Lower Bounds for Monotone Span Programs *

Amos Beimel† Anna Gál‡ Mike Paterson§

Abstract

Span programs provide a linear algebraic model of computation. Lower bounds for span programs imply lower bounds for formula size, symmetric branching programs and for contact schemes. Monotone span programs correspond also to linear secret-sharing schemes. We present a new technique for proving lower bounds for monotone span programs. We prove a lower bound of $\Omega(m^{2.5})$ for the 6-clique function. Our results improve on the previously known bounds for explicit functions.

1 Introduction

[22] 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 (see [13] and [22]). Span program size is a lower bound on the size of symmetric branching programs [22]. The model of symmetric branching programs is essentially the same as that of (undirected) contact schemes (for definitions, see [22]). 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 over a finite field is in NC (this follows from [8, 13, 22] and [27]). The

*Part of this work was done while the authors were visiting BRICS, Dept. of Computer Science, Aarhus, Denmark. A preliminary version of this paper was presented at the 36th IEEE Symposium FOCS '95.

†Dept. of Computer Science, Technion, Haifa 32000, Israel. E-mail: beimel@cs.technion.ac.il. URL: http://www.cs.technion.ac.il/~beimel.

‡School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540, USA. E-mail: panni@math.ias.edu. This author was supported in part by the Computer Science Department of the University of Chicago and by NSF Grant DMS-9304580.

§Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK. E-mail: msp@dcs.warwick.ac.uk. This author was supported in part by the EU under ESPRIT contracts 7141 (ALCOM II) and 20244 (ALCOM-IT).

monotone analog of this statement does not hold: [2] exhibit a function that is computable by monotone span programs whose size is linear but requires super-polynomial size monotone circuits. This result shows that we cannot expect to adapt Razborov's lower bound technique for monotone circuits ([31, 32]) to monotone span programs.

We note that if P $\not\subseteq$ non-uniform-NC then there are functions with polynomial monotone circuit complexity that cannot be computed by polynomial size span programs over finite fields. Consider the following language MVAL $= \{(C, w) : C$ is a monotone Boolean circuit that accepts $w\}$. If MVAL has a polynomial size span program over a finite field then it has a polynomial size NC circuit. Since MVAL is P-complete, this would imply that P $\subseteq$ non-uniform-NC. On the other hand, with the proper representation MVAL has polynomial size monotone circuits.

The reduction in [22] 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 gain even partial information about the secret (in the information-theoretic sense). The authorized sets correspond to 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 exactly those with cardinality at least a certain threshold t, then the scheme is called a *threshold* secret-sharing scheme. Threshold secret-sharing schemes were introduced by [10] and [33]. Secret-sharing schemes for general Boolean functions were first defined in [19]. Given any monotone function, they show how to construct a corresponding secret-sharing scheme.

An important issue with secret-sharing schemes is the length of shares. For example, even with the more efficient schemes of [7] or [36] and with only two possible secrets, most functions require shares of length exponential in the number of parties. Hence, 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 of whether there exist more efficient schemes or if there exists a Boolean function with no (space-)efficient scheme is open. This problem is one of the most important open problems concerning secret-sharing. Some lower bounds on the length of the shares were proved in [23, 7, 12, 14, 11, 24] and [17]. The best lower bound was proved by [15, 16]. His proof gives, for every m, a Boolean function with m variables for which the sum of the lengths of the shares in every secret-sharing scheme 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 [12, 22, 9]). We call these schemes *linear*, since the shares are linear combinations of the secret and some random inputs. These schemes are also known as geometric schemes (see [34, 35, 36]; for the equivalence see [20]). [22] 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. On the other hand, lower bounds for monotone span programs imply the same lower bounds for linear secret-sharing schemes (see [3, 4, 18]). Most of the known secret-sharing schemes are linear, e.g., those in [33, 25, 19, 7, 34, 36, 12] and all the schemes described in the survey [37].

The $\Omega(m^2/\log m)$ lower bound implied by [15, 16] for monotone span program size is the strongest previously known lower bound for an explicit function on m variables. In a preliminary version of this paper ([5]), we presented a method that yields quadratic lower bounds for explicit functions, improving on the bound by [16]. The methods presented in [5] and [15, 16] cannot give lower bounds larger than $\Omega(m^2)$.

