in a position such that $u_i = 0$ and $v_i = 1$. Thus in this case the rectangles R_{0i} for $i = 1, \ldots, n$ form a cover of $U \times V$. This cover is denoted by $\mathcal{R}_{mon}(U, V)$ and is called the *monotone canonical cover* of $U \times V$. We also use the notation $\mathcal{R}_{mon}(f) = \mathcal{R}_{mon}(f^{-1}(0), f^{-1}(1))$.

The measure α on the canonical covers of Boolean functions gives lower bounds on formula size. $L(f)$ denotes the formula size of the Boolean function f, and $L_{mon}(f)$ denotes monotone formula size.

Theorem 2.2 *[K, Ry, Ra] Let $U \subseteq f^{-1}(0)$ and $V \subseteq f^{-1}(1)$. Then*

$$L(f) \geq \alpha(\mathcal{R}_{can}(f)) \geq \alpha(\mathcal{R}_{can}(U, V)),$$

and

$$L_{mon}(f) \geq \alpha(\mathcal{R}_{mon}(f)) \geq \alpha(\mathcal{R}_{mon}(U, V)).$$

In the monotone case, Theorem 2.1 can be used to prove lower bounds for $\alpha(\mathcal{R}_{mon}(f))$. Razborov [Ra] proved that in the non-monotone case (when we have to work with $\mathcal{R}_{can}(f)$), for every function f and every matrix A we get

$$\frac{rk_{\mathcal{F}}(A)}{max_{R \in \mathcal{R}_{can}(f)} rk_{\mathcal{F}}(A_R)} = O(n).$$

The following lemma is helpful in constructing functions with large monotone complexity from covers with certain properties.

Lemma 2.3 *[Ra] Any cover of size t for $U \times V$ for arbitrary disjoint sets U and V can be interpreted as the monotone canonical cover for some monotone Boolean function f with t variables and some sets $U' \subseteq f^{-1}(0)$ and $V' \subseteq f^{-1}(1)$.*

2.3 Communication complexity

We briefly describe the communication complexity measures that we refer to in the paper.

Let $f : X \times Y \to \{0,1\}$ be a Boolean function. The two party communication problem introduced by Yao [Y] is to compute $f(x, y)$ by two players, one that knows x and another that knows y. The deterministic communication complexity of this problem is the number of bits the players have to send to each other in order to determine the value of the function on an arbitrary input, using a deterministic protocol. Nondeterministic communication complexity was defined by Lipton and Sedgewick [LS].

The Boolean function $f : X \times Y \to \{0,1\}$ can be represented by a matrix A_f on $X \times Y$ with the value $f(x,y)$ at the (x,y) entry. For a matrix A let $D(A)$ denote the deterministic communication complexity of the corresponding two party communication problem. We denote by $N(A)$ the maximum of the nondeterministic and co-nondeterministic communication complexity of the corresponding problem. A rectangle R is called *monochromatic* for A if each entry in A_R is the same. The *cover number* $C(A)$ of A is the smallest number of monochromatic rectangles needed to cover A (possibly with intersections). With the above notation, $N(A) = \lceil \log_2 C(A) \rceil$ (see [KN]).

Razborov [Ra] showed that any matrix A for which there is a superlinear gap between $D(A)$ and $N(A)$ can be used to obtain superpolynomial lower bounds on monotone formula size (see also in [KN]).

Our characterization of span program size implies that any matrix with superpolynomial separation between its rank and cover number can be used to obtain superpolynomial lower bounds on monotone span program size (see Theorem 4.1).

We note that the measure $\alpha(\mathcal{R}_{can}(f))$ can be described in communication complexity terms as the partition number of the relation defined by all triples (u,v,i) such that $f(u) = 0$, $f(v) = 1$ and $u_i \neq v_i$. The following communication problem was considered in [KW1]. Let f be a Boolean function. Player A gets an input $u \in f^{-1}(0)$, player B gets an input $v \in f^{-1}(1)$ and their goal is to find a coordinate i where $u_i \neq v_i$. One can represent this problem by a $|f^{-1}(0)| \times |f^{-1}(1)|$ matrix with the (u,v) entry containing all indices i such that $u_i \neq v_i$. The partition number of the problem is the minimum number of disjoint monochromatic rectangles covering this matrix. (A rectangle is monochromatic if some index appears in each entry of the rectangle.) It is easy to see that $\alpha(\mathcal{R}_{can}(f))$ is the partition number of the above communication problem for f, and $\alpha(\mathcal{R}_{mon}(f))$ is the partition number of the monotone version of the communication problem.

3 Characterization of span program size

In this section we define a generalization of the measure on covers described in the previous section. We prove that the new measure gives a characterization of span program size over every field.

As before, let U and V be arbitrary finite sets, such that $U \cap V = \emptyset$.

