



Expanders and time-restricted branching programs

Stasys Jukna

Institute of Mathematics and Computer Science, Akademijos 4, Vilnius, Lithuania

ARTICLE INFO

Article history:

Received 1 August 2005

Received in revised form 27 November 2007

Accepted 5 September 2008

Communicated by A. Razborov

Keywords:

Computational complexity

Branching programs

Lower bounds

Expander graphs

Ramanujan graphs

ABSTRACT

The *replication number* of a branching program is the minimum number R such that along every accepting computation at most R variables are tested more than once; the sets of variables re-tested along different computations may be different. For every branching program, this number lies between 0 (read-once programs) and the total number n of variables (general branching programs). The best results so far were exponential lower bounds on the size of branching programs with $R = o(n/\log n)$. We improve this to $R \leq \epsilon n$ for a constant $\epsilon > 0$. This also gives an alternative and simpler proof of an exponential lower bound for $(1 + \epsilon)n$ time branching programs for a constant $\epsilon > 0$. We prove these lower bounds for quadratic functions of Ramanujan graphs.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Sparse expander graphs, that is, small degree but highly connected graphs, have numerous and often surprising applications in mathematics and computer science; see [6] for a nice survey. In this paper we apply expanders to prove lower bounds on the size of time restricted branching programs.

We consider the standard model of (deterministic) branching programs. Recall that such a program is just a directed acyclic graph with one source node and two sinks, i.e., nodes of out-degree 0. The sinks are labeled by 1 (accept) and by 0 (reject). Each non-sink node has out-degree 2, and the two outgoing edges are labeled by the tests $x_i = 0$ and $x_i = 1$, for some $i \in \{1, \dots, n\}$. Such a program computes a boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ in a natural way: given an input vector $a \in \{0, 1\}^n$, we start in the source node and follow the (unique) path whose tests are consistent with the corresponding bits of a ; this path is the *computation* on a . This way we reach a sink, and the input a is accepted iff this is the 1-sink.

The *length* of a computation is the number of tests along it, and its *replication number* is the number of re-tested bits, i.e., the number of bits tested more than once along the computation; the sets of variables re-tested along different computations may be different.

We are interested in the following parameters of a branching program:

- the *size* S = the number of nodes;
- the *computation time* T = the length of a longest computation, and
- the *replication number* R = the maximum replication number of an accepting computation.

Note that for every branching program in n variables we have $0 \leq R \leq n$. Moreover, every boolean function f in n variables can be computed by a branching program with $T = n$ and $R = 0$: just take a complete binary tree of depth n . However, the size S of such (trivial) branching programs is then exponential for most functions. It is therefore interesting to understand whether S can be substantially reduced by allowing larger values for T and/or R . This is a so-called “space versus time” problem for branching programs.

E-mail address: jukna@thi.informatik.uni-frankfurt.de.

Thus, given a boolean function f in n variables, we are interested in the smallest size S of a branching program computing f when either the computation time T or the replication number R (or both) are limited.

Note that T and R are “semantic” restrictions: they concern only *consistent* paths (computations), i.e. paths that do not contain two contradicting tests $x_i = 0$ and $x_i = 1$ on some bit i . The “syntactic” case, where the restriction is on *all* paths (be they consistent or not), is usually easier to deal with, and exponential lower bounds on the size S in this case were obtained for $T = o(n \log n)$ [13,5,8] as well as for¹ $R = o(n^{1/3} / \log^{2/3} n)$ [16,15].

In the non-syntactic case, the first super-polynomial lower bounds on the size S for $R = o(\sqrt{n} / \log^2 n)$ were proved in [17]. This was improved to $R = o(n / \log^3 n)$ in [14], and further improved to $R = o(n / \log n)$ in [9]. These bounds hold also for $T = (1 + \epsilon)n$ with $\epsilon = o(1 / \log n)$.

The first exponential lower bound on S for $T = (1 + \epsilon)n$ with a (very small but constant!) $\epsilon > 0$ was proved in [3] (the proof works for $\epsilon = 0.0178$). Shortly after, this was substantially improved in [1] to $T = cn$ for an arbitrary constant $c > 0$; see also [4] for some further improvements of this result.