In this paper we present a new technique for proving lower bounds for monotone span programs, which is a generalization of the method in [5]. We present an $\Omega(m^{2.5})$ lower bound for an explicit function on m variables. We obtain this bound for the function that is defined to have the value 1 if and only if the input graph contains a 6-clique. We present several other applications of our technique to explicit functions. Some of our bounds are asymptotically tight. Our technique allows one to prove lower bounds for monotone span programs by considering a problem in extremal set theory. A recent result [2] demonstrates that our technique can yield super-polynomial lower bounds for monotone span programs. It remains open whether our method could even yield exponential lower bounds.

The paper is organized as follows. In Section 2 we give the basic definitions, an application of Nečiporuk's method ([28]) 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. In Section 3 we present the method, and in Sections 4 and 5 we present applications of the method.

2 Preliminaries

First we state the definition of the model from [22].

Let $\mathcal{F}$ be a field, and $\{x_1, \ldots, x_m\}$ be a set of variables. 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 from $\{x_1, \ldots, x_m, \bar{x}_1, \ldots, \bar{x}_m\}$ (every row is labeled by one literal, and the same literal can label many rows).

A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0,1\}^m$ define the submatrix M_δ of M consisting of those rows whose labels are set to 1 by the input δ, i.e., either rows labeled by some x_i such that $\delta_i = 1$ or rows labeled by some $\bar{x}_i$ such that $\delta_i = 0$. The span program $\hat{M}$ *accepts* δ if and only if $\mathbf{1} \in \text{span}(M_\delta)$, i.e., some linear combination of the rows of M_δ gives the vector $\mathbf{1}$. (The row vector $\mathbf{1}$ has the value 1 in each coordinate.) A span program *computes* a Boolean function f if it accepts exactly those inputs δ where $f(\delta) = 1$.

A span program is called *monotone* if the labels of the rows are only the positive literals $\{x_1, \ldots, x_m\}$. Monotone span programs compute only monotone functions.

The *size* of $\hat{M}$ is the number of rows in M. We denote by $\text{SP}_\mathcal{F}(f)$ (respectively $\text{mSP}_\mathcal{F}(f)$) the size of the smallest span program (respectively monotone span program) over $\mathcal{F}$ that computes f.

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.

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

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 [22] and with this observation, we can apply Nečiporuk's method ([28]) to span programs, and get a lower bound of $\Omega(m^{3/2}/\log m)$ for an explicit function with m variables. ([22] prove the same lower bound for parity branching programs, which does not imply our result.) This is the best lower bound known for the non-monotone span program complexity of an explicit function. Let ED_n be the "element distinctness" function which receives n numbers in the range $\{1, \cdots, n^2\}$ and decides whether all the numbers are distinct. The function ED_n has $2n \log n$ Boolean variables.

Theorem 1. $\text{SP}_{\text{GF}(2)}(ED_n) = \Omega(m^{3/2}/\log m)$, 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 $\Omega(m^{3/2}/(\log m)^3)$ monotone circuit complexity ([31, 1, 21]). We consider the function *Non-Bipartite*$_n$, whose input is an undirected graph on n vertices, represented by $\binom{n}{2}$ variables, one for each possible edge. The value of the function is 1 if and only if the graph is not bipartite.

Theorem 2. $\text{mSP}_{\text{GF}(2)}(\textit{Non-Bipartite}_n) = m$, 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 for 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 every other row.

This program rejects every bipartite graph G. This is because G is contained in some complete bipartite graph, and so there will be a column that contains only 0's in the rows labeled by the edges of G. Therefore the vector $\mathbf{1}$ is not 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 cycle with any complete bipartite graph has an even number of edges. So the odd cycle C has an odd number of edges which are *not* in any given complete 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 [31, 1, 21]) for triangles also applies to the function that accepts exactly the non-bipartite graphs, thus the monotone circuit complexity of the function $\textit{Non-Bipartite}_n$ is $\Omega((n/\log n)^3) = \Omega(m^{3/2}/(\log m)^3)$.

