On the Size of Monotone Span Programs

Ventzislav Nikov[1], Svetla Nikova[2] *, and Bart Preneel[2]

[1] Department of Mathematics and Computing Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands
`v.nikov@tue.nl`
[2] Department Electrical Engineering, ESAT/COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,
B-3001 Heverlee-Leuven, Belgium
`svetla.nikova,bart.preneel@esat.kuleuven.ac.be`

Abstract. Span programs provide a linear algebraic model of computation. Monotone span programs (MSP) correspond to linear secret sharing schemes. This paper studies the properties of monotone span programs related to their size. Using the results of van Dijk (connecting codes and MSPs) and a construction for a dual monotone span program proposed by Cramer and Fehr we prove a non-trivial upper bound for the size of monotone span programs. By combining the concept of critical families with the dual monotone span program construction of Cramer and Fehr we improve the known lower bound with a constant factor, showing that the lower bound for the size of monotone span programs should be approximately twice as large. Finally, we extend the result of van Dijk showing that for any MSP there exists a dual MSP such that the corresponding codes are dual.

1   Introduction

Motivation and Related Work. Span programs have been introduced in 1993 by Karchmer and Wigderson in [14] as a linear algebraic model of computation. A span program for a Boolean function is presented as a matrix over a field with rows labelled by literals of the variables, and the size of the program is the number of the 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. A span program is *monotone* if only positive literals are used as labels of the rows.

One main motivation to study span programs is that lower bounds for their size imply lower bounds for formula size and other interesting complexity measures including branching program size. The class of functions computable by polynomial size span programs over $GF(2)$ is equivalent to the class of functions computable by polynomial size parity branching programs [7, 14]. Span programs over other fields are related to other logspace classes [1, 7, 14].

Monotone span programs (MSPs) are also closely related to the cryptographic primitive *secret sharing schemes.* The concept of *secret sharing* was introduced by Shamir [18] as a tool to protect a secret simultaneously from exposure and from being lost. It allows a so called *dealer* to share the secret among a set of entities, usually called *players*, in such a way that only certain specified subsets of the players are able to reconstruct the secret while smaller subsets have no information about it. Monotone span programs are equivalent to a subclass of secret sharing schemes called *linear secret sharing schemes* (LSSSs). The size of MSPs measures the amount of information that has to be given to the participants in LSSSs. Lower bounds on MSPs imply lower bounds on the length of the shares in LSSSs.

In cryptographic multi-party protocols a general question is to find a "good measure", so that "often" the protocols are polynomially efficient in the number of players. Let *complexity* mean the total number of rounds, bits exchanged, local computations performed, etc. The best measure known for the *efficiency* of an SSS protocol is the *Monotone Span Program Complexity* [8] (which is the size of the MSP) and it coincides with the complexity in terms of linear secret sharing schemes over finite fields. Thus the question of estimating the MSP complexity (i.e. the size of the MSP) is a central question in several areas.

In a series of works [3, 5, 11] a lower bound for the size of an MSP has been proven. Later, Gal [12] proved that the MSP size is in the worst case superpolynomially (in the number of players) lower bounded. In addition it was proven in [4] that the size of MSPs over two fields with different characteristics is incomparable.

Our Results. We focus on studying the properties of MSPs related to their size. Using the results of van Dijk [10] (connecting codes and MSPs) and a construction for dual MSPs proposed by Cramer and Fehr [9] we prove a nontrivial upper bound for the size of MSPs. This result was announced in part in [16]. On the other hand using the same approach as in [3, 11] (critical families) together with the dual MSP construction of Cramer and Fehr [9] we improve the known lower bound with a constant factor; we show that the lower bound for the size of an MSP should be approximately twice as large. The rank of the matrix has been used a number of times to prove lower bounds on various types of complexity. In particular it has been used for the size of monotone formulas and monotone span programs [13]. We show that the nullity (the dimension of the kernel) of the matrix also should be taken into account when estimating the size of MSPs, since the nullity is linked to the rank of the matrix used in the dual MSP. Next we extend the result of van Dijk [10] showing that for any MSP $\mathcal{M}$ there exists a dual MSP $\mathcal{M}^\perp$ such that the corresponding codes $\mathcal{C}$ and $\mathcal{C}^\perp$ are dual.

Organization. In the next section we recall some definitions and notations that will be used later in the paper. In the first part of Sect. 3 we give some known properties of MSPs, then we describe our results: we modify the dual MSP construction of Cramer and Fehr and present an upper bound for the size of MSP in terms of the number of minimal and maximal sets in the access structure

computed by the MSP. In Sect. 4 we first present definitions and known results related to the approach developed in [3, 11, 12]; then we improve the known lower bound for the size of an MSP.

2   Preliminaries

Let us denote the players in a Secret Sharing Scheme by P_i, $1 \leq i \leq n$, the set of all players by $\mathcal{P} = \{P_1, \ldots, P_n\}$ and the set of all subsets of $\mathcal{P}$ (i.e., the power set of $\mathcal{P}$) by $P(\mathcal{P})$. We call the groups which are allowed to reconstruct the secret *qualified* and the groups which should not be able to obtain any information about the secret *forbidden*. The set of qualified groups is denoted by Γ ($\Gamma \subseteq P(\mathcal{P})$) and the set of forbidden groups by Δ ($\Delta \subseteq P(\mathcal{P})$). The set Γ is called *monotone increasing* if for any set A in Γ any set containing A is also in Γ. Similarly, Δ is called *monotone decreasing*, if for each set B in Δ each subset of B is also in Δ. A monotone increasing set Γ can be efficiently described by the set Γ^- consisting of the *minimal elements* in Γ, i.e., the elements in Γ for which no proper subset is also in Γ. Similarly, the set Δ^+ consists of the *maximal elements* (sets) in Δ, i.e., the elements in Δ for which no proper superset is also in Δ. The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. It is obvious that (Γ^-, Δ^+) generates (Γ, Δ). If the union of Γ and Δ is equal to $P(\mathcal{P})$ (so Γ is equal to Δ^c, the complement of Δ), then we say that the access structure (Γ, Δ) is *complete* and we denote it just by Γ. Throughout the paper we will consider *connected access structures*, i.e., the access structures in which every player is in at least one minimal set. For a complete access structure the dual access structure could be defined as follows. The *dual access structure* $\Gamma^\perp$ of an access structure Γ, defined on $\mathcal{P}$, is the collection of sets $A \subseteq \mathcal{P}$ such that $\mathcal{P} \setminus A = A^c \notin \Gamma$.