In this paper we use expander graphs to improve the lower bounds of [17,14,9] by giving an exponential lower bounds on the size S when $R \leq \epsilon n$ for a constant $\epsilon > 0$ (Theorem 7). Our argument is entirely different from those used in the previous papers. This also gives a new proof of the lower bound of [3] for $T = (1 + \epsilon)n$ (see Remark 2). Moreover, the amazing simplicity of our proofs (modulo some known deep constructions of expander graphs) indicates that expander graphs could be good candidates to construct hard boolean functions for time-restricted branching programs.

We prove our lower bounds for quadratic forms $f(x) = x^T A x$ over $GF(2)$, where A is an adjacency matrix of particular Ramanujan graphs. Let us note that quadratic forms (over different fields) were used in most papers on time-restricted branching programs: Sylvester and generalized Fourier matrices in [5,3,4], Hankel matrices in [1,4], etc. The “hardness” of the resulting functions was achieved by special *algebraic* properties of the underlying matrices A : every large enough submatrix must have large rank. The difference of our proof is that we use the *combinatorial* properties of the underlying matrices A : they must have relatively few 1’s and still do not have large all-0 submatrices. Such are, in particular, adjacency matrices of good expander graphs, including the Ramanujan graphs. Given any such graph $G = (V, E)$ with $V = \{1, \dots, n\}$, we define a boolean function f_n in n variables by:

$$f_n(x_1, \dots, x_n) = (x_1 \oplus \dots \oplus x_n \oplus 1) \wedge \bigoplus_{\{i,j\} \in E} x_i x_j.$$

That is, given an input vector $a \in \{0, 1\}^n$, we remove all vertices i with $a_i = 0$, and let $f_n(a) = 1$ iff the number of 1’s in a is even and the number of survived edges is odd.

Our main result (Theorem 7) states that there is an absolute constant $\epsilon > 0$ such that any deterministic branching program computing f_n with the replication number $R \leq \epsilon n$ requires size $S = 2^{\Omega(n)}$.

2. A general lower bound

Let n be an even natural number. A subset $A \subseteq \{0, 1\}^n$ of binary vectors is a *combinatorial rectangle*, or just a *rectangle*, if there is a partition of $\{1, \dots, n\}$ into sets S and T of size $|S| = |T| = n/2$ and subsets of vectors $A_1 \subseteq \{0, 1\}^S$ and $A_2 \subseteq \{0, 1\}^T$ such that $A = A_1 \times A_2$. In other words, a set A is a rectangle if its characteristic function $\chi_A(X)$ ($\chi_A(a) = 1$ iff $a \in A$) can be represented as an AND $\chi_A(X) = f_1(X_1) \wedge f_2(X_2)$ of two boolean functions with $X_1 \cap X_2 = \emptyset$ and $|X_1| = |X_2| = n/2$.

We say that a boolean function f in n variables is *rectangle-free* if there is an absolute constant $\delta > 0$ such that $f^{-1}(1)$ contains no rectangle A of size $|A| > 2^{n-\delta n}$. We also say that f is *dense* if it accepts at least $2^{n-o(n)}$ vectors, and *good* if any two accepted vectors differ in at least two bits.

Theorem 1. *Let f be a good and dense boolean function in n variables. If f is rectangle-free, then there is a constant $\epsilon > 0$ such that any deterministic branching program computing f with the replication number $R \leq \epsilon n$ must have size $S = 2^{\Omega(n)}$.*

Remark 2. In any branching program computing a good boolean function in n variables, any accepting computation must test all n bits at least once. This means that for branching programs computing good functions we always have $R \leq T - n$. Hence, Theorem 7 yields exponential lower bounds also for the class of time $(1 + \epsilon)n$ branching programs for a constant $\epsilon > 0$.

We postpone the proof of Theorem 1 to Section 4, and turn to its applications.

3. Explicit lower bounds

To apply Theorem 1 we need explicit dense boolean functions that do not contain large rectangles with respect to *any* balanced partition of their variables. We define such functions as quadratic functions of particular expander graphs.

¹ In the literature, branching programs with the replication number R are also called “(1, +R)-branching programs.”

Let $G = (V, E)$ be an undirected graph on $V = \{1, \dots, n\}$. The quadratic function of G over $GF(2)$ is a boolean function

$$f_G(x_1, \dots, x_n) = \sum_{(i,j) \in E} x_i x_j \pmod 2.$$

Say that a graph is s -mixed if every two disjoint sets of at least s vertices are joined by at least one edge. A graph with n vertices is mixed if it is δn -mixed for some constant $\delta < 1/2$.

