BRICS

Basic Research in Computer Science

Lower Bounds for Monotone Span Programs

Amos Beimel
Anna Gál
Mike Paterson

BRICS Report Series **RS-94-46**

ISSN 0909-0878 **December 1994**

Copyright © 1994, BRICS, Department of Computer Science University of Aarhus. All rights reserved.

Reproduction of all or part of this work is permitted for educational or research use on condition that this copyright notice is included in any copy.

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@daimi.aau.dk

Lower Bounds for Monotone Span Programs*

Amos Beimel† Anna Gál‡ Mike Paterson§

Abstract

The model of span programs is a linear algebraic model of computation. Lower bounds for span programs imply lower bounds for contact schemes, symmetric branching programs and for formula size. Monotone span programs correspond also to linear secret-sharing schemes. We present a new technique for proving lower bounds for monotone span programs. The main result proved here yields quadratic lower bounds for the size of monotone span programs, improving on the largest previously known bounds for explicit functions. The bound is asymptotically tight for the function corresponding to a class of 4-cliques.

1 Introduction

Karchmer and Wigderson [14] introduced span programs as a linear algebraic model of computation. A span program for a Boolean function is presented as a matrix over some field with rows labeled by literals of the variables, and the size of the program is the number of rows. The span program accepts an assignment if and only if the all-ones row is a linear combination of the rows whose labels are consistent with the assignment. (Definitions are given in Section 2.) The class of functions with polynomial size span programs is equivalent to the class of functions with polynomial size counting branching programs [8], [14]. Span program size is a lower bound on the size of symmetric branching programs [14]. The model of symmetric branching programs is essentially the same as that of (undirected) contact schemes (for definitions, see [14]). Lower bounds for span programs also imply lower bounds for formula size.

Monotone span programs have only positive literals (non-negated variables) as labels of the rows. They compute only monotone functions, even though the computation uses non-monotone linear algebraic operations. It is known that every function with a polynomial size span program is in NC (this follows from [3], [8], [14], [17]), but no monotone analog of

*Most of this work was done while the authors were visiting BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation, Aarhus, Denmark

†Dept. of Computer Science, Technion, Haifa 32000, Israel. Email: beimel@cs.technion.ac.il

‡Dept. of Computer Science, Univ. of Chicago, Chicago, IL 60637, USA. Email: panni@cs.uchicago.edu

§Dept. of Computer Science, Univ. of Warwick, Coventry CV4 7AL, UK. Email: msp@dcs.warwick.ac.uk

this result is known. It is not even known whether every function that has a polynomial size monotone span program also has a polynomial size monotone circuit. The reduction in [14] from symmetric branching programs to span programs preserves monotonicity, and thus lower bounds for monotone span programs imply lower bounds for monotone symmetric branching programs and for monotone formula size.

A different motivation for studying monotone span programs is secret-sharing schemes. A (generalized) *secret-sharing scheme* is a cryptographic tool in which a dealer shares a secret, taken from a finite set of possible secrets, among a set of parties such that only some pre-defined authorized sets of parties can reconstruct the secret. To achieve this goal the dealer distributes private shares to the parties such that any authorized subset of parties can reconstruct the secret from its shares and any non-authorized subset cannot even gain any partial information about the secret (in the information-theoretic sense). The authorized sets are defined by a Boolean function $f : \{0,1\}^m \to \{0,1\}$, where m is the number of parties, such that the authorized sets are the sets with their characteristic vectors in $f^{-1}(1)$.

A secret-sharing scheme can only exist for authorized sets specified by monotone functions: if a subset B can reconstruct the secret then every superset of B can also reconstruct the secret. If the subsets that can reconstruct the secret are all the sets whose cardinality is at least a certain threshold t, then the scheme is called a *threshold* secret-sharing scheme. Threshold secret-sharing schemes were introduced by Blakley [5] and by Shamir [25]. Secret-sharing schemes for general Boolean functions were first defined by Ito, Saito and Nishizeki in [12]. Given any monotone function, they show how to construct a secret sharing scheme in which the authorized sets are the sets specified by the function.

An important issue when designing secret-sharing schemes is the length of shares. For example, even with the more efficient schemes of [2] and when there are only two possible secrets, most functions require shares of length exponential in the number of parties. This means that even in fairly small networks the parties will not have enough memory to store their shares (leaving aside the question of secure storage). The question whether there exist more efficient schemes, or if there exists a Boolean function that does not have (space-) efficient schemes is open. This problem is one of the most important open problems concerning secret-sharing. Some weak lower bounds on the length of the shares were proved in [15, 2, 7, 9, 6, 20, 10]. The best lower bound was proved by Csirmaz [11]. His proof shows that for every m there exists a Boolean function with m variables for which, in every secret sharing scheme, the sum of the lengths of the shares is $\Omega(m^2/\log m)$ times the length of the secret (for every finite set of possible secrets).