In most of the works (e.g. [3, 5, 11, 14]) the connection between MSPs and monotone Boolean functions has been exploit. Here we will show that there is one-to-one correspondence between complete access structures and monotone Boolean functions. Associate with every player P_i a Boolean variable x_i. Then with any set $A \subseteq \mathcal{P}$ we associate a variable $x_A = (x_1, \ldots, x_n)$ by fixing $x_i = 1$ if and only if $P_i \in A$; x_A is sometimes called the *characteristic vector* of A. Now a one-to-one mapping between f and Γ is defined in the following way: $f(x_A) = 1$ if and only if $A \in \Gamma$. 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 are. Using the mapping between access structures and monotone functions, it is easy to see that minterms correspond to minimal sets. A *maxterm* of a monotone function is a minimal set of its variables with the property that the value of the function is 0 on any input that assigns 0 to each variable in the set, no matter what the values of the other variables are. Recall the one-to-one mapping between f and Γ. With this mapping in mind it is not difficult to verify that maxterms are equivalent to maximal sets. Let $f(x_1, \ldots, x_n)$ be a monotone Boolean function. Let $f^*(x_1, \ldots, x_n) = \overline{f(\overline{x_1}, \ldots, \overline{x_n})}$, sometimes f^* is called

the dual function of f. In fact the minterms of f^* are exactly the maxterms of f. Using again the one-to-one mapping between f and Γ it follows that if access structure Γ corresponds to a monotone function f, then the function f^* corresponds to the dual access structure $\Gamma^{\perp}$.

An SSS is linear if the dealer uses only linear operations to share (reconstruct) the secret amongst the participants. Each *linear SSS* (LSSS) can be viewed as derived from a monotone span program computing its access structure [8]. On the other hand, each monotone span program gives rise to an LSSS. Hence, one can identify an LSSS with its underlying monotone span program. Such an MSP always exists, because MSPs can compute any monotone access structure (see [2, 11, 14]). An important parameter of the MSP is its size, which turns out to be also the size of the corresponding LSSS (the sum of all shares).

Let us describe some of the tools we will employ. An $m \times d$ matrix M over a field $\mathbb{F}$ defines a map from $\mathbb{F}^d$ to $\mathbb{F}^m$ by taking a vector $\boldsymbol{v} \in \mathbb{F}^d$ to the vector $M\boldsymbol{v} \in \mathbb{F}^m$. Associated with an $m \times d$ matrix M (or a linear map) are two natural subspaces, one in $\mathbb{F}^m$ and the other in $\mathbb{F}^d$. They are defined as follows. The *kernel* of M (denoted by $\ker(M)$) is the set of vectors $\boldsymbol{u} \in \mathbb{F}^d$, such that $M\boldsymbol{u} = \boldsymbol{0}$. The *image* of M (denoted by $\operatorname{im}(M)$) is the set of vectors $\boldsymbol{v} \in \mathbb{F}^m$ such that $\boldsymbol{v} = M\boldsymbol{u}$ for some $\boldsymbol{u} \in \mathbb{F}^d$. The dimension of the image of M is called the *rank* of M, and the dimension of the kernel of M is called its *nullity*. A central result of linear algebra, called *the rank and nullity theorem* states that the dimensions of these two spaces add up to d, the number of columns in M. It is well known that the *column rank* of a matrix M (being the maximal size of a linearly independent set of columns of M) is equal to the *row rank* of M (which is the maximal size of an independent set of rows). The space generated by the rows of a matrix M will sometimes be denoted by $\operatorname{span}(M)$.

For an arbitrary matrix M over a field $\mathbb{F}$, with m rows labelled by $1, \dots, m$ and for an arbitrary non-empty subset A of $\{1, \dots, m\}$, let M_A denote the matrix obtained by keeping only those rows i with $i \in A$. In the sequel $\mathbf{v^i}$ will denote a vector but $\mathbf{v}_i$ stands for the i-th coordinate of the vector $\mathbf{v}$. With the standard inner product $\langle \mathbf{v}, \mathbf{w} \rangle$ we write $\mathbf{v} \perp \mathbf{w}$, when $\langle \mathbf{v}, \mathbf{w} \rangle = 0$. For an $\mathbb{F}$-linear subspace $\mathcal{V}$ of $\mathbb{F}^d$, $\mathcal{V}^{\perp}$ denotes the collection of elements of $\mathbb{F}^d$, that are orthogonal to all of $\mathcal{V}$ (the orthogonal complement). It is again an $\mathbb{F}$-linear subspace. For all subspaces $\mathcal{V}$ of $\mathbb{F}^d$ we have $\mathcal{V} = (\mathcal{V}^{\perp})^{\perp}$. Other standard relations are $(\operatorname{im}(M^T))^{\perp} = \ker(M)$ or $\operatorname{im}(M^T) = (\ker(M))^{\perp}$, as well as $\langle \mathbf{v}, M^T\mathbf{w} \rangle = \langle M\mathbf{v}, \mathbf{w} \rangle$.

Let $\mathbb{F}$ be a finite field and let the set of secrets be $\mathcal{K} = \mathbb{F}^{p_0}$, with $p_0 = 1$. Associate with each player P_i ($1 \leq i \leq n$) a positive integer p_i such that the sets of possible shares for player P_i, is a linear subspace $\mathcal{S}_i = \mathbb{F}^{p_i}$. Denote by $p = \sum_{i=1}^{n} p_i$ and by $N = p_0 + p$, then the sharing space $\mathcal{S} = \mathcal{S}_1 \times \dots \times \mathcal{S}_n = \mathbb{F}^p$ and $\mathcal{K} \times \mathcal{S} = \mathbb{F}^N$.

Definition 1. *[10] Consider the vector* $\mathbf{v} \in \mathbb{F}^N$. *The coordinates in* $\mathbf{v}$, *which belong to player* P_i *are collected in a sub-vector denoted by* $\mathbf{v^i}$ *and the coordinates that correspond to the secret, i.e., to the dealer* $\mathcal{D}$ *are collected in a sub-vector denoted by* $\mathbf{v^0}$ *or in other words* $\mathbf{v} = (\mathbf{v^0}, \mathbf{v^1}, \dots, \mathbf{v^n})$ *where* $\mathbf{v^i} \in \mathbb{F}^{p_i}$. *The* p-

support *of a vector* $\mathbf{v}$, *denoted by* $sup_p(\mathbf{v})$, *is defined as the set of coordinates* i, $0 \leq i \leq n$ *for which* $\mathbf{v^i} \neq \mathbf{0}$, *i.e.,* $sup_p(\mathbf{v}) = \{i \; : \; \mathbf{v^i} \neq \mathbf{0}\}$.

Now we give a formal definition of a Monotone Span Program.

Definition 2. *[14] A* Monotone Span Program *(MSP)* $\mathcal{M}$ *is a quadruple* $(\mathbb{F}, M, \boldsymbol{\varepsilon}, \psi)$, *where* $\mathbb{F}$ *is a finite field,* M *is a matrix (with* m *rows and* $d \leq m$ *columns) over* $\mathbb{F}$, $\psi : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$ *is a surjective function and* $\boldsymbol{\varepsilon}$ *is a fixed non-zero vector, called* target vector, *e.g., the column vector* $(1, 0, \ldots, 0)^T \in \mathbb{F}^d$. *The size of* $\mathcal{M}$ *is the number* m *of rows and is denoted as* size$(\mathcal{M})$.