3 The 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 approach may be viewed as an application of the "fusion method" of [32, 21] and [39].

We introduce the definition of a critical family of minterms of a monotone Boolean function. We prove that the cardinality of a critical family for a function f is a lower bound on the size of monotone span programs computing f.

Definition 1. Let f be a monotone Boolean function and $\mathcal{M}_f$ be the family of all of its minterms. Let $\mathcal{H} \subseteq \mathcal{M}_f$ be a subfamily of the minterms of f. We say that $\mathcal{H}$ is a *critical family for* f, if every $H \in \mathcal{H}$ contains a set $T_H \subseteq H$, $|T_H| \geq 2$, such that the following two conditions are satisfied.

C1. The set T_H uniquely determines H in the family $\mathcal{H}$. That is, no other set in the family $\mathcal{H}$ contains T_H.

C2. For any subset $Y \subseteq T_H$, the set $S_Y = \bigcup_{A \in \mathcal{H}, A \cap Y \neq \emptyset} A \setminus Y$ does not contain any member of $\mathcal{M}_f$.

Note that Condition C2 requires that S_Y contains no minterm from f, not just none from $\mathcal{H}$.

Observation 1. If $\mathcal{H}$ is a critical family and $|T_H| = t$ for each $H \in \mathcal{H}$, then $|\mathcal{H}| \leq \binom{m}{t}$.

Theorem 3. Let f be a monotone Boolean function, and let $\mathcal{H}$ be a critical subfamily of minterms for f. Then for every field $\mathcal{F}$,

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

Proof. Let M be the matrix of a monotone span program computing f, and let r be the number of rows of M. Any minterm of $\mathcal{H}$ is accepted by the program. By definition, this means that, for every $H \in \mathcal{H}$, there is some vector $\mathbf{c}_H \in \mathcal{F}^r$ such that $\mathbf{c}_H \cdot M = \mathbf{1}$, and where $\mathbf{c}_H$ has nonzero coordinates only at rows labeled by variables from H. For any given H there may be several such vectors, we pick one of them and denote it by $\mathbf{c}_H$.

Since $\mathbf{c}_H$ is taken from $\mathcal{F}^r$, the number of linearly independent vectors among the vectors $\mathbf{c}_H$ for $H \in \mathcal{H}$ is a lower bound for r, i.e., for the size of the span program computing f. We show that all the vectors $\mathbf{c}_H$ for $H \in \mathcal{H}$ must be linearly independent.

Suppose that this is not the case, i.e., for some $H \in \mathcal{H}$,

$$\mathbf{c}_H = \sum_{A \in \mathcal{A}} \alpha_A \mathbf{c}_A \ , \tag{1}$$

where $\alpha_A \in \mathcal{F}$ and $\mathcal{A} = \mathcal{H} \setminus \{H\}$.

Let us consider the set $T_H \subseteq H$ from Definition 1.

Lemma 1. If Equation (1) holds, then for any nonempty subset $Y \subseteq T_H$ the following must hold.

$$\sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \ = 1 \ .$$

Proof. Suppose that for some $Y \subseteq T_H$, $\sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \ = \gamma \neq 1$.

Let us consider the vector

$$\mathbf{c} = \sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \mathbf{c}_A \ - \mathbf{c}_H \ . \tag{2}$$

We have $\mathbf{c} \cdot M = (\gamma - 1)\mathbf{1}$. Thus $1/(\gamma - 1)\ \mathbf{c} \cdot M = \mathbf{1}$, and the program accepts the set of variables that label the rows corresponding to nonzero coordinates of $\mathbf{c}$.

Recall that each $\mathbf{c}_A$ has nonzero coordinates only at rows labeled by variables from A. Thus, for $A \cap Y = \emptyset$ the coordinates of $\mathbf{c}_A$ are zero at rows labeled by variables from Y. By Equation (1),

$$\mathbf{c} = \mathbf{0} - \sum_{A \in \mathcal{A}, A \cap Y = \emptyset} \alpha_A \mathbf{c}_A \ .$$

Therefore, the vector $\mathbf{c}$ has zero coordinates at all rows labeled by variables from Y.