Lemma 3. *If G is a mixed graph of constant degree, then its quadratic function f_G is rectangle-free.*

Proof. Fix an arbitrary balanced partition of the vertices of G into two parts, and call an edge *crossing* if it lies between these parts. An *induced matching* is a set of vertex disjoint edges such that the endpoints of any two of these edges are not adjacent in G .

Claim 4. *At least $m = \Omega(n)$ crossing edges of G form an induced matching.*

Proof of Claim 4. We can construct such a matching by repeatedly taking a crossing edge and removing it together with all its neighbors. In each step we remove at most $2d + 1$ vertices, where d is the degree of G . Since the graph is s -mixed and each part of the bipartition has at least $\lfloor n/2 \rfloor$ vertices, the procedure will run for m steps as long as $\lfloor n/2 \rfloor - (2d + 1)m$ is at least s . Since in our case $s = \delta n$ for a constant $\delta < 1/2$ and the degree d is constant, the procedure will run for $m = \Omega(n)$ steps. \square

The partition of the vertices of G corresponds to a partition $X = X_1 \cup X_2$ of the variables of f_G . Let $A \subseteq \{0, 1\}^n$ be an arbitrary rectangle with respect to this partition, and $\chi_A(X) = \chi_1(X_1) \wedge \chi_2(X_2)$ its characteristic function. Suppose that all the vectors of A are accepted by f_G , i.e. $\chi_A(x) \leq f_G(x)$ for all $x \in \{0, 1\}^n$. Our goal is to show that then $|A| \leq 2^{n-\Omega(n)}$.

By Claim 4, some set $M = \{x_1 y_1, \dots, x_m y_m\}$ of $m = \Omega(n)$ crossing edges, with $x_i \in X_1$ and $y_i \in X_2$, forms an induced matching of G . We set to 0 all variables corresponding to vertices outside the matching M . Since M is an induced subgraph of G , the obtained subfunction of f_G is just the inner product function

$$IP_{2m}(x_1, \dots, x_m, y_1, \dots, y_m) = \sum_{i=1}^m x_i y_i \pmod 2.$$

The obtained subfunction $\chi'_A = \chi'_1(x_1, \dots, x_m) \wedge \chi'_2(y_1, \dots, y_m)$ of the characteristic function $\chi_A(X) = \chi_1(X_1) \wedge \chi_2(X_2)$ of the rectangle A is also the characteristic function of some rectangle $B = B_1 \times B_2$ with $B_i \subseteq \{0, 1\}^m$. Since all vectors of A were accepted by f_G , all vectors of B must be accepted by the inner product function IP_{2m} .

The corresponding to IP_{2m} matrix H is an $N \times N$ matrix with $N = 2^m$ rows and columns labeled by vectors $x \in \{0, 1\}^m$ whose entries are defined by $H[x, y] = (-1)^{IP_{2m}(x,y)}$. Since, for every $x \neq 0$, $IP_{2m}(x, y) = 1$ for exactly half of vectors y , this is a Hadamard matrix, that is $H^t H = nI$, where I is the identity matrix. For such matrices we have the following well-known fact (we include its short proof for completeness).

Lemma 5 (Lindsey's Lemma). *The absolute value of the sum of all entries in any $s \times t$ submatrix of an $N \times N$ Hadamard matrix H does not exceed \sqrt{stN} .*

Proof. By the definition of H , the matrix $M = \frac{1}{\sqrt{N}}H$ is unitary: $M^t M = I$. Since such matrices preserve the euclidean norm, for every real vector v , we have $\|Mv\| = \|v\|$, and hence, $\|Hv\| = \sqrt{N}\|v\|$.

Now, if we denote by v_S the characteristic 0-1 vector of $S \subseteq \{1, \dots, n\}$, with $v_S(i) = 1$ iff $i \in S$, then the absolute value of the sum of all entries in an $|S| \times |T|$ submatrix of H is the absolute value of the scalar product of vectors v_S and Hv_T . By the Cauchy–Schwarz inequality, this value does not exceed $\|v_S\| \cdot \|Hv_T\| = \sqrt{N}\|v_S\| \|v_T\| = \sqrt{N|S||T|}$. \square

By Lindsey's Lemma, we have that

$$\left| \sum_{b \in B} (-1)^{IP_{2m}(b)} \right| \leq \sqrt{2^m |B|}.$$

In particular, IP_{2m} can be constant on B only if $|B| \leq 2^m$. Hence, the subfunction χ' of χ_A can accept at most 2^m vectors. Since χ' was obtained from χ_A by setting to 0 at most $n - 2m$ variables, the function χ_A can accept at most $2^m \cdot 2^{n-2m} = 2^{n-m}$ vectors, implying that $|A| \leq 2^{n-m} = 2^{n-\Omega(n)}$.

