Concrete Multiplicative Complexity of Symmetric Functions

Joan Boyar[1⋆] and René Peralta[2⋆⋆]

[1] Dept. of Math. and Computer Science, University of Southern Denmark
joan@imada.sdu.dk
[2] Security Division
Information Technology Laboratory, NIST
rene.peralta@nist.gov

Abstract. The multiplicative complexity of a Boolean function f is defined as the minimum number of binary conjunction (AND) gates required to construct a circuit representing f, when only exclusive-or, conjunction and negation gates may be used. This article explores in detail the multiplicative complexity of symmetric Boolean functions. New techniques that allow such exploration are introduced. They are powerful enough to give exact multiplicative complexities for several classes of symmetric functions. In particular, the multiplicative complexity of computing the Hamming weight of n bits is shown to be exactly $n-H^{\mathbb{N}}(n)$, where $H^{\mathbb{N}}(n)$ is the Hamming weight of the binary representation of n. We also show a close relationship between the complexity of symmetric functions and fractals derived from the parity of binomial coefficients.

1 Introduction

Much research in circuit complexity is devoted to the following problem: Given a Boolean function and a supply of gate types, construct a circuit which computes the function and is optimal according to some criteria. It seems to be very difficult in general to obtain exact bounds for specific functions. The *multiplicative complexity* $c_{\wedge}(f)$ of a Boolean function f is the number of conjunctions necessary and sufficient to implement a circuit which computes f over the basis $(\wedge, \oplus, 1)$ (alternatively, the number of multiplications necessary and sufficient to calculate a function over GF_2 via a straight-line program).

Our initial motivation for studying multiplicative complexity came from cryptography. Many cryptographic protocols involve proving predicates about a string X that is available in *committed* form only, i.e., the bits of X are individually encrypted using a *bit-commitment scheme*. In [3] a construction is given

⋆ Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT), and by the Danish Natural Science Research Council (SNF).

⋆⋆ Part of this work was done at the Computer Science Department, Yale University, prior to this author joining NIST. While at Yale, this work was partially supported by NSF grant CCR-0081823.

for a non-interactive cryptographic proof of an arbitrary predicate F on X. The predicate F is defined by a verification circuit C containing AND, NOT, and XOR gates only. The length of these *discreet proofs* is linear in the number of AND gates in C and is unaffected by the number of NOT or XOR gates. Another promising area of application of these results is in the communication complexity of secure multi-party computation. In general, for these protocols, multiplications require communication, but linear operations do not. This holds for very different paradigms for building protocols, those based on secret sharing were introduced in [2, 7] and those based on threshold homomorphic encryption were introduced in [6]. For more recent results, see [9].

We focus on symmetric functions, which are functions dependent only on the Hamming weight $\overrightarrow{H}(\mathbf{x})$ of the input $\mathbf{x} \in GF_2^n$. Obtaining tight bounds is important because symmetric functions can be building blocks for arithmetic circuits, some of which involve recursive use of simple symmetric functions. Sub-optimal implementations of the latter, even by an additive constant factor, translate into multiplicative extra costs when building arithmetic circuits. In cryptographic applications, whether or not a circuit is of practical use often depends on constant multiplicative factors in the number of AND gates used.

The study of multiplicative complexity may prove useful in obtaining upper bounds on the computational complexity of functions. If a function f has multiplicative complexity $O(\log(n))$, then, for all $\mathbf{x}$ in the domain of f, an element of the pre-image of $y = f(\mathbf{x})$ can be found in polynomial-time as follows: Guess the values of inputs to the AND gates in a circuit for f. This reduces the circuit to a collection of linear circuits. Now find an $\mathbf{x}$ such that $y = f(\mathbf{x})$ using Gaussian elimination over GF_2. This shows that, one-way functions, if they exist, have super-logarithmic multiplicative complexity. On the other hand, low multiplicative complexity circuits may lead to better algorithms for inverting functions of importance in cryptology.

Previous work. Multiplicative complexity has been investigated previously by Aleksanyan [1], Schnorr [14], and Mirwald and Schnorr [11]. Their work was exclusively concerned with quadratic forms. Multiplicative complexity has more often been used to refer to more general algebraic computations. This subject has an extensive history (see, for example, [5]), since multiplication is often the dominant operation in this context.