On the other hand, by Equation (2) all the nonzero coordinates of $\mathbf{c}$ are at rows labeled by variables that appear in some sets A such that $A \cap Y \neq \emptyset$. Therefore, the program accepts $S_Y = \bigcup_{A \in \mathcal{H}, A \cap Y \neq \emptyset} A \setminus Y$, that (by Definition 1) does not contain any minterm of f. This proves the lemma.

From Lemma 1, we get a system of linear equations in the unknowns α_A. We prove that this system of equations has no solution, contradicting Equation (1). Suppose that $|T_H| = t$. Let us consider the following $(2^t - 1) \times (2^t - 1)$ zero–one matrix Q. The rows and columns of Q are indexed by the nonempty subsets of T_H, and $Q(Y, Z) = 1$ if and only if $Y \cap Z \neq \emptyset$.

Observation 2. The matrix Q has full rank over any field $\mathcal{F}$.

(This can be shown by a simple transformation of Q to a triangular matrix, or by induction.)

We show that if Equation (1) holds then, taking $\beta_Z = \sum_{A \in \mathcal{A}, A \cap T_H = Z} \alpha_A$ as a coefficient for the column $Z \neq T_H$, we get the column indexed by T_H as a linear combination of the other columns of Q. Notice that the column of Q indexed by T_H consists of all 1's. We show that for any Y, $\emptyset \neq Y \subseteq T_H$, we have $\sum_{\emptyset \neq Z \subset T_H} \beta_Z Q(Y, Z) = 1$.

By Condition C1 of Definition 1, for $A \in \mathcal{A}$ we have $A \cap T_H \neq T_H$. If $Y \subseteq T_H$ then $A \cap Y = A \cap T_H \cap Y$. By Lemma 1, if Equation (1) holds we have

$$1 = \sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A = \sum_{\emptyset \neq Z \subset T_H, Z \cap Y \neq \emptyset} \left(\sum_{A \in \mathcal{A}, A \cap T_H = Z} \alpha_A \right) = \sum_{\emptyset \neq Z \subset T_H} \beta_Z Q(Y, Z) \, ,$$

and the column T_H is a linear combination of the other columns of Q. Since Q has full rank this is not possible, and so Equation (1) cannot hold, i.e., the vectors $\mathbf{c}_H$ for $H \in \mathcal{H}$ are linearly independent. This concludes the proof of the theorem.

4 Lower bounds for clique functions

We consider the function $Clique_{k,n}$, 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 k.

It is known ([1, 31]) that the monotone circuit complexity of $Clique_{k,n}$ is $2^{\Omega(\sqrt{k})}$ for $k = O((n/\log n)^{2/3})$, and for fixed k it is $\Omega((n/\log n)^k)$. However, the strongest known lower bound for the monotone span program complexity of the $Clique_{k,n}$ function is our $\Omega(n^5) = \Omega(m^{2.5})$ lower bound that holds for $k \geq 6$. For $k \leq 4$, we obtain lower bounds that are tight, up to a constant factor.

For given k, we partition the set of n vertices into k classes $C_i, i = 1, \ldots, k$, of approximately equal size. We say that a k-clique is *multicolored* if each of its k vertices belong to a different class. Thus a multicolored clique will never contain an edge between two vertices in the same class.

Let $\mathcal{M}$ be an arbitrary family of multicolored k-cliques. Let T_K be some subset of the edges of the clique $K \in \mathcal{M}$. Denote the vertices of K by $v_1, \ldots, v_k$, and consider for $Y \subseteq T_K$ the set $S_Y = \bigcup_{G \in \mathcal{M}, G \cap Y \neq \emptyset} G \setminus Y$. Suppose S_Y contains a k-clique Z with vertices $z_1, \ldots, z_k$.

First we present two simple but important observations that are helpful in finding critical families for clique functions.

Claim 1. The vertices of Z all belong to different classes, say $z_i \in C_i$, for $i = 1, \ldots, k$.

Proof. S_Y only contains edges that appear in k-cliques that belong to the family $\mathcal{M}$, and so contains only edges between vertices from different classes.