As ψ labels each row with an integer i from $\{1, \ldots, m\}$ that corresponds to player $P_{\psi(i)}$, we can think of each player as being the "owner" of one or more rows. Also consider a "function" φ from $[P_1, \ldots, P_n]$ to $\{1, \ldots, m\}$ which gives for every player P_i the set of rows owned by him (denoted by $\varphi(P_i)$). In some sense φ is the "inverse" of ψ. For any set of players $B \subseteq \mathcal{P}$ consider the matrix consisting of rows these players own in M, i.e. $M_{\varphi(B)}$. As it is common, we shall shorten the notation $M_{\varphi(B)}$ to just M_B. The reader should be aware of the difference between M_B for $B \subseteq \mathcal{P}$ and for $B \subseteq \{1, \ldots, m\}$.

An MSP is said *to compute* a (complete) access structure Γ when $\boldsymbol{\varepsilon} \in \text{im}(M_A^T)$ if and only if A is a member of Γ. We say that A is *accepted* by $\mathcal{M}$ if and only if $A \in \Gamma$, otherwise we say A is *rejected* by $\mathcal{M}$. In other words, the players in A can reconstruct the secret precisely if the rows they own contain in their linear span the target vector of $\mathcal{M}$, and otherwise they get no information about the secret. There exists a so-called *recombination vector* $\boldsymbol{\lambda}$ such that $M_A^T \boldsymbol{\lambda} = \boldsymbol{\varepsilon}$ hence $\langle \boldsymbol{\lambda}, M_A(s, \boldsymbol{\rho})^T \rangle = \langle M_A^T \boldsymbol{\lambda}, (s, \boldsymbol{\rho})^T \rangle = \langle \boldsymbol{\varepsilon}, (s, \boldsymbol{\rho})^T \rangle = s$ for any secret s and any random vector $\boldsymbol{\rho}$. It is easy to check that the vector $\boldsymbol{\varepsilon} \notin \text{im}(M_B^T)$ if and only if there exists a vector $\boldsymbol{k} \in \mathbb{F}^d$ such that $M_B \boldsymbol{k} = \mathbf{0}$ and $\boldsymbol{k}_1 = 1$. Technically these properties mean that when we consider the restricted matrix M_A for some subset A of $\mathcal{P}$, the first column is linearly dependent on the other columns if and only if $A \notin \Gamma$.

Note 1. [3, 11] It is well known that the number d of columns in an MSP $\mathcal{M}$ can be increased without changing the access structure computed by it. The space generated by the 2-nd up to the d-th column of M does not contain even a non-zero multiple of the first column. Without changing the access structure that is computed, we can always replace the 2-nd up to the d-th column of M by any set of vectors that generates the same space.

3 On Upper Bounds for the Size of MSPs

We will start with some known properties of MSPs. Cramer and Fehr [9] proposed a method to construct the dual MSP (i.e. the MSP computing the dual access structure $\Gamma^\perp$) starting from the MSP computing a given access structure Γ.

Lemma 1. *[9] Let an MSP* $\mathcal{M} = (\mathbb{F}, M, \boldsymbol{\varepsilon}, \psi)$ *compute* Γ. *Denote by* $\boldsymbol{\lambda}$ *a solution of the equation* $M^T \boldsymbol{\lambda} = \boldsymbol{\varepsilon}$ *and let* $\mathbf{b^1}, \mathbf{b^2}, \ldots, \mathbf{b^\ell}$ *denote an arbitrary generating set of* $\ker(M^T)$, *(l = nullity(M^T)). Then* $\mathcal{M}^\perp = (\mathbb{F}, M^\perp, \boldsymbol{\varepsilon}^*, \psi)$ *is an*

MSP computing $\Gamma^\perp$, where $M^\perp = [\boldsymbol{\lambda}, \mathbf{b^1}, \mathbf{b^2}, \ldots, \mathbf{b^\ell}]$ and $\boldsymbol{\varepsilon}^*$ is the column vector $(1, 0, \ldots, 0)^T \in \mathbb{F}^{\ell+1}$.

Note 2. Let us define the $d \times (\ell+1)$ matrix $\overline{E}$ to be a zero matrix except for the entry in the upper left corner which is 1, or in other words $\overline{E} = \boldsymbol{\varepsilon}(\boldsymbol{\varepsilon}^*)^T$. Then it follows from the construction proposed in Lemma 1 that the matrices M and $M^\perp$ satisfy the following equation $M^T M^\perp = \overline{E}$.

In his Ph.D. thesis van Dijk [10] investigates the more general setting when more than one secret (e.g. $s_1, \ldots, s_{p_0} \in \mathbb{F}$) should be shared with an access structure. Note that this approach allows consideration of incomplete access structures. Van Dijk proposed a method (using the generalized vector space construction) to build matrices which have the properties equivalent to the MSP. Recall that we consider only the case $p_0 = 1$, i.e. $s \in \mathbb{F}$. It is worth to note that because of [10, Lemma 3.4.14] when we share only one secret (i.e. $p_0 = 1$), the generalized vector space construction that computes (Γ, Δ) coincides with the generalized vector space construction that computes Note that this is exactly the case for an MSP, where we consider only one secret and a complete access structure.

Definition 3. *([10, Definition 3.2.2]) Let $\Gamma^- = \{X_1, \ldots, X_r\}$. Then the set of vectors $C = \{\mathbf{c^i} \in \mathbb{F}^m : 1 \le i \le r\}$ is said to be* suitable *for the access structure Γ if C satisfies the following properties called $g(\Gamma)$ respectively $d^-(\Delta)$.*

- $sup_P(\mathbf{c^i}) = X_i$ *for $1 \le i \le r$;*
- *For any vector $(\mu_1, \ldots, \mu_r)$ in $\mathbb{F}^r$, such that $\sum_{i=1}^r \mu_i \ne 0$, there exists a set $X \in \Gamma = \Delta^c$ satisfying $X \subseteq sup_P(\sum_{i=1}^r \mu_i \mathbf{c^i})$.*

In the next theorem van Dijk provides an important link between a parity check matrix of a code generated as a span of suitable vectors and the MSP matrix.