We will consider $|U|$ by $|V|$ matrices of rank 1 over given fields. We say that a set $\mathcal{K}$ of rank 1 matrices is *embedded* in a set $\mathcal{R}$ of combinatorial rectangles, if for every $K \in \mathcal{K}$ all the nonzero entries of K belong to some $R \in \mathcal{R}$.

For a cover $\mathcal{R}$ of $U \times V$ we define the following measure, with respect to a field $\mathcal{F}$.

Definition 3.1 *Let $\mathcal{F}$ be a field.*

$$\alpha_{\mathcal{F}}(\mathcal{R}) := min\{|\mathcal{K}| : \sum_{K \in \mathcal{K}} K \equiv 1\},$$

where $\mathcal{K}$ is a set of rank 1 matrices over $\mathcal{F}$ embedded in $\mathcal{R}$.

For a $|U|$ by $|V|$ matrix Q the notation $Q \equiv 1$ means that $Q(u,v) = 1$ for each pair $u \in U, v \in V$.

We prove the following properties of the above measure.

Lemma 3.2 *For any cover $\mathcal{R}$ of $U \times V$ and any nonzero matrix A on $U \times V$ over a given field $\mathcal{F}$ the following holds.*

$$\frac{rk_{\mathcal{F}}(A)}{\max_{R \in \mathcal{R}} rk_{\mathcal{F}}(A_R)} \leq \alpha_{\mathcal{F}}(\mathcal{R}) \leq \alpha(\mathcal{R}),$$

where A_R is the submatrix of A corresponding to the rectangle R.

Proof: Let $\mathcal{K}$ be a set of $|U|$ by $|V|$ matrices of rank 1 over $\mathcal{F}$ embedded in $\mathcal{R}$ such that $\sum_{K \in \mathcal{K}} K \equiv 1$. Let $A*K$ denote the element-wise product of A and K, that is let $A*K(u,v) = A(u,v)K(u,v)$. Then $A = \sum_{K \in \mathcal{K}} A*K$ holds. The first inequality in the statement of the lemma follows from

$$rk_{\mathcal{F}}(A) \leq \sum_{K \in \mathcal{K}} rk_{\mathcal{F}}(A*K)$$
$$\leq |\mathcal{K}| \max_{K \in \mathcal{K}} rk_{\mathcal{F}}(A*K)$$
$$\leq |\mathcal{K}| \max_{R \in \mathcal{R}} rk_{\mathcal{F}}(A_R),$$

since $\mathcal{K}$ is embedded in $\mathcal{R}$.

To prove the second inequality we observe that for every combinatorial rectangle R the $|U|$ by $|V|$ matrix with value 1 in entries that belong to R and 0 in every other entry is a rank 1 matrix over any field. $\square$

Lemma 3.3 *For any cover $\mathcal{R}$ of $U \times V$*

$$\alpha_{\mathcal{F}}(\mathcal{R}) = min\{\sum_{R \in \mathcal{R}} rk_{\mathcal{F}}(Q_{\subseteq R}) : \sum_{R \in \mathcal{R}} Q_{\subseteq R} \equiv 1\},$$

where $Q_{\subseteq R}$ is an arbitrary $|U|$ by $|V|$ matrix over $\mathcal{F}$ such that all nonzero entries of $Q_{\subseteq R}$ are covered by R.

Proof: Follows from the fact that the rank of a matrix Q is the same as the minimum number of rank 1 matrices whose sum equals Q. $\square$

We prove that the above measure $\alpha_{\mathcal{F}}$ on the canonical cover of a Boolean function is exactly equal to its span program complexity over the given field $\mathcal{F}$. Similarly, the measure on the monotone canonical cover of a Boolean function is exactly equal to its monotone span program complexity.

Theorem 3.4 *For an arbitrary Boolean function f and every field $\mathcal{F}$*

$$SP_{\mathcal{F}}(f) = \alpha_{\mathcal{F}}(\mathcal{R}_{can}(f)).$$

For an arbitrary monotone Boolean function f and every field $\mathcal{F}$

$$mSP_{\mathcal{F}}(f) = \alpha_{\mathcal{F}}(\mathcal{R}_{mon}(f)).$$

Proof:

First we prove that $SP_{\mathcal{F}}(f) \geq \alpha_{\mathcal{F}}(\mathcal{R}_{can}(f))$.

It is proved in [KW] that every span program computing a function f can be transformed into a canonical span program with the same number of rows computing the same function, and this transformation preserves monotonicity. A span program is called *canonical* if its columns are in 1-1 correspondence with the input vectors where the value of the function is 0, and for every $u \in f^{-1}(0)$ the column corresponding to u in M_u is identically 0. In other words, for every $u \in f^{-1}(0)$ there is a column b_u of the canonical span program M computing f with nonzero entries only at rows whose label takes the value 0 on u. By the definition of a span program computing f, for every $v \in f^{-1}(1)$ there is a vector c_v with nonzero entries only at rows whose label takes the value 1 on v, such that $c_v M = \bar{1}$. (We use $\bar{1}$ as the target vector.) Thus a canonical span program gives us sets of vectors $\{c_v : v \in f^{-1}(1)\}$ and $\{b_u : u \in f^{-1}(0)\}$ such that for every $(u, v) \in f^{-1}(0) \times f^{-1}(1)$ we have $c_v b_u = 1$.