We always list the vertices of a multicolored clique in the order of the partition classes.

Claim 2. For each edge $(v_i, v_j) \in Y$ at least one of $z_i \neq v_i$ or $z_j \neq v_j$ must hold.

Proof. If Z contained both v_i and v_j for $(v_i, v_j) \in Y$ then Z could not be a k-clique contained in S_Y since S_Y does not contain an edge between v_i and v_j.

We are ready to construct the critical families.

Lemma 2. For any partition of the n vertices into three classes, the family $\mathcal{M}$ of multicolored 3-cliques is critical for $Clique_{3,n}$.

Proof. Let H be an arbitrary multicolored 3-clique (triangle), and let T_H be the set of two of its edges, for example (v_1, v_2) and (v_2, v_3). There is only one triangle containing T_H, thus Condition C1 is satisfied. To see that Condition C2 holds, let us consider for $Y \subseteq T_H$ the set $S_Y = \cup_{G \in \mathcal{M}, G \cap Y \neq \emptyset} G \setminus Y$, and suppose that it contains a triangle Z with vertices z_1, z_2, z_3.

If $Y = T_H$, then $z_2 = v_2$ must hold, since there are no edges in S_Y incident to any other vertex from C_2. By Claim 2 we have $z_1 \neq v_1$ and $z_3 \neq v_3$. Therefore, the edge (z_1, z_3) cannot be present in S_Y, since all the edges of S_Y are contributed by triangles that contain at least one of v_1 or v_3.

If $Y \neq T_H$, then it consists of a single edge, (v_1, v_2) say. Then S_Y does not contain any edge between the classes C_1 and C_2, and so, by Claim 1, cannot contain a triangle.

Lemma 3. Given any partition of the n vertices into four classes, the family of multicolored 4-cliques is critical for $Clique_{4,n}$.

Proof. Let H be an arbitrary multicolored 4-clique, and let T_H be the set of two of its nonadjacent edges, for example (v_1, v_2) and (v_3, v_4). Condition C1 is satisfied, since two nonadjacent edges uniquely determine a 4-clique. To see that Condition C2 holds, as in the previous lemma, let us consider S_Y for $Y \subseteq T_H$ and suppose that it contains a 4-clique Z with vertices z_1, z_2, z_3, z_4.

If $Y = T_H$ then, by Claim 2, without loss of generality we have $z_1 \neq v_1$. Any edges incident to z_1 could only be contributed to S_Y by cliques that contain (v_3, v_4). Thus, a clique containing z_1 would also have to contain both v_3 and v_4, which is not possible by Claim 2.

As in the previous lemma, if $Y \neq T_H$ then it consists of a single edge, and S_Y does not contain a 4-clique.

We note that for $k \geq 5$ the family of all multicolored k-cliques is not critical for $Clique_{k,n}$. For example, for $k = 5$, any choice of T_H for a multicolored 5-clique H with vertices v_1, v_2, v_3, v_4, v_5, must contain either the set $Y_1 = \{(v_1, v_2), (v_1, v_3), (v_4, v_5)\}$ or $Y_2 = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5)\}$, up to renaming of the vertices. Each of the sets S_{Y_1} and S_{Y_2} contain the multicolored 5-clique on vertices v_1, z_2, z_3, z_4, z_5, where $z_i \neq v_i$ for $i = 2, \ldots 5$.

The critical families we use for proving lower bounds for 5-cliques and 6-cliques will be appropriately chosen subfamilies of multicolored cliques.

Theorem 4. For every field $\mathcal{F}$,

$$\mathrm{mSP}_{\mathcal{F}}(\mathit{Clique}_{6,n}) = \Omega(m^{2.5}) \ .$$

Proof. We show that the family of minterms of the $\mathit{Clique}_{6,n}$ function contains a large critical subfamily $\mathcal{K}$. Let us assume that $n = 6q$, and partition the set of n vertices into six classes of size q. The family $\mathcal{K}$ will be a subfamily of the multicolored 6-cliques under this partition.