Small monotone span programs give rise to efficient linear secret-sharing schemes (see [7, 14, 4]). We call these schemes *linear*, since the shares are a linear combination of the secret and some random inputs. Karchmer and Wigderson [14] proved that if there is a monotone span program of size s for some function then there exists a scheme for the corresponding secret-sharing problem in which the sum of the lengths of the shares of all the parties is s. Therefore, every lower bound on the total size of shares in a secret-sharing scheme is also a lower bound on the size of monotone span programs for the same function. Most

of the known secret-sharing schemes are linear, e.g., those in [25, 21, 12, 2, 26, 27, 7] and all the schemes described in the survey of Stinson [28]. Hence, proving lower bounds for span programs (linear schemes) proves lower bounds for most known schemes. It is also an important step towards proving lower bounds on the length of the shares for all secret-sharing schemes.

The $\Omega(m^2/\log m)$ lower bound implied by [11] for monotone span program size is the strongest previously known lower bound for an explicit function on m variables. In this paper we present a new technique for proving lower bounds for monotone span programs. Our largest lower bound is $\Omega(m^2)$ for an explicit function on m variables. We present several other applications of our technique to explicit functions. Some of our bounds are asymptotically tight, all of them are tightup to constant factors.

We are able to show a linear (exactly m) upper bound for the monotone span program size of a function on m variables, that is known to have $\Omega((m/\log m)^{3/2})$ monotone circuit complexity. This gives some evidence that monotone span programs may be stronger than monotone circuits. Nevertheless, our $\Omega(m^2)$ lower bound for monotone span programs computing the 4-cliques function is larger than the $\Omega((m/\log m)^2)$ lower bound by Razborov's method ([23, 1, 13]) for monotone circuits computing the same function.

The paper is organized as follows. In Section 2 we give the basic definitions, an application of Nečiporuk's method [18] for span programs and a construction of a linear size monotone span program for accepting non-bipartite graphs. The remainder of the paper is devoted to our lower bound method for monotone span programs.

2 Preliminaries

First we state the definition of the model from [14].

Let $\mathcal{F}$ be a field. A *span program* over $\mathcal{F}$ is a labeled matrix $\hat{M}(M,\rho)$ where M is a matrix over $\mathcal{F}$, and ρ is a labeling of the rows of M by literals (variables or negated variables). The *size* of $\hat{M}$ is the number of rows in M.

For every input sequence $\delta \in \{0,1\}^m$ define the submatrix M_δ of M by keeping only those rows r such that $\rho(r) = x_i^\epsilon$ and $\delta_i = \epsilon$, i.e., rows whose labels are set to 1 by the input δ. By definition, $\hat{M}$ *accepts* δ if and only if $\mathbf{1} \in \mathrm{span}(M_\delta)$, i.e., if and only if there is a linear combination of the rows of M_δ giving the vector $\mathbf{1}$. (The row vector $\mathbf{1}$ has the value 1 for each column.)

A span program is called *monotone* if the labels of the rows are only the positive literals $\{x_1,\ldots,x_m\}$. We denote by $\mathrm{SP}_{\mathcal{F}}(f)$ (resp. $\mathrm{mSP}_{\mathcal{F}}(f)$) the size of the smallest span program (resp. monotone span program) over $\mathcal{F}$ that accepts an input δ if and only if $f(\delta) = 1$.

We note that the number of columns does not effect the size of the span program. However, we observe that it is always possible to use no more columns than the size of the program (since we may restrict the matrix to a set of linearly independent columns without changing the function that is computed). Following [14] and with this observation, we can apply Nečiporuk's method [18] to span programs, and get a lower bound of $\Omega(m^{3/2}/\log m)$

for an explicit function with $m = 2n \log n$ variables. Let ED_n be the function which receives n numbers in the range $\{1, \cdots, n^2\}$ and decides whether all the numbers are distinct.

Theorem 1. $\mathrm{SP}_{\mathrm{GF}(2)}(ED_n) = \Omega(m^{3/2}/\log m)$, where $m = 2n \log n$.

Next we present a monotone span program of linear size (exactly m) for a function on m variables, that is known to have $\Omega((m/\log m)^{3/2})$ monotone circuit complexity.

We consider the *Non-Bipartite*$_n$ function, whose input is an undirected graph on n vertices, represented by $m = \binom{n}{2}$ variables, one for each possible edge. The value of the function is 1 if and only if the graph is not bipartite.

Theorem 2. $\mathrm{mSP}_{\mathrm{GF}(2)}(\textit{Non-Bipartite}_n) = m$, where $m = \binom{n}{2}$.

Proof. We construct a monotone span program accepting exactly the non-bipartite graphs as follows. There will be m rows, each labeled by a variable. There is a column for each possible complete bipartite graph on n vertices. The column of a given complete bipartite graph contains the value 0 in each row that corresponds to an edge of the given graph and contains 1 in each row that corresponds to an edge that is missing from the given graph.