Theorem 1. *([10, Theorem 3.2.5, Theorem 3.2.6]) Let $\Gamma^- = \{X_1, \ldots, X_r\}$. Consider a set of vectors $C = \{\mathbf{c^i} : 1 \le i \le r\}$. Let H be a parity check matrix of the code generated by the linear span of the vectors $(1, \mathbf{c^i})$, $1 \le i \le r$ and let H be of the form $H = (\boldsymbol{\varepsilon} \mid H')$ (This can be assumed without loss of generality). Then the MSP with the matrix M defined by $M^T = H'$ computes the access structure Γ if and only if the set of vectors C is suitable for Γ.*

There is a tight connection between an access structure and its dual. It turns out that the codes generated by the corresponding sets of suitable vectors are orthogonal.

Theorem 2. *([10, Theorem 3.5.4]) Let $\Gamma^- = \{X_1, \ldots, X_r\}$ be an access structure and $(\Gamma^\perp)^- = \{Z_1, \ldots, Z_t\}$ be its dual. Then there exists a suitable set $C = \{\mathbf{c^i} : 1 \le i \le r\}$ for Γ if and only if there exists a suitable set $C^\perp = \{\mathbf{h^j} : 1 \le j \le t\}$ for $\Gamma^\perp$.*
Suppose there exists a suitable set C for Γ and a suitable set $C^\perp$ for $\Gamma^\perp$. Let $\mathcal{C}^*$ be the code defined by the linear span of vectors $\{(1, \mathbf{c^i}) : 1 \le i \le r\}$ and let $\mathcal{C}^\perp$ be the code defined by the linear span of vectors of $\{(1, \mathbf{h^j}) : 1 \le j \le t\}$. Then the codes $\mathcal{C}^*$ and $\mathcal{C}^\perp$ are orthogonal to each other.

Note that $\mathcal{C}^*$ and $\mathcal{C}^\perp$ are not necessarily each other's dual. Now we point out that the suitable set of vectors are in fact the solutions $\boldsymbol{\lambda}$ of the equation $M^T\boldsymbol{\lambda} = \boldsymbol{\varepsilon}$ or in other words the suitable set of vectors consists of recombination vectors.

Lemma 2. *Let $\Gamma^- = \{X_1, \ldots, X_r\}$ be the access structure computed by MSP $\mathcal{M}$. Also let $\boldsymbol{\lambda^i} \in \mathbb{F}^m$ be the recombination vector that corresponds to X_i. Then the set of vectors $C = \{\boldsymbol{\lambda^i} : 1 \le i \le r\}$ defines a suitable set of vectors for the complete access structure Γ.*

Recall that Cramer and Fehr [9] proposed a method to construct the dual MSP (i.e., the MSP computing the dual access structure $\Gamma^\perp$) starting from the MSP computing the given access structure Γ (see Lemma 1). Now we will slightly modify their construction.

Lemma 3. *Let MSP $\mathcal{M} = (\mathbb{F}, M, \boldsymbol{\varepsilon}, \psi)$ compute access structure Γ. Let $\Gamma^- = \{X_1, \ldots, X_r\}$ be the set of minimal sets in Γ. For each X_i denote the corresponding recombination vector by $\boldsymbol{\lambda^i} \in \mathbb{F}^m$, so $M^T\boldsymbol{\lambda^i} = \boldsymbol{\varepsilon}$ and $\sup_P(\boldsymbol{\lambda^i}) = X_i$. Then there exists an MSP $\mathcal{M}^\perp = (\mathbb{F}, M^\perp, \boldsymbol{\varepsilon}^*, \psi)$ computing $\Gamma^\perp$, where $M^\perp = [\boldsymbol{\lambda^1}, \boldsymbol{\lambda^1} - \boldsymbol{\lambda^2}, \ldots, \boldsymbol{\lambda^1} - \boldsymbol{\lambda^r}]$ and $\boldsymbol{\varepsilon}^*$ is a column vector $(1, 0, \ldots, 0)^T$ of suitable length.*

Proof. We will follow the proof of Cramer and Fehr with some minor changes. Note that for any X_i there may be several recombination vectors $\boldsymbol{\lambda^i}$; we pick one of them and denote it by $\boldsymbol{\lambda^i}$. Note also that the vectors $\boldsymbol{\lambda^1} - \boldsymbol{\lambda^2}, \ldots, \boldsymbol{\lambda^1} - \boldsymbol{\lambda^r}$ from $\ker(M^T)$ may not generate the full kernel space.

If $A^c \notin \Gamma$, then there exists a vector $\mathbf{k}$ such that $M_{A^c}\mathbf{k} = \mathbf{0}$ and $\mathbf{k}_1 = 1$. Define $\boldsymbol{\lambda}^* = M_A\mathbf{k}$, or equivalently define $\boldsymbol{\lambda}^{**} = M\mathbf{k}$. Note again that $\boldsymbol{\lambda}^{**}_A = \boldsymbol{\lambda}^*$ and $\boldsymbol{\lambda}^{**}_{A^c} = \mathbf{0}$. Then $(M_A^\perp)^T\boldsymbol{\lambda}^* = (M^\perp)^T\boldsymbol{\lambda}^{**} = (M^\perp)^T(M\mathbf{k}) = ((M^\perp)^T M)\mathbf{k} = (M^T M^\perp)^T\mathbf{k} = \boldsymbol{\varepsilon}^*$, thus $A \in \Gamma^\perp$.

On the other hand, if $A^c \in \Gamma$, then there exists a vector $\widetilde{\boldsymbol{\lambda}}$ such that $M^T\widetilde{\boldsymbol{\lambda}} = \boldsymbol{\varepsilon}$ and $\sup_P(\widetilde{\boldsymbol{\lambda}}) \subseteq A^c$, i.e. $\widetilde{\boldsymbol{\lambda}}_A = \mathbf{0}$. Note that we can even choose $\widetilde{\boldsymbol{\lambda}}$ to be in the linear span of the vectors $\boldsymbol{\lambda^1}, \boldsymbol{\lambda^2}, \ldots, \boldsymbol{\lambda^r}$. Now by the definition of $M^\perp$, it follows that there exists a vector $\mathbf{k}$ such that $\mathbf{k}_1 = 1$ and $M^\perp\mathbf{k} = \widetilde{\boldsymbol{\lambda}}$, i.e. $M_A^\perp\mathbf{k} = \mathbf{0}$, thus $A \in \Delta^\perp$ which concludes the proof. $\square$

Thus Lemma 3 improves the construction of Cramer and Fehr (see Lemma 1) showing that a matrix with fewer columns suffices. Recall that $r = |\Gamma^-|$ and $t = |\Delta^+|$. Let

$$\bar{r} = \dim \quad \text{span}\{\boldsymbol{\lambda^i}; 1 \le i \le r\}. \tag{1}$$

Analogously define $\bar{t}$ for the dual MSP $\mathcal{M}^\perp$. Note that $\bar{r} \le r$ and $\bar{t} \le t$. Combining Lemma 3 and Note 1 yields a construction of an MSP with particular properties.