We denote by c_v^{0i} and by b_u^{0i} the parts of c_v and b_u respectively, that belong to rows of M labeled by x_i. We denote by c_v^{1i} and by b_u^{1i} the parts of c_v and b_u respectively, that belong to rows of M labeled by $\bar{x}_i$.

Recall that the canonical cover $\mathcal{R}_{can}(f)$ of the Boolean function $f : \{0,1\}^n \to \{0,1\}$ consists of the $2n$ rectangles $R_{\epsilon i}$, for $i = 1, \ldots, n$ and $\epsilon = 0, 1$.

We define $Q_{\epsilon i}(u, v) = b_u^{\epsilon i} c_v^{\epsilon i}$. Then we have $\sum_{\epsilon i} Q_{\epsilon i} \equiv 1$. We observe that all the nonzero entries of $Q_{\epsilon i}$ are covered by $R_{\epsilon i}$ because $Q_{\epsilon i}(u, v) = b_u^{\epsilon i} c_v^{\epsilon i}$ is nonzero only if $b_u^{\epsilon i}$ and $c_v^{\epsilon i}$ have a common nonzero coordinate, and this can only happen if $u_i = \epsilon$ and $v_i = 1 - \epsilon$.

Clearly, the rank of each matrix $Q_{\epsilon i}$ is at most the number of rows in $c_v^{\epsilon i}$ (and $b_u^{\epsilon i}$), thus $SP_{\mathcal{F}}(f) \geq \sum_{\epsilon i} rk_{\mathcal{F}}(Q_{\epsilon i})$. Applying Lemma 3.3, $SP_{\mathcal{F}}(f) \geq \alpha_{\mathcal{F}}(\mathcal{R}_{can}(f))$ follows.

Next we show that given $|f^{-1}(0)|$ by $|f^{-1}(1)|$ matrices $Q_{\epsilon i}$ over $\mathcal{F}$ such that all nonzero entries of $Q_{\epsilon i}$ are covered by $R_{\epsilon i}$ and $\sum_{\epsilon i} Q_{\epsilon i} \equiv 1$ then we can construct a span program computing f of size $\sum_{\epsilon i} rk_{\mathcal{F}}(Q_{\epsilon i})$.

We will construct a canonical span program computing f. In fact, we will construct sets of vectors $\{c_v : v \in f^{-1}(1)\}$ and $\{b_u : u \in f^{-1}(0)\}$ such that for every $(u, v) \in f^{-1}(0) \times f^{-1}(1)$, we have $c_v b_u = 1$. Then we can take the vectors b_u as columns and obtain a canonical span program computing f.

Let $r = rk_{\mathcal{F}}(Q_{\epsilon i})$. Let $\{u_1, \ldots, u_r\}$ be vectors from $f^{-1}(0)$ such that the corresponding rows of $Q_{\epsilon i}$ are linearly independent. For $v \in f^{-1}(1)$ we let $c_v^{\epsilon i}$ consist of the corresponding r coordinates of the column of $Q_{\epsilon i}$ indexed by v. For $j = 1, \ldots, r$ we set $b_{u_j}^{\epsilon i} := e_j^r$, where e_j^r denotes a vector of length r with 1 in the j-th position and 0 in every other position. For every $u \in f^{-1}(0)$ the row of $Q_{\epsilon i}$ indexed by u can be obtained as a linear combination of the rows indexed by $u_1, \ldots, u_r$. We let $b_u^{\epsilon i}$ consist of the corresponding coefficients in this combination. Putting together the pieces $b_u^{\epsilon i}$ and $c_v^{\epsilon i}$ we obtain vectors b_u and c_v with the desired properties.

This proves that $SP_{\mathcal{F}}(f) \leq \alpha_{\mathcal{F}}(\mathcal{R}_{can}(f))$.

The proof for the monotone case works in a similar way.

□

4 Lower bounds for monotone span programs

In this section we prove $n^{\Omega(\log n)}$ lower bounds on the size of monotone span programs computing explicit functions in n variables.

First we prove that any matrix whose rank is significantly larger than its cover number can be used to prove lower bounds on monotone span program size. Next we identify a property of bipartite graphs that is sufficient for constructing Boolean functions with large monotone span program complexity. Based on this property, we derive $n^{\Omega(\log n)}$ lower bounds on monotone span program size by two different methods: using our new characterization of span program size (Theorem 3.4) and by applying the lower bound criterion of [BGP].

4.1 Lower bounds by separating the rank and the cover number

In this section we show using the characterization of span program size given in Theorem 3.4 that any matrix whose rank is significantly larger than its cover number can be used to prove lower bounds on monotone span program size.