For members of $\mathcal{K}$, we restrict the edges allowed between vertices in the classes C_1 and C_3, and similarly between the classes C_4 and C_6. The *legal* pairs of vertices which we allow to be connected by an edge will be specified by a $q \times q$ Boolean matrix N. Between all other pairs of classes we allow arbitrary edges. The edge (a, b) with $a \in C_1$ and $b \in C_3$ ($a \in C_4$ and $b \in C_6$, respectively) is allowed in a member of $\mathcal{K}$ if and only if $N(a, b) = 1$. We choose N such that it does not contain any complete (all ones) 2×2 submatrices. For example the incidence matrix of a projective plane has this property, and its number of ones is $\Theta(q^{3/2})$, with $\Theta(q^{1/2})$ ones in each row and column. The constructions in [26] and [29] can also be used. (The construction of matrices with similar properties for arbitrary q is described in [30].)

The family $\mathcal{K}$ consists of all the multicolored 6-cliques that satisfy the restriction on the edges between classes C_1 and C_3, and between C_4 and C_6. The number of such 6-cliques is $\Theta(q^5)$, thus we have $|\mathcal{K}| = \Theta(q^5) = \Theta(m^{2.5})$.

Next we show that $\mathcal{K}$ is critical for $\mathit{Clique}_{6,n}$. Consider any $K \in \mathcal{K}$, and denote its vertices by $v_1, \ldots, v_6$. The set T_K we choose will consist of the four edges (v_1, v_2), (v_2, v_3), (v_4, v_5), (v_5, v_6). Obviously, Condition C1 is satisfied.

We prove that Condition C2 holds. For $Y \subseteq T_K$, suppose the set $S_Y = \cup_{G \in \mathcal{K}, G \cap Y \neq \emptyset} G \setminus Y$ contains a 6-clique Z with vertices $z_1, \ldots, z_6$.

Case 1. Let $Y = T_K$. Notice that if both $z_2 \neq v_2$ and $z_5 \neq v_5$, then S_Y does not contain an edge between z_2 and z_5, thus we have $z_2 = v_2$ or $z_5 = v_5$. The remaining possibilities are illustrated in Figure 1.

Suppose that only one of these equalities holds, for example $z_2 = v_2$ but $z_5 \neq v_5$. Then, by Claim 2, $z_1 \neq v_1$ and $z_3 \neq v_3$ must hold. The edge (z_1, z_5) can only be contributed to S_Y by a clique that contains the edge (v_2, v_3), and similarly the edge (z_3, z_5) can only be contributed to S_Y by a clique that contains the edge (v_2, v_1). This means that the edges $(z_1, v_3), (v_1, z_3)$ as well as the edges (v_1, v_3) and (z_1, z_3) appear in some member of the family $\mathcal{K}$. However, this is not possible by our restriction on the legal edges between C_1 and C_3.

Suppose now that both $z_2 = v_2$ and $z_5 = v_5$ holds. Then by Claim 2 we have $z_1 \neq v_1$, $z_3 \neq v_3$, $z_4 \neq v_4$ and $z_6 \neq v_6$. The edge (z_1, z_4) can only be contributed by a clique that contains (v_2, v_3) or (v_5, v_6). This means that at least one of the edges (z_4, v_6) or (z_1, v_3) is legal. Similarly, from the presence in Z of the edges (z_1, z_6), (z_3, z_4) and (z_3, z_6), respectively, we know that at least one each of (v_4, z_6) or (z_1, v_3), (z_4, v_6) or (v_1, z_3), and (v_4, z_6) or (v_1, z_3), respectively, are legal edges. This means that either both (z_4, v_6) and

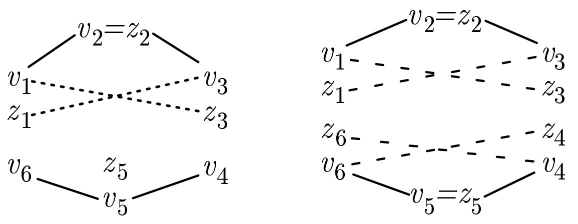

Figure 1: Illustrations for Case 1

(v_4, z_6) or both (z_1, v_3) and (v_1, z_3) are legal, and since (v_i, v_j) and (z_i, z_j) must be legal for all i, j, we get a contradiction with our restriction on the possible edges of members from $\mathcal{K}$.