Very little is known about multiplicative complexity of specific functions. In this paper we concentrate on the concrete (as opposed to asymptotic) multiplicative complexity of symmetric functions. In an earlier paper [4], we showed the following results:

- A general upper bound of $n+3\sqrt{n}$ for any symmetric function f. This establishes a separation between Boolean and multiplicative complexity for sym-

metric functions. Paul [12] and Stockmeyer [15] have shown lower bounds of the form $2.5n - O(1)$ for the Boolean complexity of infinite families of symmetric functions;

– Let $\Sigma^n_\cdot$ be the set of symmetric predicates on n bits. We showed an upper bound of $2n - \log_2 n$ for the complexity $c_\wedge(\Sigma^n_\cdot)$ of simultaneously computing all symmetric functions on n bits (the asymptotic result $c_\wedge(\Sigma^n_\cdot) = O(n)$ was obtained earlier by Mihaĭljuk [10]).

Our results. Several new upper and lower bounds on the multiplicative complexity of symmetric functions are obtained. In particular, it is shown that the multiplicative complexity of computing the Hamming weight is exactly $n - H^{\mathbb{N}}(n)$, where $H^{\mathbb{N}}(n)$ is the Hamming weight of the binary representation of n. This is a rather surprising result, given the sparsity of exact computational complexity bounds known.

A new technique, using a normal form for $(\oplus, 1, \wedge)$ circuits and elementary linear algebra, is used to show that any non-linear symmetric function on n variables has multiplicative complexity at least $\lfloor \frac{n}{2} \rfloor$. Properties of binomial coefficients are shown to yield the following lower bounds for the counting (exactly-k) and threshold-k functions on n variables:

$$c_\wedge(E^n_k) \geq \max\{k-1, n-k-1, 2^{\lfloor \log_2 n \rfloor} - 2, l_{n,k} - 1\}$$
$$c_\wedge(T^n_k) \geq \max\{k-1, n-k, 2^{\lfloor \log_2 n \rfloor} - 1, l_{n-1,k-1}\}$$

where $l_{n,k}$ is the bitwise OR of $n-k$ and k. Tighter bounds for several families of symmetric functions are obtained by considering the multiplicative complexity of such functions when restricted to hyperplanes in GF^n_2. In particular, this technique yields the exact complexities of the elementary symmetric functions $\Sigma^n_2, \Sigma^n_3, \Sigma^n_{n-1}, \Sigma^n_{n-2}, \Sigma^n_{n-3}$. Yet another application of hyperplane restrictions yields new general lower bounds for infinite subclasses of symmetric functions. Intriguingly, these subclasses are defined by fractals on the Cartesian plane.

More constructively, general techniques are developed for proving upper bounds for elementary symmetric functions. These, plus properties of Pascal's triangle modulo 2 (known in the fractals literature as Sierpinski's gasket), are used to prove upper bounds for the counting functions, $E^n_k(\mathbf{x})$, and the threshold functions, $T^n_k(\mathbf{x})$. These general techniques are shown to give many tight results. In addition, a general upper bound on the threshold-k functions, T^n_k, is found: $c_\wedge(T^n_k) \leq n - H^{\mathbb{N}}(n) + \lceil \log_2(n+1) \rceil - 1$ for all $k \geq 1$.

In the following sections, and due to space constraints, most proofs will be omitted.

2 Some simple observations and a normal form

Each Boolean function f on n variables has a unique representation as a multilinear (i.e. square-free) polynomial over GF_2. Since $x^i = x$ over GF_2, we assume throughout the following that all polynomials are multilinear. By the "degree of f", we will mean the degree of its unique representing polynomial. It is known that a Boolean function of degree d has multiplicative complexity at least $d - 1$. This we call the *degree lower bound*.

We say that a circuit is optimal for f if it has $c_\wedge(f)$ AND gates. Since $y \wedge (x \oplus 1) = (y \wedge x) \oplus y$, optimal circuits need not have more than one negation. If present, we may assume this negation is the last gate in the circuit. It is not hard to see that optimal circuits for a Boolean function $f(\mathbf{x})$ require a negation if and only if $f(\mathbf{0}) = 1$, which holds if and only if the polynomial of f has a constant term. Thus we may divide Boolean functions into "positive" functions (those for which $f(\mathbf{0}) = 0$) and "negative" functions. There is a bijection $\sigma(f) = f \oplus 1$ between positive and negative functions. Since the bijection preserves multiplicative complexity, we may restrict our study of multiplicative complexity to functions over the basis $(\oplus, \wedge)$. For technical reasons, and without affecting the multiplicative complexity of functions, we allow $\oplus$ gates to contain any number of inputs (at least one). AND gates, though, are restricted to fan-in exactly 2. We call a gate "internal" if its output is not the output to the circuit. We say a circuit is in *Layered Normal Form* (LNF) if i) all inputs go only to $\oplus$ gates; and ii) outputs of all internal $\oplus$ gates are inputs only to $\wedge$ gates. It is not hard to see that all positive functions have optimal circuits in Layered Normal Form.