Lemma 4. *Let Γ be a connected access structure and let $\Gamma^\perp$ be its dual. Then there exist MSPs such that $M^\perp$ has size $m \times \bar{r}$ and M has size $m \times \bar{t}$, where $\bar{r}$ is defined by (1).*

Lemma 5. *Let Γ be a connected access structure and let $\Gamma^\perp$ be its dual. Then there exists an MSP program computing Γ with size:*

$$m = \bar{r} + \bar{t} - 1.$$

and such that the matrix $M^\perp$ has size $m \times \bar{r}$ and matrix M has size $m \times \bar{t}$.

Proof. Let $H = (\boldsymbol{\varepsilon} \mid M^T)$ and $H^\perp = (\boldsymbol{\varepsilon} \mid -(M^\perp)^T)$. We prove (see Lemma 2) that the vectors $(1, -\boldsymbol{\lambda^i})$ generate the code $\mathcal{C}$ since they are a suitable set of vectors. From the construction of the dual MSP (see Lemma 3) it follows that the generator matrix M can be rewritten as $G = (\boldsymbol{\varepsilon} \mid -(M^\perp)^T)$. But the last observation implies that these matrices are the same, i.e. $G = H^\perp$ holds. It is now straightforward to obtain the equality $\bar{r} + \bar{t} = m + 1$. Finally, note that because of Lemma 4 we have for M and $M^\perp$ that $M^\perp$ has size $m \times \bar{r}$ and M has size $m \times \bar{t}$. $\qquad\square$

Now we are ready to state the main result in this section.

Theorem 3. *Let Γ be a connected access structure and let $\Gamma^\perp$ be its dual. Let $|\Gamma^-| = r$ and $|(\Gamma^\perp)^-| = t$. Then for any field $\mathbb{F}$, there exists a monotone span program $\mathcal{M}$ computing Γ with size satisfying the following upper bound:*

$$size(\mathcal{M}) \leq r + t - 1.$$

Proof. From Lemma 5 and the obvious facts that $\bar{r} \leq r$ and $\bar{t} \leq t$ we obtain that $m \leq r + t - 1$. $\qquad\square$

Note 3. By Definition $(\Gamma^\perp)^- = \{Z_1, \ldots, Z_t\}$ implies that $\Delta^+ = \{Y_1, \ldots, Y_t\}$, with $Z_j = Y_j^c$. In other words the size of an MSP is limited from above by the sum of the number of minimal and the number of maximal sets minus one.

We will provide an alternative proof of Lemma 5 using van Dijk's approach. Recall that the matrix G is the generator matrix of the code $\mathcal{C}^*$, generated by the suitable set of vectors $(1, \mathbf{c^i}), 1 \leq i \leq r$. The matrix H is the parity check matrix of the code $\mathcal{C}^*$; it is of the form $H = (\boldsymbol{\varepsilon} \mid M^T)$. Analogously we have the matrix $G^\perp$ as a generator matrix of the code $\mathcal{C}^\perp$, generated by a suitable set of vectors $(1, \mathbf{h^j}), 1 \leq j \leq t$. The matrix $H^\perp$ is a parity check matrix for the code $\mathcal{C}^\perp$, is of the form $H^\perp = (\boldsymbol{\varepsilon} \mid (M^\perp)^T)$. Here we will use MSP $\mathcal{M}^\perp$ with target vector $-\boldsymbol{\varepsilon}$. If we summarize the results from Theorems 1 and 2 we have:

$$\begin{array}{rclcl} GH^T & = & HG^T & = 0 \\ G^\perp(H^\perp)^T & = & H^\perp(G^\perp)^T & = 0 \\ G(G^\perp)^T & = & G^\perp G^T & = 0 \end{array}$$

As we pointed out the codes $\mathcal{C}^*$ and $\mathcal{C}^\perp$ are not necessarily each other's dual, i.e. $H^\perp H^T = H(H^\perp)^T \neq 0$. Thus our goal now is to prove that for any MSP $\mathcal{M}$ there exists an MSP $\mathcal{M}^\perp$ such that $\mathcal{C}^*$ and $\mathcal{C}^\perp$ are dual, i.e. $\mathcal{C}^* = \mathcal{C}$.

Lemma 6. *Denote the linear span of the rows of matrices G and $H^\perp$ by $span(G)$ respectively $span(H^\perp)$. There are matrices G and $H^\perp$ such that $span(G) = span(H^\perp)$.*

Proof. As van Dijk proved in Theorem 2, $\text{span}(G) \subseteq \text{span}(H^{\perp})$ (since $G(G^{\perp})^T = G^{\perp}G^T = 0$). Note that these equations also mean that vectors $(1, \mathbf{c^i}); 1 \leq i \leq r$ and $(1, \mathbf{h^j}); 1 \leq j \leq t$ are orthogonal. Thus the matrices have the following form:

$$H = \begin{pmatrix} (1, \mathbf{h^1}) \\ (1, \mathbf{h^2}) \\ \vdots \\ (1, \widetilde{\mathbf{h}}^{\mathbf{1}}) \\ (1, \widetilde{\mathbf{h}}^{\mathbf{2}}) \\ \vdots \\ (0, \overline{\mathbf{h}}^{\mathbf{1}}) \\ (0, \overline{\mathbf{h}}^{\mathbf{2}}) \\ \vdots \end{pmatrix} \quad H^{\perp} = \begin{pmatrix} (1, \mathbf{c^1}) \\ (1, \mathbf{c^1}) \\ \vdots \\ (1, \widetilde{\mathbf{c}}^{\mathbf{1}}) \\ (1, \widetilde{\mathbf{c}}^{\mathbf{2}}) \\ \vdots \\ (0, \overline{\mathbf{c}}^{\mathbf{1}}) \\ (0, \overline{\mathbf{c}}^{\mathbf{2}}) \\ \vdots \end{pmatrix} \quad M^T = \begin{pmatrix} \mathbf{h^1} \\ \mathbf{h^2} - \mathbf{h^1} \\ \mathbf{h^3} - \mathbf{h^1} \\ \vdots \\ \widetilde{\mathbf{h}}^{\mathbf{1}} - \mathbf{h^1} \\ \widetilde{\mathbf{h}}^{\mathbf{2}} - \mathbf{h^1} \\ \vdots \\ \overline{\mathbf{h}}^{\mathbf{1}} \\ \overline{\mathbf{h}}^{\mathbf{2}} \\ \vdots \end{pmatrix} (M^{\perp})^T = \begin{pmatrix} \mathbf{c^1} \\ \mathbf{c^2} - \mathbf{c^1} \\ \mathbf{c^3} - \mathbf{c^1} \\ \vdots \\ \widetilde{\mathbf{c}}^{\mathbf{1}} - \mathbf{c^1} \\ \widetilde{\mathbf{c}}^{\mathbf{2}} - \mathbf{c^1} \\ \vdots \\ \overline{\mathbf{c}}^{\mathbf{1}} \\ \overline{\mathbf{c}}^{\mathbf{2}} \\ \vdots \end{pmatrix}$$