Case 2. Let $Y \neq T_K$. In this case the edges in Y cover t vertices, $2 \leq t \leq 5$. We show that S_Y does not even contain a t-clique on the t classes involved. For $t \leq 4$ this follows directly from Lemma 2 and Lemma 3.

We still have to deal with the case when $t = 5$, which can only happen if Y consists of three edges. Suppose (without loss of generality) that the three edges of Y are (v_1, v_2), (v_2, v_3) and (v_4, v_5). If $z_2 \neq v_2$, then all the edges incident to z_2 could only be contributed to S_Y by cliques that contain (v_4, v_5). That would mean that the only vertices in C_4 and C_5 connected to z_2 in S_Y are v_4 and v_5. Thus we could not get a 6-clique in S_Y that contains z_2. Therefore, $z_2 = v_2$ must hold. Then we have by Claim 2 that $z_1 \neq v_1$, $z_3 \neq v_3$ and, without loss of generality, $z_5 \neq v_5$. We get a contradiction with the restriction on the edges between C_1 and C_3 as in Case 1.

We have proved that Condition C2 is also satisfied, and $\mathcal{K}$ is a critical family for f. The lower bound follows from Theorem 3.

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

$$\begin{array}{rcl} \mathrm{mSP}_{\mathcal{F}}(Clique_{5,n}) & = & \Omega(m^{2.25}) \ , \\ \mathrm{mSP}_{\mathcal{F}}(Clique_{4,n}) & \geq & 3(n/4)^4 - O(n^2) \ = \ \Omega(m^2) \ , \\ \mathrm{mSP}_{\mathcal{F}}(Clique_{3,n}) & \geq & 2(n/3)^3 - O(n) \ = \ \Omega(m^{1.5}) \ . \end{array}$$

The proof of this theorem is basically included in the proof of the lower bound for 6-cliques and in Lemmas 2 and 3. The bounds for $Clique_{4,n}$ and $Clique_{3,n}$ are slightly stronger (by constant factors) than the bound directly implied by Theorem 3. These constants are implied by the proof in [5]. 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 the set of multicolored 4-cliques defined for a fixed partition of the vertices into four classes of sizes as equal as possible. We observe that the above lower bound applies to this function as well, and is asymptotically tight in this case.

Corollary 6. Let $n = 4q$. Then, for every field $\mathcal{F}$,

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

For the upper bound it is enough to show a monotone formula whose size is $3q^4 + 3q^3$. For this, first note that the DNF formula with a term for every clique has size $6q^4$. By grouping all the cliques with the same vertices in the first three classes, we construct a monotone formula of the desired size.

5 A function with minterms of size 2

In [17] it is proved that there exists an explicit Boolean function on m variables with minterms of size 2 for which the sum of the lengths of the shares in every secret-sharing scheme is $\Omega(m \log m)$ times the length of the secret (for every finite set of possible secrets). That is, its monotone span program complexity is $\Omega(m \log m)$. In this section we exhibit an explicit function whose minterms are of size 2 with monotone span program complexity $\Omega(m^{3/2})$. Let $L_1, \ldots, L_n$ be n subsets of $\{1, \ldots, n\}$ such that the intersection of every two subsets is of size at most one. For example, the lines of a projective plane can be used. Given the sets $L_1, \ldots, L_n$, we define the function *Lines*, which has $m = 2n$ variables denoted $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$, and whose minterms are $\{\{a_i, b_j\} : j \in L_i\}$.

Theorem 7. For every field $\mathcal{F}$,

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

Proof. We prove that the family of all minterms of the function *Lines* is a critical family for *Lines*. The set T_H for every minterm H is simply H, and so Condition C1 is obviously satisfied.

To prove Condition C2, we take an arbitrary minterm, say $\{a_1, b_1\}$ without loss of generality, and consider the set $X = S_{\{a_1, b_1\}} = \{b_j : j \in L_1\} \cup \{a_i : 1 \in L_i\} \setminus \{a_1, b_1\}$. Suppose that there is some minterm $\{a_i, b_j\}$ 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 one. Obviously, the sets $S_{\{a_1\}}$ and $S_{\{b_1\}}$ do not contain any minterms either.