Logical expressions over the basis $(\wedge, \oplus)$ correspond to arithmetic expressions over GF_2. We will use the latter notation for the most part of this paper: $a \oplus b, a \wedge b, \bar{a}$ will be written $a \oplus b, ab, a \oplus 1$, respectively. The kth elementary symmetric function on n variables $x_1, x_2, \ldots, x_n$ is defined by

$$\Sigma_k^n(x_1, x_2, \ldots, x_n) = \bigoplus\nolimits_{S \subseteq \{1,\ldots,n\}, |S|=k} \prod_{i \in S} x_i \qquad (1 \leq k \leq n).$$

For readability we will also use the alternative notations $\Sigma_k^n(\mathbf{x})$ or simply Σ_k^n. It will prove convenient as well to define $\Sigma_0^n = 1$.

A classical result states that every symmetric function can be represented as a sum of elementary symmetric functions (see [16]). Consider, for example, the MAJORITY function on three variables (i.e. the threshold function $T_2^3 = \Sigma_2^3$). $\Sigma_2^3(x_1, x_2, x_3) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_2)(x_1 \oplus x_3) \oplus x_1$. The last equality establishes $c_\wedge(T_2^3) = 1$, and also serves to show that the algebraic manipulations necessary to obtain optimal circuits may not be obvious.

The following lemmas appear in [4]:

Lemma 1. *Represent the positive integer k as a sum of powers of 2: $k = 2^{i_0} + 2^{i_1} + \ldots + 2^{i_j}$. Each i is a position of a non-zero bit in the binary representation of k. Then for any $n \geq k$, $\Sigma_k^n = \Sigma_{2^{i_0}}^n \Sigma_{2^{i_1}}^n \ldots \Sigma_{2^{i_j}}^n$.*

Lemma 2. *Let $\mathbf{y} = y_k y_{k-1} \ldots y_0$ be the Hamming weight, in binary representation, of the n-bit string $\mathbf{x}$. Then $y_i = \Sigma_{2^i}^n(\mathbf{x})$ for $i = 0, \ldots, k$.*[3]

These show, for example, that $\Sigma_{11}^n = \Sigma_8^n \Sigma_2^n \Sigma_1^n$ for $n \geq 11$, and the Hamming weight of a 10-bit string $\mathbf{x}$ is a string of length 4 whose bits are $\Sigma_8^{10}(\mathbf{x})$, $\Sigma_4^{10}(\mathbf{x})$, $\Sigma_2^{10}(\mathbf{x})$, and $\Sigma_1^{10}(\mathbf{x})$. Finally, we observe that if $g : GF_2^k \rightarrow GF_2$ is derived from $f : GF_2^n \rightarrow GF_2$ by fixing the values of $n - k$ variables of f, then $c_\wedge(g) \leq c_\wedge(f)$. We call g a *restriction* of f.

3 A tight lower bound on the multiplicative complexity of symmetric functions

Given a Boolean function f over GF_2^n and a subset S of $\{x_1, \ldots, x_n\}$, we denote by $f_{\bar{S}}$ the function obtained from f by complementing the inputs in S. If $f_{\bar{S}} = f$, we say S is *complementable*. We say S is "proper" if $0 < |S| < n$.

Lemma 3. *If a Boolean function f over GF_2^n has multiplicative complexity less than $\lfloor \frac{n-1}{2} \rfloor$, then it has a proper complementable set.*

Proof. Consider an optimal LNF circuit for f. If the circuit has at most $\lfloor \frac{n-1}{2} \rfloor - 1$ AND gates, the number of $\oplus$ gates is at most $k = 2(\lfloor \frac{n-1}{2} \rfloor - 1) + 1 \leq n - 2$ (recall that a circuit in LNF form may have at most one $\oplus$ gate which is not the input to an $\wedge$ gate). Label these gates $\gamma_1, \ldots, \gamma_k$. Define an $n \times k$ matrix $A = (a_{ij})$ over GF_2 as follows: $a_{ij} = 1$ iff x_i is an input to γ_j. Rows of the matrix correspond to inputs of the circuit. Columns correspond to $\oplus$ gates. Since $rank(A) \leq k \leq n - 2$, there is a subset S (with $0 < |S| \leq n - 1$) of the rows whose sum over GF_2^k is $\mathbf{0}$. Since in a LNF circuit all inputs go only to $\oplus$ gates, and each $\oplus$ gate has an even number of inputs from S, S is a complementable set of inputs. □

For a symmetric function f, if a proper set S of cardinality k is complementable, then *every* set of cardinality k is complementable, including the sets $\{x_1, \ldots, x_k\}$ and $\{x_2, \ldots, x_{k+1}\}$. Hence, $\{x_1, x_{k+1}\}$ is also complementable, so any two inputs are complementable. Thus if the Hamming weights of $\mathbf{x}$ and $\mathbf{y}$ have the same parity, then $f(\mathbf{x}) = f(\mathbf{y})$, so f is linear. We have shown

[3] See also [13].