Let f be a monotone Boolean function in n variables, $U \subseteq f^{-1}(0)$, $V \subseteq f^{-1}(1)$ and $\mathcal{R} = \mathcal{R}_{mon}(U, V)$. Our characterization of span program size in Theorem 3.4 and Lemma 3.2 suggest the following approach to prove lower bounds on the size of monotone span programs computing f over a given field $\mathcal{F}$. Suppose we could find a matrix A on $U \times V$ such that A_R is monochromatic for each $R \in \mathcal{R}$. Then $rk_{\mathcal{F}}(A_R) = 1$ for each $R \in \mathcal{R}$, and by Lemma 3.2 together with Theorem 3.4 we get $rk_{\mathcal{F}}(A) \leq mSP_{\mathcal{F}}(f)$.

If we want to use this approach and obtain a large lower bound then $rk_{\mathcal{F}}(A)$ must be significantly larger than the cover number $C(A)$, since the number of variables of f is at least $C(A)$. To see that the number of variables of f is at least $C(A)$ recall that $\mathcal{R} = \mathcal{R}_{mon}(U, V)$ is a cover of $U \times V$ by n rectangles. Since we require for our approach that A_R is monochromatic for each $R \in \mathcal{R}$, $\mathcal{R}$ is also a monochromatic cover of A by n rectangles, thus $n \geq C(A)$. Since $rk_{\mathcal{F}}(A) \leq 2^{D(A)}$ [MS] and by definition $N(A) = \lceil \log_2 C(A) \rceil$ this also means that $D(A)$ must be significantly larger than $N(A)$. However, unlike in the case of formula size, a separation between $D(A)$ and $N(A)$ is not sufficient for proving lower bounds on monotone span program size. Nevertheless, our characterization of span program size implies that any matrix A with $rk_{\mathcal{F}}(A)$ significantly larger than $C(A)$ can be used to prove lower bounds on monotone span program size over the field $\mathcal{F}$.

Theorem 4.1 *Given an arbitrary matrix A over a field $\mathcal{F}$ one can define a monotone Boolean function in $C(A)$ variables with monotone span program complexity at least $rk_{\mathcal{F}}(A)$ over $\mathcal{F}$.*

433

Proof: By the definition of $C(A)$ there is a cover $\mathcal{R}$ of A by $\dot{C}(A)$ monochromatic rectangles. Since the rectangles in $\mathcal{R}$ are monochromatic $rk_{\mathcal{F}}(A_R) = 1$ for each $R \in \mathcal{R}$ and any field $\mathcal{F}$. By Lemma 2.3 the cover $\mathcal{R}$ can be interpreted as the monotone canonical cover of some set $U \times V$ such that $U \subseteq f^{-1}(0)$ and $V \subseteq f^{-1}(1)$ for a monotone Boolean function f in $C(A)$ variables. For $U \subseteq f^{-1}(0)$ and $V \subseteq f^{-1}(1)$ we have $\alpha_{\mathcal{F}}(\mathcal{R}_{mon}(f)) \geq \alpha_{\mathcal{F}}(\mathcal{R}_{mon}(U,V))$, thus by Lemma 3.2 and Theorem 3.4 we get $\mathrm{mSP}_{\mathcal{F}}(f) \geq rk_{\mathcal{F}}(A)$. $\square$

The $n^{\Omega(\log n)}$ lower bound on monotone formula size in [Ra] is based on a superpolynomial separation between the rank and the cover number of the following matrix.

Let $Disj_t^n$ denote the $(n, \leq t)$-disjointness matrix, such that the rows and columns of $Disj_t^n$ are indexed by subsets of size *at most t* of an n element set, and $Disj_t^n(x,y) = 1$ if and only if $x \cap y = \emptyset$. Razborov [Ra] proved that for $t = \Theta(\log n)$ we get $C(Disj_t^n) = O(n)$ but $rk_{\mathcal{F}}(Disj_t^n) = n^{\Omega(\log n)}$ for any field $\mathcal{F}$. Razborov [Ra] also showed that one can obtain covers of $Disj_t^n$ for $t = \Theta(\log n)$ by $O(n)$ monochromatic rectangles from random bipartite graphs and Paley graphs can be used to obtain explicit monochromatic covers of size $O(n)$.

Given any explicit monochromatic cover of size $O(n)$ of $Disj_t^n$ for $t = \Theta(\log n)$, we consider the monotone Boolean function provided by Lemma 2.3. Our characterization of span program size in Theorem 3.4 together with Lemma 3.2 immediately implies $n^{\Omega(\log n)}$ lower bounds for the monotone span program complexity of these explicit Boolean functions over arbitrary fields (see the proof of Theorem 4.1).

We describe in more details the functions to which the above lower bounds apply in the next section.

4.2 A sufficient condition for proving lower bounds on monotone span program size

In this section we identify a property of bipartite graphs that is sufficient for constructing functions with large monotone span program complexity.