This program rejects every bipartite graph G. This is because for every bipartite graph we can find at least one complete bipartite graph that contains it. Therefore, there will be a column that contains only 0's in the rows labeled by the edges of G. This means that we cannot get the vector $\mathbf{1}$ as a linear combination of these rows.

Next we show that the program accepts every non-bipartite graph. Since the span program is monotone, it is sufficient to show that it accepts every *minimal* non-bipartite graph, i.e., every odd cycle. Let C be an arbitrary odd cycle. The intersection of any odd cycle with any bipartite graph has an even number of edges, so C has an odd number of edges which are *not* in any given bipartite graph. Hence the sum of the row vectors corresponding to all the edges in C is odd in each column, i.e., gives the vector $\mathbf{1}$ over GF(2), and so C is accepted by the span program. □

We note that the lower bound by Razborov's method (see [23, 1, 13]) for triangles also applies to the function that accepts exactly the non-bipartite graphs, thus the monotone circuit complexity of the function *Non-Bipartite*$_n$ is $\Omega((n/\log n)^3) = \Omega((m/\log m)^{3/2})$.

A *minterm* of a monotone function is a minimal set of its variables with the property that the value of the function is 1 on any input that assigns 1 to each variable in the set, no matter what the values of the other variables. It is convenient for us to refer to minterms as sets of indices, by simply identifying a set of variables with the set of indices of the corresponding variables.

We denote indices (variables) by lower case letters, and minterms (sets of variables) by upper case letters, e.g., A. Script letters, such as $\mathcal{M}$, will be used for families (sets) of sets, and bold letters for vectors.

3 The General Method for Proving Lower Bounds

The idea of our technique is to show that if the size of a span program (i.e., the number of rows in the matrix) is too small, and the program accepts all the minterms of the function f then it must also accept an input that does not contain a minterm of f, which means that the program does not compute f. Our method may be viewed as an application of the "fusion method" [24, 13, 30].

Let $f : \{0,1\}^m \to \{0,1\}$ be a monotone Boolean function, and $\mathcal{M}$ the family of minterms of f. Let $\mathcal{A} \subseteq \mathcal{M}$ be a subfamily of the minterms, and let $\mathcal{A}_i \subseteq \mathcal{A}$ be all the minterms in $\mathcal{A}$ that contain i.

Let $\mathcal{F}$ be any field, and P a span program over $\mathcal{F}$ that accepts all the minterms from $\mathcal{A}$. Since P accepts all the minterms in $\mathcal{A}$, for every minterm $A \in \mathcal{A}$ there exists at least one linear combination V of the rows labeled by elements of A that equals the vector $\mathbf{1}$. We can consider this combination as a sum of $|A|$ vectors, each taken from the space of the rows labeled by one element of A. That is, for every element $i \in A$ there exists a vector, denoted by $V(A,i)$, in the span of the rows labeled by i, such that $\sum_{i \in A} V(A,i) = \mathbf{1}$. The vector $V(A,i)$ is called the *vector of* i in A.

We next study the consequences when the vectors of i in different minterms are linearly dependent. For the next lemmas we recall that an *affine combination* of vectors is a linear combination of vectors in which the sum (over $\mathcal{F}$) of the coefficients of the vectors in the combination is 1.

Lemma 1. Let the subfamily $\mathcal{A}$ of minterms of f satisfy the following condition.

C0. For each $i \in \{1,\ldots,m\}$, the set $\bigcup_{B \in \mathcal{A}_i} B \setminus \{i\}$ does not contain any minterm of f.

Suppose that $i \in A \in \mathcal{A}$, and that for some monotone span program P that computes f, the vector $V(A,i)$ is a linear combination of vectors of i in other minterms from $\mathcal{A}_i$. Then such a combination must be an affine combination.

Proof. If $V(A,i)$ is a linear combination of vectors of i in other minterms from $\mathcal{A}_i$, then there exist constants $\alpha_1,\ldots,\alpha_\ell$ such that $\sum_{q=1}^{\ell} \alpha_q V(A_q,i) = V(A,i)$, where $A_1,\ldots,A_\ell$ are minterms in $\mathcal{A}_i \setminus \{A\}$. Consider the vector

$$\mathbf{v} = \sum_{q=1}^{\ell} \alpha_q \sum_{k \in A_q} V(A_q,k) - \sum_{k \in A} V(A,k) \ .$$

The contribution of vectors labeled by i to $\mathbf{v}$ is $\sum \alpha_q V(A_q,i) - V(A,i) = \mathbf{0}$. Hence $\mathbf{v}$ is in the span of the set of vectors labeled by elements of $X = \bigcup_{B \in \mathcal{A}_i} B \setminus \{i\}$, and

$$\mathbf{v} = \sum_{q=1}^{\ell} \alpha_q \mathbf{1} - \mathbf{1} = (\sum_{q=1}^{\ell} \alpha_q - 1)\mathbf{1} \ .$$

Since P computes f and, by condition C0, X does not contain a minterm of f, the vector $\mathbf{1}$ cannot be in the span of vectors labeled by X. The conclusion is that $(\sum_{q=1}^{\ell} \alpha_q - 1) = 0$ and the combination is affine. $\square$

Definition 1. A pair (A, i), where $A \in \mathcal{A}$ is a minterm and i is an element of A, is *dangerous for the element* j if $j \in A$ and $V(A, i)$ is an affine combination of the vectors in $\{V(B, i) : B \in \mathcal{A}_i \setminus \mathcal{A}_j\}$.