Lemma 4. *If a symmetric Boolean function f has a proper complementable set S, then f must be linear (i.e. $c_\wedge(f) = 0$).*

A lower bound of $\lfloor \frac{n-1}{2} \rfloor$ for non-linear symmetric functions immediately follows. In the full paper, we prove the slightly stronger result:

Theorem 1. *The multiplicative complexity of an $n-$variate non-linear symmetric function is at least $\lfloor \frac{n}{2} \rfloor$.*

4 Hyperplane restrictions yield fractal lower bounds

We now describe a new technique which uses the degree lower bound, but often achieves stronger lower bounds. A plane E in GF_2^n can be specified by an equation $\bigoplus_{i \in I_E} x_i = 0$, where $I_E \subseteq \{1, \ldots, n\}$. For notational simplicity, if the index set is empty, we define $\bigoplus_{i \in \phi} x_i = 0$. Given a Boolean function f on n-bits, we denote the restriction of f to the plane E by $f_{\downarrow E}$. Letting $t = Max(I_E)$, we view $f_{\downarrow E}$ as a function on $n-1$ variables obtained by substituting $\bigoplus_{i \in I_E - \{t\}} x_i$ for x_t in the polynomial for f. There are many ways to obtain a circuit for $f_{\downarrow E}$ from a circuit for f. For C in Layered Normal Form, $C_{\downarrow E}$ will denote the circuit constructed by replacing x_t by all of the other variables in I_E, removing pairs of identical inputs to XOR gates, and repeatedly removing XOR gates with no inputs and unnecessary AND gates. $C_{\downarrow E}$ will be in Layered Normal Form. We now proceed to prove lower bounds by choosing planes which will decrease the number of AND gates in a circuit without decreasing the degree of the function which is computed. The degree lower bound is then applied to the function resulting from the restriction.

Lemma 5. *Suppose f is an $n-$variate function of degree $k > 1$. If $c_\wedge(f) = k - 1 + e$, where $e \geq 0$, then there exist $u \leq e + 1$ planes $E_1, E_2, ..., E_u$ such that the degree of $(\ldots((f_{\downarrow E_1})_{\downarrow E_2})\ldots)_{\downarrow E_u}$ is at most $k - 1$.*

Corollary 1. *Suppose f is an $n-$variate symmetric function of degree $k > 1$. If $c_\wedge(f) = k - 1$, then $deg(f_{\downarrow E}) \leq k - 1$ for at least two distinct planes E_1, E_2 where E_1 can be specified by $x_n = \bigoplus_{i=1}^{t_1} x_i \quad (t_1 < n)$, and E_2 can be specified using an equation with at most $n - 2$ terms in the sum.*

The technique of hyperplane restrictions yields lower bounds on multiplicative complexity which are better than the degree lower bound for many symmetric functions, including all with degree less than $n - 1$. We next state some of these bounds. In section 6, the bound given by the following theorem is shown to be tight for Σ_{n-2}^n and Σ_{n-3}^n.

Theorem 2. *Let f be a $n-$variate symmetric function of degree m, with $1 < m < n-1$. Then $c_\wedge(f) \geq m$.*

The proof of Theorem 2 involves one hyperplane restriction. Lemma 5 can be used to prove tighter bounds using successive hyperplane restrictions under certain combinatorial constraints.

Theorem 3. *Let f be a $n-$variate symmetric function of degree m. Suppose $1 < m \leq n-2$ and $n > 4$. Then, if $\binom{n-4}{m-2}$ is even, $\binom{n-3}{m-1}$ is even, and $\binom{n-2}{m}$ is odd, then $c_\wedge(f) \geq m+1$.*

Theorem 4. *Let f be a $n-$variate symmetric function of degree m. If $\binom{n-6}{m-3}$, $\binom{n-5}{m-2}$, and $\binom{n-4}{m-1}$ are even, while $\binom{n-3}{m}$ is odd, then $c_\wedge(f) \geq m+2$.*

Theorem 3 gives the nontrivial lower bound $c_\wedge(\Sigma_4^8) \geq 5$. The set of points in the plane that satisfy the conditions of either Theorem 3 or Theorem 4 form fractals. Figure 1 plots these points for Theorem 3. The hyperplane restriction technique is a general tool for relating combinatorial constraints to multiplicative complexity. The combinatorial constraints thus derived seem to always yield fractals. An interesting question is whether this is solely a result of the bounding technique or the exact complexity of the elementary symmetric functions is in fact fractal in nature.