Using the lines of a projective plane or the constructions from [26, 29] and [30] for the sets $L_1, \ldots, L_n$ we have $\sum_{i=1}^{n} |L_i| = n^{3/2} + O(n)$. There exists a monotone formula for this function of size $n^{3/2} + O(n)$ (again, take the DNF formula with a term for every minterm and group the terms that include each a_i). Thus, we show an asymptotically matching upper bound for this function.

Corollary 8. For some explicitly given sets $L_1, \ldots, L_n$ we have, for every field $\mathcal{F}$,

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

Acknowledgements

We would like to thank Avi Wigderson, László Babai, Noam Nisan, Benny Chor and Teresa Przytycka for helpful discussions. Part of this work was done while the authors were visiting BRICS, Dept. of Computer Science, Aarhus, Denmark. Anna Gál is grateful to László Babai, Lance Fortnow and Ketan Mulmuley for supporting her visit to BRICS. Amos Beimel is grateful to Benny Chor for supporting his visit to BRICS. A preliminary version of this paper was presented at the 36th IEEE Symposium FOCS '95.

References

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

[2] L. Babai, A. Gál, J. Kollár, L. Rónyai, T. Szabó, A. Wigderson. Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs. Manuscript, November 1995.

[3] A. Beimel. Ideal Secret Sharing Schemes. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1992. (In Hebrew, abstract in English).

[4] A. Beimel and B. Chor. Universally ideal secret sharing schemes. *IEEE Transactions on Information Theory*, IT-40,3 (1994) 786–794.

[5] A. Beimel, A. Gál and M. Paterson. Lower bounds for monotone span programs. Technical Report BRICS-RS-94-46, BRICS, Department of Computer Science, University of Aarhus, December 1994.

[6] A. Beimel, A. Gál and M. Paterson. Lower bounds for monotone span programs. In *Proceedings of the 36th IEEE Symposium on the Foundations of Computer Science*, (1995) 674–681.

[7] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88*, *Lecture Notes in Computer Science* 403, (Springer-Verlag 1990) 27–35.

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

[9] 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.

[10] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 NCC, vol. 48*, (1979) 313–317.

[11] 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*, *Lecture Notes in Computer Science* 740, (Springer-Verlag 1993) 148–167.

[12] E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4(73)(1991) 123–134.

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

[14] 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.

[15] L. Csirmaz. The size of a share must be large . In A. De Santis, editor, *Advances in Cryptology – Eurocrypt '94*, *Lecture Notes in Computer Science* 950, (Springer-Verlag 1995), 13-22.

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

[17] M. van Dijk. On the information rate of perfect secret sharing schemes. *Designs, Codes, and Cryptography*, 6 (1995) 143–169.

[18] M. van Dijk. A linear construction of perfect secret sharing schemes. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt '94*, *Lecture Notes in Computer Science* 950, (Springer-Verlag 1995), 23–34.

[19] 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.

[20] W. Jackson and K. M. Martin. Geometric secret sharing schemes and their duals. *Designs, Codes and Cryptography*, 4 (1994), 83–95.

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

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

[23] 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.

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

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

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

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

[28] E. I. Nečiporuk. 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.

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

[30] N. Pippenger. On another Boolean matrix. *Theoretical Computer Science*, 11 (1980) 49–56.

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

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

[33] A. Shamir. How to share a secret. *Communications of the ACM*, 22 (1979) 612–613.

[34] G. J. Simmons. How to (really) share a secret. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88*, *Lecture Notes in Computer Science* 403, (Springer-Verlag 1990) 390–448.

[35] G. J. Simmons. An introduction to shared secret and/or shared control and their application. In G. J. Simmons, editor, *Contemporary Cryptology, The Science of Information Integrity*, (IEEE Press 1991), 441–497.

[36] G. J. Simmons, W. Jackson, and K. M. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1 (1991) 71–88.

[37] D. R. Stinson. An explicitation of secret sharing schemes. *Design, Codes and Cryptography*, 2 (1992) 357–390.

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

[39] 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.