This completes the proof of Lemma 3. \square

By Lemma 3, the quadratic function f_G of a graph G is rectangle-free, if G has constant degree and is still mixed enough. The following useful bound, observed by many researchers (see, e.g., [2]), says that good expander graphs have this property.

Lemma 6 (Expander Mixing Lemma). *If G is a d -regular graph on n vertices and λ is the second largest eigenvalue of its adjacency matrix, then the number $e(S, T)$ of edges between every two (not necessarily disjoint) subsets S and T of vertices satisfies*

$$\left| e(S, T) - \frac{d|S||T|}{n} \right| \leq \lambda \sqrt{|S||T|}.$$

Proof. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ be the eigenvalues of the adjacency matrix M of G , and let x_1, \dots, x_n be the corresponding orthonormal basis of eigenvectors; here x_1 is $\frac{1}{\sqrt{n}}$ times the all-1 vector $\vec{1}$. Let v_S and v_T be the characteristic vectors of S and T . Expand these two vectors as linear combinations $v_S = \sum_{i=1}^n a_i x_i$ and $v_T = \sum_{i=1}^n b_i x_i$ of the basis vectors. Since the x_i are orthonormal eigenvectors,

$$e(S, T) = v_S^t M v_T = \left(\sum_{i=1}^n a_i x_i \right)^t M \left(\sum_{i=1}^n b_i x_i \right) = \sum_{i=1}^n \lambda_i a_i b_i. \tag{1}$$

Since the graph G is d -regular, we have $\lambda_1 = d$. The first two coefficients a_1 and b_1 are scalar products of $x_1 = \frac{1}{\sqrt{n}} \vec{1}$ with v_S and v_T ; hence, $a_1 = |S|/\sqrt{n}$ and $b_1 = |T|/\sqrt{n}$. Thus, the first term $\lambda_1 a_1 b_1$ in the sum (1) is precisely $\frac{d|S||T|}{n}$. Since $\lambda = \lambda_2$ is the second largest eigenvalue, the absolute value of the sum of the remaining $n - 1$ terms in this sum does not exceed $\lambda \sum_{i=2}^n |a_i b_i|$ which, by Cauchy–Schwarz inequality, does not exceed $\lambda \|\vec{a}\| \|\vec{b}\| = \lambda \|v_S\| \|v_T\| = \lambda \sqrt{|S||T|}$. \square

Hence, a d -regular graph is s -mixed, that is, $e(S, T) > 0$ holds for disjoint sets S and T with $|S| = |T| = s$, if $ds^2/n - \lambda s > 0$, or equivalently if $\lambda < ds/n$. Hence, we need graphs, for which λ is as small as possible.

For this purpose we take *Ramanujan graphs* $RG(n, q)$. These are $(q + 1)$ -regular graphs with the property that $|\lambda| \leq 2\sqrt{q}$. Explicit constructions of Ramanujan graphs on n vertices for every prime $q \equiv 1 \pmod{4}$ and infinitely many values of n were given in [11,10]; these were later extended to the case where q is an arbitrary prime power in [12,7]. According to the Expander Mixing Lemma, Ramanujan graphs $G = RG(n, q)$ are s -mixed for $s = 2n/\sqrt{q}$, and hence, are δn -mixed for a constant $\delta < 1/2$, as long as $q > 16$.

By Lemma 3, the quadratic functions f_G of these graphs are rectangle-free. As a consequence, Theorem 1 implies the following lower bound.

Theorem 7. *Let G be an n -vertex Ramanujan graph of sufficiently large but constant degree. Then there is a constant $\epsilon > 0$ such that any deterministic branching program computing the function $f = f_G \wedge (x_1 \oplus x_2 \oplus \dots \oplus x_n \oplus 1)$ with the replication number $R \leq \epsilon n$ requires size $2^{\Omega(n)}$.*

Proof. We consider the AND with the parity function just to ensure the goodness (accepted vectors must lie at Hamming distance at least two). It is clear that the resulting function f remains rectangle-free. Hence, it is enough to verify that f is dense. This follows from the well-known characterization of the minimum distance of Reed–Muller codes.