The matrix H consists of the row vectors $(1, \mathbf{h^j})$ and probably other vectors of the form $(1, \widetilde{\mathbf{h}}^{\mathbf{j}})$ and/or $(0, \overline{\mathbf{h}}^{\mathbf{j}})$ and all of them are orthogonal to $(1, \mathbf{c^i})$. Analogously, the matrix $H^{\perp}$ consists of the row vectors $(1, \mathbf{c^i})$ and probably other vectors of the form $(1, \widetilde{\mathbf{c}}^{\mathbf{i}})$ and/or $(0, \overline{\mathbf{c}}^{\mathbf{i}})$ and all of them are orthogonal to $(1, \mathbf{h^j})$. First, note that in the matrix $\overline{E}$ defined in Note 2 the entry in the upper left corner could be any non-zero number. Now this entry is -1 since we choose the target vector in $\mathcal{M}^{\perp}$ to be $-\boldsymbol{\varepsilon}$. Consider the equation $(M^{\perp})^T M = M^T M^{\perp} = \overline{E}$ from Note 2. This equation implies that the vectors $\mathbf{h^1}$, $\mathbf{h^j} - \mathbf{h^1}$, $\widetilde{\mathbf{h}}^{\mathbf{j}} - \mathbf{h^1}$ and $\overline{\mathbf{h}}^{\mathbf{j}}$ are orthogonal to the vectors $\mathbf{c^1}$, $\mathbf{c^i} - \mathbf{c^1}$, $\widetilde{\mathbf{c}}^{\mathbf{i}} - \mathbf{c^1}$ and $\overline{\mathbf{c}}^{\mathbf{i}}$, except that $\langle \mathbf{h^1}, \mathbf{c^1} \rangle = -1$ should hold. Now using the orthogonality relations between the vectors $(1, \mathbf{c^i})$ and the vectors $(1, \mathbf{h^j})$, $(1, \widetilde{\mathbf{h}}^{\mathbf{j}})$, $(0, \overline{\mathbf{h}}^{\mathbf{j}})$ and also between $(1, \mathbf{h^j})$ and $(1, \mathbf{c^i})$, $(1, \widetilde{\mathbf{c}}^{\mathbf{i}})$, $(0, \overline{\mathbf{c}}^{\mathbf{i}})$ we obtain:

$$\langle \overline{\mathbf{h}}^{\mathbf{j}}, \widetilde{\mathbf{c}}^{\mathbf{i}} \rangle = 0, \qquad \langle \overline{\mathbf{c}}^{\mathbf{i}}, \widetilde{\mathbf{h}}^{\mathbf{j}} \rangle = 0, \qquad \langle \overline{\mathbf{h}}^{\mathbf{j}}, \overline{\mathbf{c}}^{\mathbf{i}} \rangle = 0, \qquad \langle \widetilde{\mathbf{h}}^{\mathbf{j}}, \widetilde{\mathbf{c}}^{\mathbf{i}} \rangle = -1.$$

Thus, we have

$$\langle (0, \overline{\mathbf{h}}^{\mathbf{j}}), (1, \widetilde{\mathbf{c}}^{\mathbf{i}}) \rangle = 0, \qquad \langle (0, \overline{\mathbf{h}}^{\mathbf{j}}), (0, \overline{\mathbf{c}}^{\mathbf{i}}) \rangle = 0,$$

$$\langle (1, \widetilde{\mathbf{h}}^{\mathbf{j}}), (1, \widetilde{\mathbf{c}}^{\mathbf{i}}) \rangle = 0, \qquad \langle (1, \widetilde{\mathbf{h}}^{\mathbf{j}}), (0, \overline{\mathbf{c}}^{\mathbf{i}}) \rangle = 0.$$

Hence H is orthogonal to $H^{\perp}$, i.e. $H(H^{\perp})^T = H^{\perp}H^T = 0$ holds. But, now it immediately follows that $\text{span}(G) \supseteq \text{span}(H^{\perp})$. Hence $\text{span}(H^{\perp}) = \text{span}(G)$, which completes the proof. $\square$

Define

$$\widetilde{r} = \dim \; \text{span}\{(1, \mathbf{c^i}); 1 \leq i \leq r\}, \qquad (2)$$
$$\widetilde{t} = \dim \; \text{span}\{(1, \mathbf{h^j}); 1 \leq j \leq t\}.$$

Now we are in position to prove the following result.

Lemma 7. *Let Γ be a connected access structure and let $\Gamma^\perp$ be its dual. Then there exists an MSP program computing Γ of size m satisfying:*

$$m = \widetilde{r} + \widetilde{t} - 1.$$

and such that the matrix $M^\perp$ has size $m \times \widetilde{r}$ and the matrix M has size $m \times \widetilde{t}$.

Proof. We have that G is an $\widetilde{r} \times (m+1)$ matrix, since $\widetilde{r}$ is the dimension of the code $\mathcal{C}$. It also follows that $\widetilde{r} \leq r$. On the other hand H is a parity check matrix of code $\mathcal{C}$. Hence H is an $(m+1-\widetilde{r}) \times (m+1)$ matrix, and thus M is an $m \times (m+1-\widetilde{r})$ matrix, since $H = (\boldsymbol{\varepsilon} \mid M^T)$.

Analogously we have that $G^\perp$ is a $\widetilde{t} \times (m+1)$ matrix, since $\widetilde{t}$ is the dimension of the code $\mathcal{C}^\perp$. Also it follows that $\widetilde{t} \leq t$. On the other hand $H^\perp$ is a parity check matrix of the code $\mathcal{C}^\perp$. Hence $H^\perp$ is an $(m+1-\widetilde{t}) \times (m+1)$ matrix, and thus $M^\perp$ is an $m \times (m+1-\widetilde{t})$ matrix, since $H^\perp = (\boldsymbol{\varepsilon} \mid (M^\perp)^T)$. Note that M and $M^\perp$ have the same size m. As a consequence of Lemma 6, i.e. from $\mathrm{span}(G) = \mathrm{span}(H^\perp)$ the following equality holds: $\widetilde{r} + \widetilde{t} = m + 1$. $\qquad\square$

Recall that the vectors $\boldsymbol{\lambda}^i$ form a suitable set of vectors. Note that Lemma 7 actually restates Lemma 5.