We will need bipartite graphs on vertex set $V = V_1 \cup V_2$ which satisfy the property that whenever $W_1, W_2 \subseteq V_1$ are disjoint subsets of at most k vertices each, then there is a vertex $v \in V_2$ which is joined to every vertex in W_1 and is not joined to any vertex in W_2. For appropriate values of k, this is a well known property of random graphs (see for example in [Bo]).

Definition 4.2 *We say that a bipartite graph on vertex set* $V = V_1 \cup V_2$ *satisfies the* isolated neighbor *condition for* k *if for arbitrary disjoint subsets* $W_1, W_2 \subseteq V_1$ *of at most* k *vertices each there is a vertex* $v \in V_2$ *which is a common neighbor of all the vertices in* W_1 *and is isolated from all the vertices in* W_2.

A set $T \subseteq \{0,1\}^n$ of vectors is called an (n,k) universal set, if for any subset of k indices $S = \{i_1, \ldots, i_k\}$ the projection of T on the indices in S contains all possible 2^k configurations. Such sets of vectors with as few members as

possible have been studied for example in [KS, SB, AI, NN]. It is clear that bipartite graphs such that the columns of their adjacency matrix form an $(n, 2k)$ universal set of vectors satisfy the isolated neighbor condition for k. On the other hand, the columns of the adjacency matrix of bipartite graphs satisfying the isolated neighbor condition for k form an (n,k) universal set of vectors.

A bipartite Paley graph is defined on vertex sets $V_1 = V_2 = GF(p)$ for p odd prime, and two vertices $x \in V_1$ and $y \in V_2$ are joined by an edge if and only if $x + y$ is a quadratic residue $\bmod p$. The results of [GS, BT] (see also in [Bo]) prove that Paley graphs satisfy the isolated neighbor condition for $k = \Theta(\log n)$, where n is the number of vertices. An analysis for the bipartite case is given for example in [AGHP]. Also, all the constructions of almost k-wise independent random variables in [AGHP, NN] yield bipartite graphs satisfying the condition for $k = \Theta(\log n)$.

We use the following notation. Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$, where $|V_1| = |V_2| = n$. For a set $W \subseteq V_1$ we denote by $\Gamma(W)$ the set of all of its common neighbors in V_2, and by $\hat{\Gamma}(W)$ the set of all the common non-neighbors of W, that is all the vertices in V_2 that are not connected to any vertex in W. For our purposes we let $\Gamma(\emptyset) = \hat{\Gamma}(\emptyset) = V_2$. For a set $T \subseteq V$ we denote by $\bar{T}$ its complement with respect to V, that is $\bar{T} = V \setminus T$. We use the notation $\bar{\mathcal{T}} = \{\bar{T} : T \in \mathcal{T}\}$.

A *minterm* of a monotone Boolean function is a minimal set of its variables such that on any input that assigns 1 to each variable in the set the value of the function must be 1 regardless of the values assigned to the other variables. Similarly, a *maxterm* of a monotone Boolean function is a minimal set of its variables such that on any input that assigns 0 to each variable in the set the value of the function must be 0 regardless of the values assigned to the other variables.

We will construct monotone Boolean functions from bipartite graphs the following way. Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$, where $|V_1| = |V_2| = n$.

Definition 4.3 *Let* $f_{G,t}$ *be the function on* $2n$ *variables, defined such that the set of minterms of* $f_{G,t}$ *consist of all the sets* $W \cup \Gamma(W)$, *where* W *is a subset of size at most* t *of* V_1.

We note that our definition of $f_{G,t}$ gives the duals of the functions considered in [Ra] for given G, and a similar definition of Boolean functions constructed from bipartite graphs has also been used in [B+, BGW].

Let $\mathcal{H}_{G,t} := \{W \cup \Gamma(W) : W \subseteq V_1, |W| \leq t\}$ and $\mathcal{T}_{G,t} := \{W \cup \hat{\Gamma}(W) : W \subseteq V_1, |W| \leq t\}$. With this notation, the set of minterms of $f_{G,t}$ is exactly the set $\mathcal{H}_{G,t}$ by the definition of $f_{G,t}$. The next lemma shows that under certain conditions the members of $\mathcal{T}_{G,t}$ contain maxterms of $f_{G,t}$.

Lemma 4.4 *Let* G *be a bipartite graph on vertex set* $V = V_1 \cup V_2$ *satisfying the isolated neighbor condition for* t. *Then each set in the family* $\mathcal{T}_{G,t}$ *contains a maxterm of the function* $f_{G,t}$. *That is for* $T \in \mathcal{T}_{G,t}$ *and input* X, *if* $T \subseteq \bar{X}$

then $f_{G,t}(X) = 0$ regardless of the value of the other input variables.

Proof: It is enough to show that for every pair $T \in \mathcal{T}_{G,t}$ and $H \in \mathcal{H}_{G,t}$ we have $T \cap H \neq \emptyset$.