We say that the pairs (A, i) and (A, j) are *mutually dangerous* if (A, i) is dangerous for j and (A, j) is dangerous for i.

The next lemma is a key ingredient of our method.

Lemma 2. Let f be a monotone Boolean function, $\mathcal{A}$ a subfamily of its minterms, and let P be a monotone span program that computes f. Let $A \in \mathcal{A}$, and let $i, j \in A$ be two elements in A with the property that the set $\bigcup_{B \in \mathcal{A}_i \cup \mathcal{A}_j} B \setminus \{i, j\}$ does not contain any minterm of f. Then the pairs (A, i) and (A, j) cannot be mutually dangerous.

Proof. Assume that the pairs (A, i) and (A, j) are mutually dangerous. This means that there exist constants $\alpha_1, \ldots, \alpha_\ell$, such that $\sum_{q=1}^{\ell} \alpha_q = 1$ and $\sum_{q=1}^{\ell} \alpha_q V(A_q, i) = V(A, i)$, where $A_1, \ldots, A_\ell$ are minterms in $\mathcal{A}_i \setminus \{A\}$ that do not contain j. Similarly, there exist constants $\alpha'_1, \ldots, \alpha'_{\ell'}$, such that $\sum_{q=1}^{\ell'} \alpha'_q = 1$ and $\sum_{q=1}^{\ell'} \alpha'_q V(A'_q, j) = V(A, j)$, where $A'_1, \ldots, A'_{\ell'}$ are minterms in $\mathcal{A}_j \setminus \{A\}$ that do not contain i. Consider the vector

$$\mathbf{v} = \sum_{q=1}^{\ell} \alpha_q \sum_{k \in A_q} V(A_q, k) + \sum_{q=1}^{\ell'} \alpha'_q \sum_{k \in A'_q} V(A'_q, k) - \sum_{k \in A} V(A, k) \ .$$

The minterms $A'_1, \ldots, A'_{\ell'}$ do not contain i, and so the contribution of vectors labeled by i to $\mathbf{v}$ is $\sum \alpha_q V(A_q, i) - V(A, i) = \mathbf{0}$, and similarly the vectors labeled by j do not contribute anything to $\mathbf{v}$. Hence $\mathbf{v}$ is in the span of the set of vectors labeled by elements of $Y = \bigcup_{B \in \mathcal{A}_i \cup \mathcal{A}_j} B \setminus \{i, j\}$. However,

$$\mathbf{v} = \sum_{q=1}^{\ell} \alpha_q \mathbf{1} + \sum_{q=1}^{\ell'} \alpha'_q \mathbf{1} - \mathbf{1} = \mathbf{1} \ .$$

Thus P accepts the set Y that, by a hypothesis of the lemma, does not contain a minterm of f. $\square$

A simple technical lemma is proved now that will be used in the proof of Theorem 3.

Lemma 3. Let S be a set of vectors from a linear space, and let the dimension of S be d. Suppose that no vector in S is an affine combination of the other vectors in S. Then the number of vectors in the set S is at most $d + 1$.

Proof. Let S' be a set of the same cardinality as S, which contains the vectors of S with a new coordinate, coordinate zero, added to each vector and fixed to be 1. The dimension of the set S' can only increase by 1 with respect to the set S, therefore the dimension is at most $d + 1$. Now assume that this set S' is not linearly independent, so there exists a vector in S' which is a linear combination of the rest of the vectors in S'. Since the zero

coordinate in all the vectors is 1, this combination must be affine. Hence the original vector (without the coordinate zero) is also an affine combination of the rest of the vectors, which contradicts the assumption of the lemma. Hence the set S' is linearly independent, and so its cardinality is its dimension, which is at most $d+1$. But $|S'|$ is the same as $|S|$, and the statement of the lemma follows. $\square$

Definition 2. Consider a pair (A, i) where $A \in \mathcal{A}$ is a minterm that contains the element i. We say that this pair is *good* if $V(A, i)$ is *not* an affine combination of the vectors in $\{V(B, i) : B \in \mathcal{A}_i \setminus \{A\}\}$.

Definition 3. Let f be a monotone Boolean function, and $\mathcal{A}$ a subfamily of its minterms. We say that $\mathcal{A}$ is a *critical subfamily*, if for every minterm $A \in \mathcal{A}$ there exist two elements $i, j \in A$ such that the following two conditions are satisfied.

C1. The only minterm in $\mathcal{A}$ that contains both i and j is A, i.e., $\mathcal{A}_i \cap \mathcal{A}_j = \{A\}$.