Corollary 1. *Let $\mathcal{M}$ be an MSP program computing Γ, and $\mathcal{M}^\perp$ be an MSP computing the dual access structure $\Gamma^\perp$. Let the code $\mathcal{C}^\perp$ have the parity check matrix $H^\perp = (\boldsymbol{\varepsilon} \mid (M^\perp)^T)$ and the code $\mathcal{C}$ have the parity check matrix $H = (\boldsymbol{\varepsilon} \mid M^T)$. Then for any MSP $\mathcal{M}$ there is an MSP $\mathcal{M}^\perp$ such that $\mathcal{C}$ and $\mathcal{C}^\perp$ are dual.*

4 On Lower Bounds for the Size of MSPs

In earlier works [3, 5, 11, 12] a lower bound for the size of an MSP has been proven. As we pointed out the problem of estimating the size of an MSP is related to many problems in complexity theory such as (symmetric) branching programs, (undirected) contact schemes, formula size as well as with the complexity of some distributed protocols in cryptography.

That is why it should not surprise the reader that the notation in this section differs from the original author's notation. The idea used in [3, 11, 5, 12] 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 minimal sets of the access structure then it must also accept an input that does not contain a minimal set. The latter means that the program does not compute the access structure, since any input accepted by the MSP should contain at least one minimal set.

Beimel *et al.* [3] introduced a notion of a *critical family*, which we will redefine as the notion of *critical set of minimal sets*.

Definition 4. *Let $\Gamma^- = \{X_1, \ldots, X_r\}$ be the set of minimal sets in the access structure Γ. Let $\mathcal{H} \subseteq \Gamma^-$ be a subset of the set of minimal sets. We say that a subset $\mathcal{H} \subseteq \Gamma^-$ is a* critical set of minimal sets *for Γ^-, if every $X_i \in \mathcal{H}$ contains a set $B_i \subseteq X_i$, $|B_i| \geq 2$, such that the following two conditions are satisfied.*

B1. The set B_i uniquely determines X_i in the set $\mathcal{H}$. That is, no other set in $\mathcal{H}$ contains B_i.

B2. For any subset $Y \subseteq B_i$, the set $S_Y = \cup_{X_j \in \mathcal{H}, X_j \cap Y \neq \emptyset}(X_j \backslash Y)$ does not contain any member of Γ^-.

Note that Condition $B2$ requires that S_Y does not contain any minimal set of Γ, not just a minimal set from $\mathcal{H}$. We can rewrite the set S_Y also as

$$S_Y = \cup_{X_j \in \mathcal{H}, X_j \cap Y \neq \emptyset}(X_j \cap Y^c) = (\cup_{X_j \in \mathcal{H}, X_j \cap Y \neq \emptyset}X_j) \cap Y^c$$
$$= (\cup_{X_j \in \mathcal{H}, X_j \cap Y \neq \emptyset}X_j) \backslash Y.$$

Thus we can restate $B2$ as follows:

$B2'$: For any subset $Y \subseteq B_i$, there is no member of Γ^- which is contained in the set $S'_Y = \cup_{X_j \in \mathcal{H}, X_j \cap Y \neq \emptyset}X_j$ and is a subset of Y^c.

Theorem 4. *[3, 11, 5, 12] Let Γ be an access structure, and let $\mathcal{H}$ be a critical set of minimal sets for Γ. Then for every field $\mathbb{F}$, the size of any monotone span program $\mathcal{M}$ computing Γ*

$$size(\mathcal{M}) \geq |\mathcal{H}|.$$

Proof. [sketch]

Let M be the matrix of a monotone span program computing Γ, and let m be the number of rows of M. Any minimal set of $\mathcal{H}$ is accepted by the program. By definition, this means that, for every $X \in \mathcal{H}$, there is some recombination vector $\boldsymbol{\lambda}_X \in \mathbb{F}^m$ such that $M^T \boldsymbol{\lambda}_X = \boldsymbol{\varepsilon}$, where $\boldsymbol{\lambda}_X$ has nonzero coordinates only at rows labelled by variables from X. For any given X there may be several such vectors, we pick one of them and denote it by $\boldsymbol{\lambda}_X$.

Since $\boldsymbol{\lambda}_X$ is taken from $\mathbb{F}^m$, the number of linearly independent vectors among the vectors $\boldsymbol{\lambda}_X$ for $X \in \mathcal{H}$ is a lower bound for m, i.e., for the size of the span program computing Γ. Thus the following lemma concludes the proof. $\square$

Lemma 8. *[3] Let Γ be an access structure, and let $\mathcal{H}$ be a critical set of minimal sets for Γ. Then the recombination vectors $\boldsymbol{\lambda}_X$ for $X \in \mathcal{H}$ are linearly independent.*

Gal [12] derives a superpolynomial (in the number of players) worst case asymptotic lower bound for the size of MSPs, showing that there are access structures Γ, with suitable critical sets of minimal sets $\mathcal{H}$. In [17] the authors argued that there are cases in which asymptotically the number of columns and the number of rows (the size of MSP) are identical. Beimel *et al.* observe also that sizes of a MSP and its dual MSP are equal.

Theorem 5. *[3, 11] For every field $\mathbb{F}$, $size(\mathcal{M}) = size(\mathcal{M}^\perp)$.*

Note that $size(\mathcal{M}) \geq \max(|\mathcal{H}|, |\mathcal{H}^\perp|) \geq \frac{|\mathcal{H}| + |\mathcal{H}^\perp|}{2}$. Now we are ready to prove the main theorem of this section, the improvement of the bound of Beimel *et al.* [3] (see Theorem 4).

Theorem 6. *Let Γ be an access structure and $\Gamma^\perp$ be its dual, let $\mathcal{H}$ be a critical set of minimal sets for Γ and let $\mathcal{H}^\perp$ be a critical set of minimal sets for $\Gamma^\perp$. Then for any field $\mathbb{F}$, the size of any monotone span program $\mathcal{M}$ computing Γ is bounded from below by the sum of the sizes of both critical minimal sets minus one, i.e.,*

$$size(\mathcal{M}) \geq |\mathcal{H}| + |\mathcal{H}^\perp| - 1.$$

Proof. Let M be the matrix of a monotone span program computing the access structure Γ, and let m be the number of rows of M. Let $\Gamma^- = \{X_1, \ldots, X_r\}$ be a set of minimal sets in the access structure Γ and let $\Delta^+ = \{Y_1, \ldots, Y_t\}$ be a set of maximal sets in $\Delta = \Gamma^c$.

For each minimal set X_i consider the corresponding recombination vector $\boldsymbol{\lambda^i} \in \mathbb{F}^m$, so $M^T \boldsymbol{\lambda^i} = \boldsymbol{\varepsilon}$ and $\sup_P(\boldsymbol{\lambda^i}) = X_i$. Recall that the recombination vector $\boldsymbol{\lambda^i}$ corresponds to the vectors $\boldsymbol{\lambda}_X$ in the original proof of Beimel *et al.* [3] (see Theorem 4). For any X_i there may be several such vectors; in that case we pick one of them and denote it by $\boldsymbol{\lambda^i}$. From the proof of Beimel *et al.* (see Lemma 8) it follows that for any critical set of minimal sets $\mathcal{H}$ of Γ^- the corresponding recombination vectors $\boldsymbol{\lambda}$ are linearly independent. Now consider the vectors $\boldsymbol{\lambda^1} - \boldsymbol{\lambda^i}$ for $i = 2, \ldots, r$. It is easy to see that all these vectors are in the kernel of the transposed matrix M^T, i.e. in $\ker(M^T)$. Therefore for any $\mathcal{H}$ we have nullity$(M^T) \geq |\mathcal{H}| - 1$.