Consider arbitrary $T \in \mathcal{T}_{G,t}$ and $H \in \mathcal{H}_{G,t}$, say $T = W_T \cup \hat{\Gamma}(W_T)$ and $H = W_H \cup \Gamma(W_H)$. We need to show that if $W_T \cap W_H = \emptyset$ then $\hat{\Gamma}(W_T) \cap \Gamma(W_H) \neq \emptyset$, which directly follows from the conditions of the lemma, since $|W_T|$ and $|W_H|$ are at most t. $\square$

The following simple lemma is the basis of both of our different proofs of Theorem 4.7 and it was implicitly used also in Razborov's argument to prove $C(Disj_t^n) = O(n)$ [Ra].

Lemma 4.5 *Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$ satisfying the isolated neighbor condition for t. Then for $T = W_T \cup \hat{\Gamma}(W_T) \in \mathcal{T}_{G,t}$ and $H = W_H \cup \Gamma(W_H) \in \mathcal{H}_{G,t}$ we have*

$$W_T \cap W_H = \emptyset \Leftrightarrow \hat{\Gamma}(W_T) \cap \Gamma(W_H) \neq \emptyset.$$

Proof: We have already proved in Lemma 4.4 that if $W_T \cap W_H = \emptyset$ then $\hat{\Gamma}(W_T) \cap \Gamma(W_H) \neq \emptyset$. The other direction follows from the definitions, since $W_H \times \Gamma(W_H)$ corresponds to an all 1 submatrix, and $W_T \times \hat{\Gamma}(W_T)$ corresponds to an all 0 submatrix in the adjacency matrix of the graph G. $\square$

The following lemma is implicit in Razborov's proof of $C(Disj_t^n) = O(n)$ [Ra].

Lemma 4.6 *[Ra] Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$ satisfying the isolated neighbor condition for t. Then $\mathcal{R}_{mon}(\mathcal{T}_{G,t}, \mathcal{H}_{G,t})$ gives a cover of the $(n, \leq t)$-disjointness matrix $Disj_t^n$ by monochromatic rectangles.*

For completeness we include a proof based on the arguments of [Ra].

Proof: First recall that $\mathcal{H}_{G,t} \subseteq f_{G,t}^{-1}(1)$ and $\mathcal{T}_{G,t} \subseteq f_{G,t}^{-1}(0)$, and the $2n$ variables of $f_{G,t}$ correspond to the vertices $v \in V_1 \cup V_2$. Thus $\mathcal{R}_{mon}(\mathcal{T}_{G,t}, \mathcal{H}_{G,t})$ consists of $2n$ rectangles R_v for $v \in V_1 \cup V_2$.

Next note that the sets $H \in \mathcal{H}_{G,t}$ and $\bar{T} \in \mathcal{T}_{G,t}$ are in 1-1 correspondence with the subsets (W_H and W_T respectively) of size at most t of V_1.

A pair $\bar{T} \in \mathcal{T}_{G,t}$ and $H \in \mathcal{H}_{G,t}$ belongs to R_v if and only if $v \in T \cap H$. By Lemma 4.5 each pair $T \in \mathcal{T}_{G,t}$ and $H \in \mathcal{H}_{G,t}$ intersects either in a vertex $v \in V_1$ or in a vertex $v \in V_2$. In the first case the corresponding entry $Disj_t^n(W_T, W_H) = 0$ and in the second case $Disj_t^n(W_T, W_H) = 1$. Thus for $v \in V_1$ the rectangle R_v will be 0-monochromatic and for $v \in V_2$ the rectangle R_v will be 1-monochromatic in $Disj_t^n$. $\square$

Now we are ready to present a sufficient condition for constructing functions with large monotone span program complexity.

Theorem 4.7 *Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$ with $|V_1| = |V_2| = n$ satisfying the isolated neighbor condition for $t = \Theta(\log n)$. Then*

$$mSP_{\mathcal{F}}(f_{G,t}) = n^{\Theta(\log n)}$$

over any field $\mathcal{F}$.

Proof: To prove the lower bound we use our characterization of span program size in Theorem 3.4. First we note that for $U \subseteq f^{-1}(0)$ and $V \subseteq f^{-1}(1)$ we have

$$\alpha_{\mathcal{F}}(\mathcal{R}_{mon}(f)) \geq \alpha_{\mathcal{F}}(\mathcal{R}_{mon}(U, V)),$$

thus

$$mSP_{\mathcal{F}}(f_{G,t}) \geq \alpha_{\mathcal{F}}(\mathcal{R}_{mon}(\mathcal{T}_{G,t}, \mathcal{H}_{G,t})).$$

By Lemma 4.6 we have $rk_{\mathcal{F}}((Disj_t^n)_R) = 1$ for $R \in \mathcal{R}_{mon}(\mathcal{T}_{G,t}, \mathcal{H}_{G,t})$. By Lemma 3.2 this implies that

$$mSP_{\mathcal{F}}(f_{G,t}) \geq rk_{\mathcal{F}}(Disj_t^n) = n^{\Theta(\log n)}.$$

Since the number of minterms of the function is $n^{\Theta(\log n)}$ it is easy to see that the upper bound holds. $\square$

As in [BGW] Theorem 4.7 implies the same lower bound for the size of monotone span programs computing the clique function, since the clique problem is monotone complete for NP [SV, GSi]. Let $CLIQUE_n$ be the function on $m = \binom{n}{2}$ variables taking value 1 if and only if the input graph on n vertices contains a clique of size $n/2$.