C2. The set $\bigcup_{B \in \mathcal{A}_i \cup \mathcal{A}_j} B \setminus \{i, j\}$ does not contain any minterm of f.

We are now ready to present two general theorems for proving lower bounds for monotone span programs.

Theorem 3. Let $f : \{0,1\}^m \to \{0,1\}$ be a monotone Boolean function, and $\mathcal{A}$ a critical subfamily of its minterms. Then, for every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(f) \geq |\mathcal{A}| - m \ .$$

Proof. Let $\mathcal{F}$ be any field, and P a monotone span program over $\mathcal{F}$ that computes f. For any $A \in \mathcal{A}$, let i, j be elements of A satisfying conditions C1 and C2. By condition C1, if $V(A, i)$ is an affine combination of the vectors in $\{V(B, i) : B \in \mathcal{A}_i \setminus \{A\}\}$, then it is an affine combination of vectors of i in other minterms from $\mathcal{A}_i$ that do not contain j. This means that if the pair (A, i) is not good then it is dangerous for j, and similarly if the pair (A, j) is not good then it is dangerous for i. Thus by Lemma 2 and C2, every minterm $A \in \mathcal{A}$ contains at least one element i such that the pair (A, i) is good. This shows the following claim.

Claim 1. There are at least $|\mathcal{A}|$ good pairs.

We can now bound the number of good pairs that one element i can belong to. Let d_i denote the dimension of the linear space spanned by the rows labeled by i.

Claim 2. There are at most $d_i + 1$ good pairs that contain the element i.

Proof. Consider the set S of vectors of i in the minterms of good pairs. By the definition of good pairs, the same vector cannot be the vector of i for two good pairs. That is, the number of good pairs to which i belongs is simply the cardinality of S. Furthermore, S satisfies the property that no vector in S is an affine combination of the other vectors in S. Now the claim follows from Lemma 3. □

To complete the proof of Theorem 3 we note that by Claim 2 the number of good pairs is at most $\sum_{i=1}^{m}(d_i+1)$, and by Claim 1 the number of good pairs is at least $|\mathcal{A}|$. Hence, the size of the span program, which by definition is $\sum_{i=1}^{m} d_i$, is at least $|\mathcal{A}|-m$. □

Adding an extra condition to the theorem we get an even simpler proof technique giving slightly stronger bounds.

Theorem 4. Let f be a monotone Boolean function and $\mathcal{A}$ a critical subfamily of its minterms. In addition, let $\mathcal{A}$ satisfy condition C0. Then, for every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(f) \geq |\mathcal{A}| \ .$$

Proof. Let $\mathcal{F}$ be any field, and P a monotone span program over $\mathcal{F}$ that computes f. In addition to the observations made in the proof of Theorem 3 we only need the following claim.

Claim 3. There are at most d_i good pairs that contain the element i.

Proof. Consider the set S of vectors of i in the minterms of good pairs. As in the proof of Claim 2, the number of good pairs to which i belongs is simply the cardinality of S, and S satisfies the property that no vector in S is an affine combination of the other vectors in S. Now the claim follows from Lemma 1. □

From Claim 3 it follows that the total number of good pairs is a lower bound on the size of the span program. From Claim 1, there are at least $|\mathcal{A}|$ good pairs, which concludes the proof. □

We emphasize that condition C2 of Definition 3 requires that the set $\bigcup_{B\in\mathcal{A}_i\cup\mathcal{A}_j} B\setminus\{i,j\}$ does not contain any minterm of f, not just the weaker condition that it does not contain any minterm from $\mathcal{A}$. However it is possible to apply our technique to functions with a subfamily of minterms that satisfies only this weaker condition, if the subfamily may be described by a restriction of the function. This follows from the theorem, proved in [14], that if g is a restriction of f then $\mathrm{SP}_{\mathcal{F}}(g) \leq \mathrm{SP}_{\mathcal{F}}(f)$, and if f is monotone then also $\mathrm{mSP}_{\mathcal{F}}(g) \leq \mathrm{mSP}_{\mathcal{F}}(f)$. (A *restriction* of a function is obtained by fixing the values of some variables and considering the function on the remaining variables.)

4 Applications of the General Method

4.1 Cliques of size three

We consider the $\mathit{Clique}_{3,n}$ function, whose input is an undirected graph on n vertices, represented by $m=\binom{n}{2}$ variables, one for each possible edge. The value of the function is

1 if and only if the graph contains a clique of size three (a triangle).

Corollary 5. For every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(\mathit{Clique}_{3,n}) \geq 2(n/3)^3 - O(n) = \Omega(m^{\frac{3}{2}}) \ .$$

We omit the proof of this corollary. It is similar to the proof of the lower bound for cliques of size four, that we present in detail.

4.2 Cliques of size four

The function $\mathit{Clique}_{4,n}$ is defined similarly to $\mathit{Clique}_{3,n}$.