For each maximal set Y_i consider a vector $\mathbf{k} \in \mathbb{F}^d$ such that $M_{Y_i} \mathbf{k} = \mathbf{0}$ and $\mathbf{k}_1 = 1$. For any given Y_i there may be several such vectors, again we pick one of them. Define $\widetilde{\boldsymbol{\lambda^i}} = M\mathbf{k}$. Note that $\sup_P(\widetilde{\boldsymbol{\lambda^i}}) = Y_i^c \in (\Gamma^\perp)^-$. From the proof of Lemma 1 as well as from the proof of Lemma 3 we have that $(M^\perp)^T \widetilde{\boldsymbol{\lambda^i}} = \varepsilon^*$. Hence we have the same correspondence between recombination vectors $\widetilde{\boldsymbol{\lambda^i}}$ and sets $Y_i^c \in (\Gamma^\perp)^-$ as we have for recombination vectors $\boldsymbol{\lambda^i}$ and sets $X_i \in \Gamma^-$. Applying again the result of Beimel *et al.* Lemma 8 but for the dual access structure $\Gamma^\perp$ we obtain that for any critical set of minimal sets $\mathcal{H}^\perp$ of $(\Gamma^\perp)^-$ the corresponding recombination vectors $\widetilde{\boldsymbol{\lambda^i}}$ are linearly independent. Now note that by construction the vectors $\widetilde{\boldsymbol{\lambda^i}}$ are in the image of the matrix M, i.e. $\widetilde{\boldsymbol{\lambda^i}} \in \text{im}(M)$. Hence rank$(M) \geq |\mathcal{H}^\perp|$. On the other hand since the row rank is equal to column rank we have rank$(M^T) = \text{rank}(M) \geq |\mathcal{H}^\perp|$.

The last step is to apply the rank and nullity theorem for the transposed matrix M^T:

$$m = \text{rank}(M^T) + \text{nullity}(M^T) \geq |\mathcal{H}^\perp| + |\mathcal{H}| - 1,$$

which completes the proof. $\qquad\square$

Note that the worst case superpolynomial asymptotic estimation for the size of MSPs due to Gal [12] does not change because of this relation.

Revisiting the proof of Theorem 6 we notice that nullity$(M^T) = \bar{r} - 1$ and rank$(M^T) = \bar{t}$. Hence we have actually three different proofs of the fact that $m = \text{rank}(M^T) + \text{nullity}(M^T) = \bar{r} + \bar{t} - 1$ (see also Lemma 5 and Lemma 7). Now observe that $|\mathcal{H}| \leq \bar{r}$ and $|\mathcal{H}^\perp| \leq \bar{t}$ give the lower bound (Theorem 6)

and that $\bar{r} \leq r$ and $\bar{t} \leq t$ give the upper bound (Theorem 3). Note that the lower bound is achieved if there exist critical minimal and maximal sets with exactly (the maximum possible number) $\bar{r}$ and $\bar{t}$ elements. However, how one can efficiently build an MSP computing Γ with the smallest size remains still an open question.

5 Conclusions

In this paper we have shown an upper and improve the lower bound for the size of monotone span programs. Next we extend the result of van Dijk showing that for any MSP there exists a dual MSP such that the corresponding codes are dual.

6 Acknowledgements

References

1. E. Allender, R. Beals, M. Ogihara, The Complexity of Matrix Rank and Feasible Systems of Linear Equations, ACM STOC'96, 1996, pp. 161-167.
2. A. Beimel, Secure Schemes for Secret Sharing and Key Distribution, *Ph.D. Thesis*, Technion, 1996.
3. A. Beimel, A. Gal, M. Paterson, Lower Bounds for Monotone Span Programs, *Computational Complexity*, 6, 1996/1997, pp. 29-45.
4. A. Beimel, E. Weinreb, Separating the Power of Monotone Span Programs over Different Fields, *FOCS'03*, 2003, pp. 428-437.
5. L. Babai, A. Gal, A. Wigderson, Superpolynomial Lower Bounds for Monotone Span Programs, *Combinatorica* 19 (3), 1999, pp. 301-319.
6. E. Brickell, Some Ideal Secret Sharing Schemes, *J. of Comb. Math. and Comb. Computing* **9**, 1989, pp. 105-113.
7. G. Buntrock, C. Damm, H. Hertrampf, C. Meinel, Structure and Importance of the Logspace-mod Class, *Math. Systems Theory* 25, 1992, pp. 223-237.
8. R. Cramer, I. Damgard, U. Maurer, General Secure Multi-Party Computation from any Linear Secret Sharing Scheme, *EUROCRYPT'2000*, Springer-Verlag LNCS 1807, 2000, pp. 316-334.
9. R. Cramer, S. Fehr, Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups, *CRYPTO 2002*, Springer-Verlag LNCS 2442, 2002, pp. 272-287.
10. M. van Dijk, Secret Key Sharing and Secret Key Generation, *Ph.D. Thesis*, 1997, TU Eindhoven.
11. A. Gal, Combinatorial Methods in Boolean Functions Complexity, *Ph.D. Thesis*, Chicago, Illinois, 1995.
12. A. Gal, A Characterization of Span Program Size and Improved Lower Bounds for Monotone Span Programs, *Computational Complexity*, Vol. 10, No. 4, 2001, pp. 277-296.

13. A. Gal, P. Pudlak, Monotone Complexity and the Rank of Matrices, *Inform. Proc. Lett.* 87, 2003, pp. 321-326.

14. M. Karchmer, A. Wigderson, On Span Programs, *Proc. of 8-th Annual Structure in Complexity Theory Conference*, 1993, IEEE Computer Society Press, pp. 102-111.

15. V. Nikov, S. Nikova, B. Preneel, J. Vandewalle, Applying General Access Structure to Metering Schemes, *WCC* 2003, *Cryptology ePrint Archive*: Report 2002/102.

16. V. Nikov, S. Nikova, B. Preneel, Upper Bound for the Size of Monotone Span Programs, *ISIT 2003*, 2003, pp. 284.

17. P. Pudlak, J. Sgall, Algebraic Models of Computations and Interpolation for Algebraic Proof Systems, *Proof Complexity and Feasible Arithmetic*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 39, 1998, pp. 279-295.

18. A. Shamir, How to Share a Secret, *Commun. ACM*, 22, 1979, pp. 612-613.