Claim 8 (Folklore). *Every nonzero polynomial of degree k in n variables over $GF(2)$ has at least 2^{n-k} nonzero points.*

Proof of Claim 8. In each such polynomial $f(x_1, \dots, x_n)$ we can find a monomial $X_I = \prod_{i \in I} x_i$ with $|I| = k$ such that no monomial X_J with $J \supset I$ is present in f . Hence, after each of 2^{n-k} assignments a of constants to variables x_j with $j \notin I$, we obtain a polynomial f_a in k variables $\{x_i; i \in I\}$ whose all monomials, other than X_I , have degree strictly less than k , implying that $f_a \neq 0$. \square

In our case f is a polynomial of degree at most 3. Moreover, f is nonzero because $f(a) = 1$ for an input vector $a \in \{0, 1\}^n$ with precisely two 1's corresponding to the endpoints of some edge of G . Hence, f accepts at least 2^{n-3} vectors.

This completes the proof of Theorem 7. \square

4. Proof of Theorem 1

Let f be a good and dense boolean function in n variables. Suppose also that the function f is rectangle-free, that is, $f^{-1}(1)$ does not contain a rectangle of size larger than $2^{n-\delta n}$, for some constant $\delta > 0$. Take an arbitrary deterministic branching program computing f with replication number $R \leq \epsilon n$, where $\epsilon > 0$ is a sufficiently small constant to be specified later. Our goal is to prove that then the program must have at least $2^{\Omega(n)}$ nodes.

For an input $a \in \{0, 1\}^n$ accepted by f , let $comp(a)$ denote the (accepting) computation path on a . Since the function f is good, all n bits are tested at least once along each of these paths. Split each of the paths $comp(a)$ into two parts $comp(a) = (p_a, q_a)$, where p_a is an initial segment of $comp(a)$ along which $n/2$ different bits are tested. Hence, the remaining part q_a can test at most $n/2 + R$ different bits.² Looking at segments p_a and q_a as monomials (ANDs of literals), we obtain that f can be written as an OR of ANDs $P \wedge Q$ of DNFs satisfying the following three conditions:

- (i) All monomials have length at least $n/2$ and at most $n/2 + R$. This holds by the choice of segments p_a and q_a .
- (ii) Any two monomials in each DNF are inconsistent, that is, one contains a variable and the other contains its negation. This holds because the program is deterministic: the paths must split before they meet.
- (iii) For all monomials $p \in P$ and $q \in Q$, either $pq = 0$ (the monomials are inconsistent) or $|X(p) \cap X(q)| \leq R$ and $|X(p) \cup X(q)| = n$, where $X(p)$ is the set of variables in a monomial p . This holds because the program has replication number R .

² Note that we count only the number of tests of *different* bits—the total length of (the number of tests along) $comp(a)$ may be much larger than $n + R$.

Fix now one AND $P \wedge Q$ for which the set B of accepted vectors is the largest one; hence, the program must have at least $|f^{-1}(1)|/|B| \geq 2^{n-o(n)}/|B|$ nodes, and it remains to show that the set B cannot be too large, $|B| \leq 2^{n-\Omega(n)}$. We do this by showing that otherwise the set B , and hence, also the set $f^{-1}(1)$, would contain a large rectangle in contradiction with the rectangle-freeness of f .

When doing this we only use the fact that all vectors of B must be accepted by an AND of DNFs satisfying the properties (i)-(iii) above. By (iii) we know that every vector $a \in B$ must be accepted by some pair of monomials $p \in P$ and $q \in Q$ such that $|X(p) \cap X(q)| \leq R$. A (potential) problem, however, is that for different vectors a the corresponding monomials p and q may share *different* variables in common. This may prohibit their combination into a rectangle (see Remark 10). To get rid of this problem, we just fix a set Y of $|Y| \leq R$ variables for which the set $A \subseteq B$ of all vectors in B accepted by pairs of monomials with $X(p) \cap X(q) = Y$ is the largest one. Hence,

$$|A| \geq |B| / \sum_{i=0}^R \binom{n}{i} \geq |B| \cdot 2^{-n \cdot H(\epsilon)},$$

where $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function.