Corollary 6. For every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(\mathit{Clique}_{4,n}) \geq 3(n/4)^4 - O(n^2) = \Omega(m^2) \ .$$

Proof. We choose a critical subfamily of minterms for $\mathit{Clique}_{4,n}$ as follows. We fix a partition of the vertices into four color classes, each of size $\lfloor n/4 \rfloor$ or $\lceil n/4 \rceil$. Let $\mathcal{K}$ consist of all 4-cliques having one vertex from each class. We refer to the 4-cliques that belong to $\mathcal{K}$ as 4-*colored* cliques.

We show that $\mathcal{K}$ is a critical subfamily. Consider a clique $K \in \mathcal{K}$ with vertices a_1, a_2, a_3, a_4 from color classes $1, 2, 3, 4$ respectively, and two of its non-adjacent edges $e_{12} = \{a_1, a_2\}$ and $e_{34} = \{a_3, a_4\}$. We shall prove that the choice of two non-adjacent edges from each clique in $\mathcal{K}$ satisfies the conditions of Definition 3. Two non-adjacent edges uniquely determine a 4-clique, and so condition C1 is satisfied.

Let $\mathcal{K}_{12}$ be the family of cliques that contain the edge e_{12}, and define $\mathcal{K}_{34}$ similarly. In the next claim we prove that condition C2 is also satisfied.

Claim 4. Let G be the graph with edges $E = \bigcup_{B \in \mathcal{K}_{12} \cup \mathcal{K}_{34}} B \setminus \{e_{12}, e_{34}\}$. Then G does not contain a 4-clique.

Proof. Since $\mathcal{K}$ only contains 4-colored cliques, each edge in E has its endpoints in different color classes. Thus if G contains a 4-clique it has to be a 4-colored clique. The vertices a_1, a_2, a_3, a_4 do not form a clique in G, since the edges e_{12} and e_{34} are missing. Hence, without loss of generality assume some vertex b_1 (where $b_1 \neq a_1$) from color class 1 is in the 4-clique. No edge incident to b_1 is contributed by a clique from $\mathcal{K}_{12}$. Since all the cliques from $\mathcal{K}_{34}$ contain the edge e_{34}, the only neighbour of b_1 in class 3 is a_3, and its only neighbour in class 4 is a_4. But the edge $\{a_3, a_4\}$ is missing. Thus b_1 cannot participate in a 4-clique. $\square$

We have proved that $\mathcal{K}$ is a critical subfamily. It is easy to see that $\mathcal{K}$ satisfies condition C0 as well. Theorem 4 gives a lower bound of $|\mathcal{K}| = (n/4)^4 - O(n^2)$.

We get a stronger bound by observing that any two non-adjacent edges in each $K \in \mathcal{K}$ satisfy the conditions of Definition 3. By Lemma 2, for at least one of any two non-adjacent

edges of a clique, the corresponding clique–edge pair must be good. Thus for every clique $K \in \mathcal{K}$ there are at least three edges such that the pair (K, e) is good. This shows that there are at least $3|\mathcal{K}|$ good pairs. As we have shown in the proof of Theorem 4, the number of good pairs is a lower bound on the size of the span program, and the improved bound follows.

□

Our lower bounds for $Clique_{3,n}$ and $Clique_{4,n}$ are tight up to constant factors.

Let us define $Clique^*_{4,n}$ to be the monotone Boolean function whose set of minterms is $\mathcal{K}$, i.e., the set of 4-colored cliques defined above for a fixed partition of the vertices into approximately equal color classes. We observe that the above lower bound applies to this function as well, and is asymptotically tight in this case.

Corollary 7. Let $n = 4q$. Then, for every field $\mathcal{F}$,

$$3q^4 \leq \mathrm{mSP}_{\mathcal{F}}(Clique^*_{4,n}) \leq 3q^4 + 3q^3 \ .$$

4.3 A function with minterms of size 2

In this subsection we exhibit a function whose minterms are of size 2 with monotone span complexity $\Omega(m^{3/2})$. Let $L_1, ..., L_n$ be n subsets of $\{1, ..., n\}$ such that the intersection of every two subsets is of size at most 1. We will describe a simple construction of such sets later. Define the function f, which has $m = 2n$ variables denoted $\{a_1, a_2, ..., a_n, b_1, ..., b_n\}$, and whose minterms are $\{\{a_i, b_j\} : j \in L_i\}$.