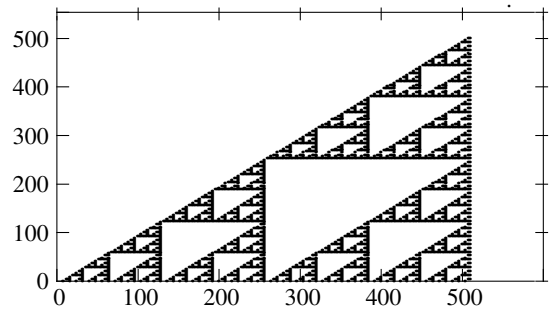

Fig. 1. Points (n,m) for which $c_\wedge(\Sigma_m^n) \geq m+1$, $m < n < 512$.

5 The exact multiplicative complexity of the Hamming weight function

The result of computing a symmetric function on some inputs is determined completely by the Hamming weight of those inputs. In this section, we investigate the multiplicative complexity of computing the Hamming weight. Let

$\overrightarrow{H}(\mathbf{x})$ denote the binary representation of the Hamming weight of a bit string $\mathbf{x} \in GF_2^n$. $\overrightarrow{H}(\mathbf{x})$ has fixed length $\lceil \log_2(n+1) \rceil$ and may contain leading zeros. The function $\overrightarrow{H}()$ will be denoted by H^n when the parameter n needs to be explicitly stated. Let $H^{\mathbb{N}}(n)$ denote the Hamming weight of the binary representation of the integer n. Theorem 8 in [4] can be seen to give the result that $c_\wedge(H^n) \leq n - H^{\mathbb{N}}(n)$. Here we prove a matching lower bound. It will prove useful to define the Hamming weight of the empty string λ to be 0, i.e. $\overrightarrow{H}(\lambda) = H^{\mathbb{N}}(0) = 0$.

Theorem 5. *$c_\wedge(H^n) = n - H^{\mathbb{N}}(n)$, for all $n \geq 1$.*

Proof. We begin supposing that $\mathbf{x}$ is a bit string of length 2^k. By Lemma 2, the $k+1$st bit of $\overrightarrow{H}(\mathbf{x})$ is $\Sigma_{2^k}^{2^k}(\mathbf{x})$, which is a polynomial of degree 2^k. Thus, by the degree lower bound, $c_\wedge(H^{2^k}) \geq 2^k - H^{\mathbb{N}}(2^k) = 2^k - 1$ for all $k \geq 0$. This matches the upper bound, and these known bounds will now be used to prove the lower bound for lengths which are not powers of 2. For notational brevity, we will denote $c_\wedge(H^n)$ by h_n. Our proof is by induction on k with base $k = 1$. Let $k > 1$ and assume the theorem holds for all $n' \leq 2^{k-1}$. Let $n = 2^k - i$ for some integer $1 \leq i < 2^{k-1}$. Then $n + (i-1) = 2^k - 1$. Note that if $0 \leq a, b, k$ and $n = 2^k - 1 = a + b$, then $H^{\mathbb{N}}(n) = H^{\mathbb{N}}(a) + H^{\mathbb{N}}(b)$. Thus, $k - H^{\mathbb{N}}(i-1) = H^{\mathbb{N}}(n)$. We design a circuit for the Hamming weight of a string $\mathbf{x}$ of length $2^k = n + (i-1) + 1$ as follows. We split $\mathbf{x}$ into three strings $\mathbf{u}, \mathbf{v}, c$ of lengths $n, i-1$, and 1, respectively. We use optimal circuits to compute $\overrightarrow{H}(\mathbf{u})$ and $\overrightarrow{H}(\mathbf{v})$. Note that the longest of these two strings is $\overrightarrow{H}(\mathbf{u})$, which has length k. Then we use the standard addition circuit with carry-in c to compute $c + \overrightarrow{H}(\mathbf{u}) + \overrightarrow{H}(\mathbf{v})$ (which uses k multiplications since a full adder uses just one multiplication for T_2^3). The result is $\overrightarrow{H}(\mathbf{x})$. By the inductive hypothesis, the circuit for $\overrightarrow{H}(\mathbf{v})$ contains $h_{i-1} = (i-1) - H^{\mathbb{N}}(i-1)$ multiplications. Thus the circuit for $\overrightarrow{H}(\mathbf{x})$ contains $h_n + (i-1) - H^{\mathbb{N}}(i-1) + k$ multiplications. Since $c_\wedge(H^{2^k}) \geq 2^k - 1$, this quantity must be at least $2^k - 1$, i.e.