Claim 9. *The set A contains a rectangle C of size*

$$|C| \geq \frac{1}{9} |A|^2 / 2^{n+R}.$$

By the rectangle-freeness of f , we know that $|C| \leq 2^{n-\delta n}$ for a constant $\delta > 0$. Hence, if $R \leq \epsilon n$ for a constant $\epsilon > 0$ satisfying $\epsilon + 2H(\epsilon) < \delta$, then $|B| \leq |A| \cdot 2^{H(\epsilon)n} \leq 2^{n-\Omega(n)}$.

It remains therefore to prove the claim.

Proof of Claim 9. Each monomial of length at most k accepts at least a 2^{-k} fraction of all vectors from $\{0, 1\}^n$. Hence, there can be at most 2^k mutually inconsistent monomials of length at most k . By (i) and (ii), this implies that

$$|P| \leq 2^{n/2} \text{ and } |Q| \leq 2^{n/2+R}. \tag{2}$$

For each monomial $p \in P \cup Q$, let $A_p = \{a \in A : p(a) = 1\}$ be the set of all vectors in A accepted by p ; we call these vectors *extensions* of p . Note that, by the definition of A , $a \in A_p$ iff $pq(a) = 1$ for some monomial $q \in Q$ with $X(p) \cap X(q) = Y$. Since, by (ii), the monomials in P are mutually inconsistent, no two of them can have a common extension. Since every vector from A is an extension of at least one monomial $p \in P$, the sets A_p with $p \in P$ form a partition of A into $|P|$ disjoint blocks. The average size of a block in this partition is $|A|/|P|$. Say that a monomial $p \in P$ is *rich* if the corresponding block A_p contains $|A_p| \geq \frac{1}{3} |A|/|P|$ vectors. Similarly for monomials in Q . By averaging, at least two-thirds of vectors in A must be extensions of rich monomials in P . Since the same holds also for monomials in Q , at least one vector $x \in A$ must be an extension of some rich monomial $p \in P$ and, at the same time, of some rich monomial $q \in Q$.

Let y be the projection of x onto $Y = X(p) \cap X(q)$. Since all variables in Y are tested in both monomials p and q , all the vectors in A_p and in A_q coincide with y on Y . Consider the set of vectors $C = C_1 \times \{y\} \times C_2$, where C_1 is the set of projections of vectors in A_q onto the set of variables $X \setminus X(q)$, and C_2 is the set of projections of A_p onto the set of variables $X \setminus X(p)$. Since both monomials p and q have at least $n/2$ variables, the set C is a rectangle of size

$$|C| = |C_1| \cdot |C_2| = |A_p| \cdot |A_q| \geq \frac{|A|}{3|P|} \cdot \frac{|A|}{3|Q|} \geq \frac{1}{9} \frac{|A|^2}{2^{n/2}} \cdot \frac{|A|}{2^{n/2+R}} = \frac{1}{9} \frac{|A|^2}{2^{n+R}}.$$

Hence, it remains to verify that $C \subseteq A$, i.e., that all vectors $c \in C$ are accepted by $P \wedge Q$.

The vector x belongs to C and has the form $x = (x_1, y, x_2)$ with $x_i \in C_i$. Take now an arbitrary vector $c = (c_1, y, c_2)$ in C . The vector (x_1, y, c_2) belongs to A_p . Hence, there must be a monomial $q' \in Q$ such that $X(p) \cap X(q') = Y$ and pq' accepts this vector. Since all bits of x_1 are tested in p and none of them belongs to Y , none of these bits is tested in q' . Hence, q' must accept also the vector $c = (c_1, y, c_2)$. Similarly, using the fact that (c_1, y, x_2) belongs to A_q , we can conclude that the vector $c = (c_1, y, c_2)$ is accepted by some monomial $p' \in P$. Thus, the vector c is accepted by the monomial $p'q'$, and hence, by $P \wedge Q$.

This completes the proof of the proof of Claim 9, and thus, the proof of Theorem 7. \square

Remark 10. Note that in the last step of the proof it was important that every vector from A is accepted by a pair of monomials shearing the *same* set of variables Y . Would A not have this property, then the rectangle C would not necessarily lie within the set A . Take for example $P = \{x_1, \bar{x}_1\}$ and $Q = \{x_2, x_1 \bar{x}_2\}$ with $p = x_1$ and $q = x_2$. The AND $P \wedge Q$ accepts the set of vectors $A = \{11, 01, 10\}$. The projection of $A_q = \{11, 01\}$ onto $X \setminus X(q) = \{x_1\}$ is $C_1 = \{0, 1\}$, and the projection of $A_p = \{11, 10\}$ onto $X \setminus X(p) = \{x_2\}$ is also $C_2 = \{0, 1\}$. But $C = C_1 \times C_2 \not\subseteq A$, because 00 does not belong to A .