Corollary 4.8

$$mSP_{\mathcal{F}}(CLIQUE_n) = n^{\Omega(\log n)}$$

over any field $\mathcal{F}$.

4.3 Critical families of minterms

We show that Theorem 4.7 can also be proved by applying the lower bound condition from [BGP]. First we state the criterion.

Definition 4.9 *[BGP] Let f be a monotone Boolean function and $\mathcal{M}_f$ be the family of all of its minterms. We say that a subfamily $\mathcal{H} \subseteq \mathcal{M}_f$ is a critical family for f, if every $H_0 \in \mathcal{H}$ contains a set $D(H_0) \subseteq H_0$, $|D(H_0)| \geq 2$, such that the following two conditions are satisfied.*

1. The set $D(H_0)$ uniquely determines H_0 in the family. That is, no other set in the family $\mathcal{H}$ contains $D(H_0)$ as a subset.

2. For any subset $Y \subseteq D(H_0)$, the set

$$S_Y = \bigcup_{H \in \mathcal{H}, H \cap Y \neq \emptyset} H \setminus Y$$

does not contain any member of $\mathcal{M}_f$.

It is proved in [BGP] that if $\mathcal{H}$ is a critical family for f then $mSP_{\mathcal{F}}(f) \geq |\mathcal{H}|$ over any field $\mathcal{F}$.

We show the following sufficient condition for obtaining critical families.

Lemma 4.10 *Let G be a bipartite graph on vertex set $V = V_1 \cup V_2$ satisfying the isolated neighbor condition for t. Then the family $\mathcal{H} := \{W \cup \Gamma(W) : W \subseteq V_1, |W| = t\}$, of minterms of the function $f_{G,t}$ is a critical family for $f_{G,t}$.*

Proof: We can take the set W_{H_0} as the subset $D(H_0)$, since we only include into the family $\mathcal{H}$ the minterms defined by sets of cardinality exactly t. We need to show that for every $H_0 \in \mathcal{H}$ and $\emptyset \neq Y \subseteq W_{H_0}$ the set

$$S_Y := \bigcup_{H \in \mathcal{H}, H \cap Y \neq \emptyset} H \setminus Y$$

does not contain any minterms of $f_{G,t}$. This will follow if we show that every such set S_Y is missing a maxterm. More precisely, we show that $\exists T \in T_{G,t}$ such that $T \subseteq \bar{S}_Y$. By Lemma 4.5 if $W_H \cap Y \neq \emptyset$ then $H \cap \hat{\Gamma}(Y) = \emptyset$. Thus $Y \cup \hat{\Gamma}(Y) \subseteq \bar{S}_Y$, which proves the lemma. $\square$

To conclude a proof of Theorem 4.7 by applying the criterion of [BGP] it is enough to note that $|\mathcal{H}| = n^{\Theta(\log n)}$ for $t = \Theta(\log n)$. $\square$

Acknowledgments

References

[A]   N. Alon: Personal communication.

[Al]   N. Alon: Explicit constructions of exponential sized families of k-independent sets. *Discrete Math.*58 (1986), pp. 191-193.

[AB]   N. Alon and R. Boppana: The monotone circuit complexity of Boolean functions. *Combinatorica* 7 (1987), 1-22.

[ABO] E. Allender, R. Beals, M. Ogihara: The complexity of matrix rank and feasible systems of linear equations. In *Proc. 28th ACM STOC*, 1996, pp. 161-167.

[AGHP] N. Alon, O. Goldreich, J. Håstad, R. Peralta: Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms* 3 (1992), 289-304.

[An]   A. Andreev: On a method for obtaining lower bounds for the complexity of individual monotone functions. it Sov. Math. Dokl. 31 (1985), pp. 530-534.

[AMN] Y. Azar, R. Motwani, J. Naor: Approximating probability distributions using small sample spaces. *Combinatorica*, to appear.

[B+]   L. Babai, A. Gál, J. Kollár, L. Rónyai, T. Szabó, A. Wigderson: Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs. In *Proc. 28th ACM STOC*, 1996, pp. 603-611.

[BGW] L. Babai, A. Gál, A. Wigderson: Superpolynomial lower bounds for monotone span programs. DIMACS Technical Report 96-37, 1996. To appear in *Combinatorica*.