$$h_n + (i-1) - H^{\mathbb{N}}(i-1) + k \geq 2^k - 1.$$

Substituting $H^{\mathbb{N}}(n)$ for $k - H^{\mathbb{N}}(i-1)$, n for $2^k - i$, and rearranging terms, we obtain $h_n \geq n - H^{\mathbb{N}}(n)$. This proves the theorem since the lower bound matches the upper bound from [4]. □

Truncated Hamming weight. Let H_r^n be the function which computes the r low-order bits of the Hamming weight of a vector of length $n \geq 2^{r-1}$. The complexity of this function is 0 when $r = 1$ and $n - H^{\mathbb{N}}(n)$ when $n \leq 2^r - 1$. A recursive construction (see the full paper) yields the following results:

Lemma 6. *For $j \geq r \geq 1$, we have $c_\wedge(H_r^{2^j-1}) \leq \left(\frac{2^{r-1}-1}{2^{r-1}}\right) 2^j - r + 1$.*

Lemma 7. *Let $r \geq 1$ and $n \geq 2^r$. Let $\gamma = n \bmod 2^r$. Then, $c_\wedge(H_r^n) \leq \left(\frac{2^{r-1}-1}{2^{r-1}}\right)(n-\gamma) + \gamma - H^{\mathbb{N}}(\gamma)$.*

6 Building blocks

We now discuss subclasses of symmetric functions. The idea is to bound, as tightly as possible, the multiplicative complexity of classes of functions which can be used to construct arbitrary symmetric functions. We focus on three classes of functions:

- The elementary symmetric functions $\Sigma_k^n(\mathbf{x})$.
- The "counting" function $E_k^n(\mathbf{x})$, which is 1 if and only if the Hamming weight of $\mathbf{x}$ is k.
- The "threshold" function $T_k^n(\mathbf{x})$, which is 1 if and only if the Hamming weight of $\mathbf{x}$ is k or more.

First, we consider the elementary symmetric functions, Σ_k^n. Let $c_\wedge(f_1, \ldots, f_k)$ denote the multiplicative complexity of simultaneously computing $f_1, \ldots, f_k$. An immediate corollary of Lemma 7 is the following:

Corollary 2. *Let $r \geq 1$, $n \geq 2^{r-1}$, and $\gamma = (n \bmod 2^r)$. Then*

$$c_\wedge(\Sigma_{2^0}^n, \ldots, \Sigma_{2^{r-1}}^n) \leq \left(\frac{2^{r-1}-1}{2^{r-1}}\right)(n-\gamma) + \gamma - H^{\mathbb{N}}(\gamma).$$

By Lemma 1, the value of $\Sigma_k^n(\mathbf{x})$ is simply the GF_2 product of at most $H^{\mathbb{N}}(k)$ of the low-order $\lceil \log_2(k+1) \rceil$ bits of the Hamming weight of $\mathbf{x}$. Therefore, Corollary 2 yields a general upper bound for Σ_k^n and a less general result:

Theorem 6. *Let $n \geq k \geq 1$, and $r = \lceil \log_2(k+1) \rceil$. Let $\gamma = (n \bmod 2^r)$.* $c_\wedge(\Sigma_k^n) \leq \left(\frac{2^{r-1}-1}{2^{r-1}}\right)(n-\gamma) + \gamma - H^{\mathbb{N}}(\gamma) + H^{\mathbb{N}}(k) - 1$.

Corollary 3. *For $n \geq 4$ and $n' = n \bmod 4$,*
$c_\wedge(\Sigma_4^n) \leq c_\wedge(\Sigma_2^n, \Sigma_4^n) \leq \frac{3}{4}n' + \lfloor \frac{n \bmod 4}{2} \rfloor$.

For example, Corollary 3 yields the result $c_\wedge(\Sigma_4^5) = 3$, though this upper bound also follows from Theorem 5, since $\Sigma_4^5(\mathbf{x})$ is the high-order bit of $\overrightarrow{H}(\mathbf{x})$. We now state several results for the complexity of Σ_k^n for various specific values of k.