5. Conclusion and an open problem

We have used a new argument to prove exponential lower bounds for *deterministic* branching programs with replication number $R \leq \epsilon n$ for a constant $\epsilon > 0$. Previous arguments could only do this for $R = o(n/\log n)$.

Important in our proof was that the branching program is *deterministic*: this resulted in the property (ii) in the proof of Theorem 7, and hence, into upper bounds (2) on the number of monomials. In *non-deterministic* branching programs (see, e. g., [5]) we do not necessarily have this property, and in this case no exponential lower bounds are known even for $R = 1$.

Even worse, no exponential lower bounds are known for read-once switching-and-rectifying networks. Such a network is just a directed acyclic graph whose edges are labeled by variables and their negations. A network is *read-once* if, along any consistent path from the source to a sink, each variable is tested at most once. Important here is that the restriction is only on *consistent* paths—along paths, containing a variable and its negation, each variable may appear many times. As noted in [9], such networks seem to be the weakest nondeterministic model, for which no nontrivial lower bounds are known.

Acknowledgement

I would like to thank Martin Sauerhoff, Detlef Sieling and the referee for useful comments. Research was supported by the DFG grant SCHN 503/4-1.

References

- [1] M. Ajtai, A non-linear time lower bound for boolean branching programs, *Theory Comput.* 1 (2005) 149–176.
- [2] N. Alon, F.R.K. Chung, Explicit construction of linear sized tolerant networks, *Discrete Math.* 72 (1989) 15–19.
- [3] P. Beame, T.S. Jayram, M. Saks, Time-space tradeoffs for branching programs, *J. Comput. System Sci.* 63 (4) (2001) 542–572.
- [4] P. Beame, M. Saks, X. Sun, E. Vee, Time-space trade-off lower bounds for randomized computation of decision problems, *J. ACM* 50 (2) (2003) 154–195.
- [5] A. Borodin, A. Razborov, R. Smolensky, On lower bounds for read- k times branching programs, *Comput. Complexity* 3 (1993) 1–18.
- [6] S. Hoory, N. Linial, A. Wigderson, Expander graphs and their applications, *Bull. AMS* 43 (4) (2006) 439–561.
- [7] J.W. Jordan, R. Livné, Ramanujan local systems on graphs, *Topology* 36 (5) (1997) 1007–1024.
- [8] S. Jukna, A note on read- k -times branching programs, *Theoret. Inform. Appl.* 29 (1) (1995) 75–83.
- [9] S. Jukna, A. Razborov, Neither reading few bits twice nor reading illegally helps much, *Discrete Appl. Math.* 85 (1998) 223–238.
- [10] A. Lubotzky, R. Phillips, P. Sarnak, Ramanujan graphs, *Combinatorica* 8 (3) (1988) 261–277.
- [11] G.A. Margulis, Explicit constructions of concentrators, *Probl. Peredachi Inf.* 9 (1973) 71–80 (in Russian). Translation: *Problems of Inf. Transm.* (1975) 323–332.
- [12] M. Morgenstern, Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q , *J. Comb. Theory Ser. B* 62 (1) (1994) 44–62.
- [13] E.A. Okolnishnikova, Lower bounds for branching programs computing characteristic functions of binary codes, in: *Metody diskretnogo analiza*, vol. 51, 1991, pp. 61–83. (in Russian).
- [14] P. Savický, S. Žák, A lower bound on branching programs reading some bits twice, *Theoret. Comput. Sci.* 172 (1–2) (1997) 293–301.
- [15] D. Sieling, New lower bounds and hierarchy results for restricted branching programs, *J. Comput. System Sci.* 53 (1) (1996) 79–87.
- [16] D. Sieling, I. Wegener, New lower bounds and hierarchy results for restricted branching programs, in: *Lecture Notes in Comput. Sci.*, vol. 903, Springer, Berlin, 1994, pp. 359–370.
- [17] S. Žák, A superpolynomial lower bound for $(1, +k(n))$ -branching programs, in: *Lecture Notes in Comput. Sci.*, vol. 969, Springer, Berlin, 1995, pp. 319–325.