Corollary 8. For every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(f) \geq \sum_{i=1}^{n} |L_i| \ .$$

Proof. We have to prove that f satisfies the conditions of Theorem 4. First we show that the family of all the minterms of f is critical. Condition C1 is obvious since the intersection of any two minterms contains at most one variable. To prove condition C2, we take an arbitrary minterm, say $\{a_1, b_1\}$ without loss of generality, and consider the set $X = \{b_j : j \in L_1\} \cup \{a_i : 1 \in L_i\} \setminus \{a_1, b_1\}$. Suppose that there is some minterm, say $\{a_i, b_j\}$, contained in X. Now $1 \in L_i$ since $a_i \in X$, and $j \in L_1$ since $b_j \in X$. We also have $1 \in L_1$ since $\{a_1, b_1\}$ is a minterm, and $j \in L_i$ since $\{a_i, b_j\}$ is a minterm. However $j \neq 1$, and this contradicts the fact that the size of the intersection of L_1 and L_i is at most 1. It is easy to see that condition C0 is also satisfied and we get the lower bound of $\sum |L_i|$. □

One construction for the sets L_i is to take simply the lines of the projective plane of order q, where q can be a prime power. The number of points and lines is $n = q^2 + q + 1$. Each line has $q + 1$ points, and each point is on $q + 1$ lines. The intersection of every two lines is a single point. We note that this and similar constructions of sets with pairwise

intersections of size at most 1 have other applications in Boolean complexity theory, see for example [19], [22], [29].

We call the function obtained by these sets the *Lines* function. Hence, the lower bound for the function *Lines* follows. We can show also an asymptotically matching upper bound.

Corollary 9. For every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(Lines) = n^{3/2} + O(n) = \Theta(m^{3/2}) \ .$$

5 Limitations of Theorems 3 and 4

We proved tight bounds of $\Theta(m^2)$ for one function and $\Theta(m^{3/2})$ for another function with minterms of size 2. The following theorem proves that these lower bounds are the best achievable from Theorem 3 and Theorem 4.

Theorem 10. Let f be a monotone function and $\mathcal{A}$ a critical subfamily of its minterms. Then $|\mathcal{A}|$ is at most $\binom{m}{2}$. If, in addition, all the minterms in $\mathcal{A}$ are of size 2, then $|\mathcal{A}|$ is at most $\frac{1}{2}(m^{3/2} + m/2)$.

Proof. By our assumptions, each minterm in $\mathcal{A}$ has a pair of elements that does not appear in any other minterm in $\mathcal{A}$. Hence there are at most $\binom{m}{2}$ minterms in $\mathcal{A}$.

Let $\mathcal{A}$ contain only minterms of size 2. If $\mathcal{A}$ contained four minterms of the form $\{i,j\}$, $\{i,j'\}$, $\{i',j\}$, $\{i',j'\}$, then the minterm $\{i',j'\}$ would be contained in $\bigcup_{B\in\mathcal{A}_i\cup\mathcal{A}_j} B \setminus \{i,j\}$, violating condition C2. Now let $S_i = \{a : \{i,a\} \text{ is a minterm of } f\}$, the set of other variables that appear with i in a minterm in $\mathcal{A}$. It follows that for each pair of variables there is at most one i such that S_i contains both elements of the pair. We get

$$\binom{m}{2} \geq \sum_{i=1}^{m} \binom{|S_i|}{2} \geq \frac{1}{2m}\left(\sum_{i=1}^{m} |S_i|\right)^2 - \sum_{i=1}^{m} |S_i|/2$$

by Schwarz's inequality. Hence $\sum |S_i| \leq m^{3/2} + m/2$ for $m > 2$. (We note that this fact is also known from extremal graph theory [16].) Since $\sum |S_i|$ counts every minterm from $\mathcal{A}$ twice, we have $|\mathcal{A}| \leq \frac{1}{2}(m^{3/2} + m/2)$. $\square$

Acknowledgements

We would like to thank Avi Wigderson, Noam Nisan and Teresa Przytycka for helpful discussions. Anna Gál is grateful to László Babai, Lance Fortnow and Ketan Mulmuley for supporting her visit to BRICS.

References

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

[2] J. Benaloh and J. Leichter. Generalized Secret Sharing and Monotone Functions. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88 Proceedings*, *Lecture Notes in Computer Science* 403, (Springer-Verlag 1990) 27–35.

[3] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18 (1984) 147–150.

[4] M. Bertilsson and I. Ingemarsson. A Construction of Practical Secret Sharing Schemes Using Linear Block Codes. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, *Lecture Notes in Computer Science* 718, (Springer-Verlag 1993) 67–79.

[5] G. R. Blakley. Safeguarding Cryptographic Keys. In *Proc. AFIPS 1979 NCC, vol. 48*, (1979) 313–317.

[6] C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro. On the Information Rate of Secret Sharing Schemes. In E. F. Brickell, editor, *Advances in Cryptology - CRYPTO '92 Proceedings*, *Lecture Notes in Computer Science* 740, ((Springer-Verlag 1993) 148–167.

[7] E. F. Brickell and D. M. Davenport. On the Classification of Ideal Secret Sharing Schemes. *Journal of Cryptology*, 4(73)(1991) 123–134.

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

[9] R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro. On the Size of Shares for Secret Sharing Schemes. *Journal of Cryptology*, 6(3)(1993) 157–168.

[10] L. Csirmaz. The Size of a Share Must be Large . In A. De Santis, editor, *Advances in Cryptology – Eurocrypt 94, pre-proceedings*, 1994.