Theorem 7. *$c_\wedge(\Sigma_2^n) = \lfloor \frac{n}{2} \rfloor$ and $c_\wedge(\Sigma_3^n) = \lceil \frac{n}{2} \rceil$.*

Lemma 8. *If m is odd and $1 \leq m \leq n$, then $\Sigma_m^n = \Sigma_{m-1}^{n-1}\Sigma_1^n$ and therefore $c_\wedge(\Sigma_m^n) \leq c_\wedge(\Sigma_{m-1}^{n-1}) + 1$.*

Lemma 9. *$c_\wedge(\Sigma_{n-1}^n) = n-2$, $c_\wedge(\Sigma_{n-2}^n) = n-2$ for $n > 3$, and $c_\wedge(\Sigma_{n-3}^n) = n-3$ for $n > 4$.*

We now turn to the counting and threshold functions, $E_k^n(\mathbf{x})$ and $T_k^n(\mathbf{x})$. The degree of $E_k^n = a_0\Sigma_0^n \oplus \ldots \oplus a_n\Sigma_n^n$ is the largest i such that a_i is non-zero. It is clear that $a_i = 0$ for $i < k$. It turns out there is a simple formula for the remaining a_i.

Lemma 10. *$E_k^n = \bigoplus_{i=k}^n a_i\Sigma_i^n$, where $a_i = \binom{i}{k} \bmod 2$.*

Thus, the expansions of the exactly-k functions can be "read off" rows of Sierpinsky's gasket. For example the expansion of E_6^{13} corresponds to the sixth column (1 1 0 0 0 0 0 0) of the fractal: $E_6^{13} = \Sigma_6^{13} \oplus \Sigma_7^{13}$. Now, $\Sigma_6^{13} \oplus \Sigma_7^{13} = \Sigma_4^{13} \cdot \Sigma_2^{13} \cdot (1 \oplus \Sigma_1^{13})$. Thus $c_\wedge(E_6^{13}) \leq c_\wedge(\Sigma_4^{13}, \Sigma_2^{13}) + 2$. By Corollary 3, $c_\wedge(\Sigma_4^{13}, \Sigma_2^{13}) \leq 9$. Therefore $c_\wedge(E_6^{13}) \leq 11$. This is quite remarkable given the general upper bound of $13 + 3\sqrt{13} > 23$ from [4] (or if one considers that the associated polynomial has over 18 thousand multiplications).

A similar lemma holds for the threshold functions since T_k^n can be expressed recursively using $T_k^n = x_n E_{k-1}^{n-1} \oplus T_k^{n-1}$, which says that at least k of $x_1, \ldots, x_n$ are ones if and only if at least k out of $x_1, \ldots, x_{n-1}$ are ones or (exclusive) x_n is one and exactly $k-1$ out of $x_1, \ldots, x_{n-1}$ are ones. This leads to the following characterization of the expansion of T_k^n based on Sierpinski's gasket.

Lemma 11. *$T_k^n = \bigoplus_{i=k}^n b_i\Sigma_i^n$ where $b_i = \binom{i-1}{k-1} \pmod 2$.*

Since $E_k^n(\mathbf{x}) = E_{n-k}^n(\bar{\mathbf{x}})$, we have $c_\wedge(E_k^n) = c_\wedge(E_{n-k}^n)$ for $0 \leq k \leq n$. Then the degree lower bound yields $c_\wedge(E_k^n) \geq \max\{k-1, n-k-1\}$. Similarly, since $T_k^n(\mathbf{x}) = 1 \oplus T_{n-k+1}^n(\bar{\mathbf{x}})$, we have $c_\wedge(T_k^n) = c_\wedge(T_{n-k+1}^n)$ for $1 \leq k \leq n$, and the degree lower bound yields $c_\wedge(T_k^n) \geq \max\{k-1, n-k\}$. Since $T_n^n = \Sigma_n^n$, we have $c_\wedge(T_1^n) = c_\wedge(T_n^n) = c_\wedge(\Sigma_n^n) = n-1$.

As mentioned above, the degree of E_k^n (or T_k^n) will be the largest value j such that the expansion of E_k^n (T_k^n) contains the term Σ_j^n. In the case of E_k^n this will be the largest $k \leq j \leq n$ such that the binomial coefficient $a_j = \binom{j}{k}$ is odd, and in the case of T_k^n this will be the largest $k \leq j \leq n$ such that $b_j = \binom{j-1}{k-1}$ is odd. Thus, the degree of T_k^n is one more than the degree of E_{k-1}^{n-1}. Given this relation, we will only consider the degree of E_k^n.