[BGP] A. Beimel, A. Gál and M. Paterson: Lower bounds for monotone span programs. In *Proc. 36th IEEE FOCS*, Milwaukee WI 1995, pp. 674-681. Journal version in *Computational Complexity* 6 (1996/97), pp. 29-45.

[Bo]   B. Bollobás: Random graphs. Academic Press, New York, 1985.

[BT]   B. Bollobás and A. Thomason: Graphs which contain all small graphs, *European J. Comb.*2, 1981, pp. 13-15.

[BDHM] G. Buntrock, C. Damm, H. Hertrampf, and C. Meinel: Structure and importance of the logspace-mod class.   *Math. Systems Theory* 25 (1992), pp. 223-237.

[Cs]   L. Csirmaz: The dealer's random bits in perfect secret sharing schemes. Preprint, Mathematical Inst. Hungarian Acad. Sci., 1994.

[GH]   M. Goldmann and J. Håstad: A simple lower bound for monotone clique using a communication game. *Information Processing Letters* 41 (1992), pp. 221-226.

[GSi] M. Grigni, M. Sipser: Monotone Complexity. In *Boolean function complexity*, edited by M. Paterson, London Math. Society Lecture Note Series, 169, Cambridge University Press, 1992, pp. 57-75.

[GS]   R. L. Graham, J. H. Spencer: A constructive solution to a tournament problem. *Canad. Math. Bull.* 14 (1971), 45-48.

[H]   J. Håstad: The shrinkage exponent is 2. In *Proc. 34th IEEE FOCS*, 1993, pp. 114-123.

[Ha]   Armin Haken: Counting bottlenecks to show monotone P≠NP. In *Proc. 36th IEEE FOCS*, Milwaukee WI 1995, pp. 36-40.

[KW1] M. Karchmer and A. Wigderson: Monotone Circuits for Connectivity require Super-Logarithmic Depth. In *SIAM Journal on Discrete Mathematics*, Vol 3, No. 2, 1990, pp. 255-265.

[KW]   M. Karchmer and A. Wigderson: On span programs. In *Proc. 8th Ann. Symp. Structure in Complexity Theory*, IEEE 1993, pp. 102-111.

[K]   V. M. Khrapchenko: Methods of determining lower bounds for the complexity of Π-schemes. *Mat. Zametki*10 (1972) pp. 83-92, (in Russian). English translation in: *Math. Notes Acad. Sciences USSR* 10 (1972) pp. 474-479.

[KS]   D. J. Kleitman and J. Spencer: Families of k-independent sets. *Discrete Math.*6 (1973), pp. 255-262.

[KN]   E. Kushilevitz and N. Nisan: Communication complexity. Cambridge University Press, 1997.

[LS]   R. Lipton and R. Sedgewick: Lower bounds for

VLSI. In *Proc. of 13th ACM STOC*, 1981, pp. 300-307.

[MS] K. Mehlhorn and E. Schmidt: Las Vegas is better than determinism in VLSI and Distributed Computing. In *Proc. of 14th ACM STOC*, (1982), pp. 330-337.

[NN] J. Naor and M. Naor: Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, Vol. 22, No. 4, (1993), pp. 838-856.

[PS] P. Pudlak and J. Sgall: Algebraic models of computation and interpolation for algebraic proof systems. To appear in *Feasible Arithmetic and Lengths of Proofs*, ed. S. Buss, Lecture Notes in Comput. Sci. Springer-Verlag.

[RW] R. Raz and A. Wigderson: Monotone Circuits for Matching require Linear Depth. *Journal of the ACM*, Vol 39, 1992, pp. 736-744.

[Ra] A. A. Razborov: Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica* 10 (1990), pp. 81-93.

[Ra1] A. A. Razborov: Lower bounds for the monotone complexity of some Boolean functions. *Soviet Math. Doklady*, 31 (1985) 354–357.

[Ra2] A. A. Razborov: A lower bound on the monotone network complexity of the logical permanent. *Mat. Zametki* 37 (1985), 887-900.

[Ra3] A. A. Razborov: On the method of approximation. In *Proc. 21st ACM STOC*, 1989, pp. 167–176.

[Ry] K. L. Rychkov: A modification of Krapchenko's method and its applications to bounds on the complexity of Π-schemes and coding functions. *Methody Discretnovo Analyza* 42 (1985), pp. 91-98 (in Russian).

[SB] G. Seroussi and N. Bshouti: Vector sets for exhaustive testing of logic circuits. *IEEE Trans. Inform. Theory*, 34 (1988), pp. 513-522.

[SV] S. Skyum, L.G. Valiant: A complexity theory based on Boolean algebra. JACM, 32(2), 1985, pp. 484–502.

[T] É. Tardos: The gap between monotone and nonmonotone circuit complexity is exponential Combinatorica, 7(4), 1987, pp. 141-142.

[Y] A. Yao: Some complexity questions related to distributed computing. In *Proc. of the 11th ACM STOC*, 1979, pp. 209-213.