[11] L. Csirmaz. The dealer's random bits in perfect secret sharing schemes. Preprint, 1994.

[12] M. Ito, A. Saito, and T. Nishizeki. Secret Sharing Schemes Realizing General Access Structure. In *Proc. IEEE Global Telecommunication Conf., Globecom 87*, (1987) 99–102.

[13] M. Karchmer. On proving lower bounds for circuit size. In *Proceedings of the 8th Annual Structure in Complexity Theory*, (1993) 112–118.

[14] M. Karchmer and A. Wigderson. On Span Programs. In *Proceedings of the 8th Annual Structure in Complexity Theory*, (1993) 102–111.

[15] E. D. Karnin, J. W. Greene, and M. E. Hellman. On Secret Sharing Systems. *IEEE Transactions on Information Theory*, IT-29,1 (1983) 35–41.

[16] T. Kővári, V. T. Sós and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3 (1954) 50–57.

[17] K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7 (1987) 101–104.

[18] E. I. Nečiporuk. On a Boolean function. *Doklady of the Academy of Sciences of the USSR (in Russian)*, 169(4) (1966) 765–766. English translation in *Soviet Mathematics Doklady*, 7(4) 999–1000.

[19] E. I. Nečiporuk. On a Boolean matrix. *Problemy Kibernet.*, 21 (1969) 237–240. English translation in *Systems Theory Res.*, 21 (1971) 236–239.

[20] J. Kilian and N. Nisan. Private communication, 1990.

[21] S. C. Kothari. Generalized Linear Threshold Scheme. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology - CRYPTO '84 Proceedings*, *Lecture Notes in Computer Science* 196, (Springer-Verlag 1985) 231–241.

[22] N. Pippenger. On Another Boolean Matrix. *Theoretical Computer Science*, 11 (1980) 49–56.

[23] A. A. Razborov. Lower bounds for the monotone complexity of some Boolean functions. *Sov. Math. Dokl.*, 31 (1985) 354–357.

[24] A. A. Razborov. On the method of approximation. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, (1989) 167–176.

[25] A. Shamir. How to Share a Secret. *Communications of the ACM* 22, (1979) 612–613.

[26] G. J. Simmons. How to (Really) Share a Secret. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88 Proceedings*, *Lecture Notes in Computer Science* 403, (Springer-Verlag 1990) 390–448.

[27] G. J. Simmons, W. Jackson, and K. M. Martin. The Geometry of Shared Secret Schemes. *Bulletin of the ICA* 1, (1991) 71–88.

[28] D. R. Stinson. An Explicitation of Secret Sharing Schemes. *Design, Codes and Cryptography* 2, (1992) 357–390.

[29] I. Wegener. The Complexity of Boolean Functions. Wiley-Teubner 1987.

[30] A. Wigderson. The fusion method for lower bounds in circuit complexity. *Bolyai Society Mathematical Studies, Combinatorics, Paul Erdős is Eighty*, (Volume 1) Keszthely (Hungary) (1993) 453–467.

Recent Publications in the BRICS Report Series

RS-94-46 Amos Beimel, Anna Gál, and Mike Paterson. *Lower Bounds for Monotone Span Programs*. December 1994. 14 pp.

RS-94-45 Jørgen H. Andersen, Kåre J. Kristoffersen, Kim G. Larsen, and Jesper Niedermann. *Automatic Synthesis of Real Time Systems*. December 1994. 17 pp.

RS-94-44 Sten Agerholm. *A HOL Basis for Reasoning about Functional Programs*. December 1994. PhD thesis. 233 pp.

RS-94-43 Luca Aceto and Alan Jeffrey. *A Complete Axiomatization of Timed Bisimulation for a Class of Timed Regular Behaviours (Revised Version)*. December 1994. 18 pp. To appear in *Theoretical Computer Science*.

RS-94-42 Dany Breslauer and Leszek Gąsieniec. *Efficient String Matching on Coded Texts*. December 1994. 20 pp.

RS-94-41 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. *On Data Structures and Asymmetric Communication Complexity*. December 1994. 17 pp.

RS-94-40 Luca Aceto and Anna Ingólfsdóttir. *CPO Models for GSOS Languages — Part I: Compact GSOS Languages*. December 1994. 70 pp. An extended abstract of the paper will appear in: *Proceedings of CAAP '95*, LNCS, 1995.

RS-94-39 Ivan Damgård, Oded Goldreich, and Avi Wigderson. *Hashing Functions can Simplify Zero-Knowledge Protocol Design (too)*. November 1994. 18 pp.

RS-94-38 Ivan B. Damgård and Lars Ramkilde Knudsen. *Enhancing the Strength of Conventional Cryptosystems*. November 1994. 12 pp.

RS-94-37 Jaap van Oosten. *Fibrations and Calculi of Fractions*. November 1994. 21 pp.

RS-94-36 Alexander A. Razborov. *On provably disjoint* NP-*pairs*. November 1994. 27 pp.