A theorem by Kummer [8] shows that the binomial coefficient $\binom{j}{k}$ is odd if and only if $k \sqsubseteq j$, where the notation $k \sqsubseteq j$ means that if the binary representations of k and j are $k_s k_{s-1} \ldots k_1$ and $j_s j_{s-1} \ldots j_1$, respectively, then for each i

such that $k_i = 1$, it also the case that $j_i = 1$. This can be used to give the following degree lower bounds on the multiplicative complexity of the exactly-k and threshold-k functions:

Theorem 8. *$c_\wedge(E_k^n) \geq \max\{k-1, n-k-1, 2^{\lfloor \log_2 n \rfloor} - 2, l_{n,k} - 1\}$ and $c_\wedge(T_k^n) \geq \max\{k-1, n-k, 2^{\lfloor \log_2 n \rfloor} - 1, l_{n-1,k-1}\}$, where $l_{n,k}$ is the bitwise OR of $n-k$ and k.*

We now turn to upper bounds. We develop new techniques for producing circuits with few AND gates. We refer to a set of Boolean functions on n variables as a *complete basis* if any symmetric function can be expressed as a linear combination of these functions. Examples of complete bases are $\{\Sigma_i^n \mid 0 \leq i \leq n\}$, and $\{E_i^n \mid 0 \leq i \leq n\}$. Define $A_m^q = \bigoplus_{i=m}^q \Sigma_i^n$ for $m \leq q \leq n$.[4] Then $\Sigma_n^n = A_n^n$ and $\Sigma_m^n = A_m^n \oplus A_{m+1}^n$ for $m < n$. Therefore, $\{A_i^n \mid 0 \leq i \leq n\}$ is complete basis. We will prove upper bounds on the multiplicative complexity of several classes of functions by constructing circuits for functions in the class A_i^q with $0 \leq i \leq q \leq n$.

Lemma 12. *Let $r \geq 1$ and $2^r - 1 \leq n$. Assume the values of $\Sigma_{2^i}^n$ are known for $i = 0, \ldots, r-1$. Then $A_0^{2^r-1}$ can be computed using $r-1$ additional AND gates.*

Corollary 4. *Let $r \geq 1$ and $2^r - 1 \leq n$. Assume the values of $\Sigma_{2^i}^n$ are known for $i = 0, \ldots, r-1$. Then the functions $A_0^{2^s-1}$ $(0 \leq s \leq r)$ can be simultaneously computed using at most $r-1$ additional AND gates.*

We view the set of functions $\{A_0^{2^s-1} \mid 0 \leq s \leq r\} \cup \{\Sigma_{2^i}^n \mid i = 0, \ldots, r\}$ as a basis. The number of AND gates sufficient to compute any linear combination of functions in this basis is no more than $c_\wedge(H_r^n) + r - 1$.[5] The following corollary allows us to expand the basis.

Corollary 5. *Let $r \geq 0$ and $2^r - 1 \leq n$. Assume the values of $\Sigma_{2^i}^n$ are known for $i = 0, \ldots, r-1$. Then the basis $\{A_0^{2^s-1} \mid 0 \leq s \leq r\} \cup \{A_m^{2^s-1} \mid 0 \leq s \leq r, m = 2^q, q < s\} \cup \{A_m^{2^s-1} \mid 0 \leq s \leq r, m = 2^q + 1, q < s\}$ can be computed using $r-1$ additional AND gates.*

Examples of results obtained using this basis are:

Lemma 13. *Any symmetric function on 7 inputs has multiplicative complexity at most 8.*

Corollary 6. *Let $r \geq 1$, $n = 2^r - 1$, and $m = 2^{r-1}$. Then $c_\wedge(E_m^n) = n - 1$.*

[4] Note that, in the notation A_m^q, the parameter n is implicit.

[5] H_r^n is defined in section 5.

The majority function, a special case of the threshold function, is of particular importance in applications of this theory (e.g. electronic voting protocols). The first two results below give bounds for the majority function and the third general result on threshold functions is obtained using similar techniques.

Theorem 9. *Let* $n = 2^r$ *and* $m = 2^{r-1} + 1$. *Then* $c_\wedge(T_m^n) = n - 1$.

Theorem 10. $c_\wedge(T_m^{2m-1}) \leq 2^{\lceil \log_2 m \rceil} + m - \lceil \log_2 m \rceil - 2$ *for all* $m \geq 2$.

Theorem 11. $c_\wedge(T_m^n) \leq n - H^{\mathbb{N}}(n) + \lceil \log_2(n+1) \rceil - 1$ *for all* $m \geq 1$.

References

1. A. A. Aleksanyan. On realization of quadratic Boolean functions by systems of linear equations. *Cybernetics*, 25(1):9–17, 1989.
2. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pages 1–10, 1988.
3. J. Boyar, I. Damgård, and R. Peralta. Short non-interactive cryptographic proofs. *Journal of Cryptology*, 13:449–472, 2000.
4. J. Boyar, R. Peralta, and D. Pochuev. On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$. *Theoretical Computer Science*, 235:43–57, 2000.
5. P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1997.
6. R. Cramer, I. Damgård, and J. B. Nielsen. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–300. Springer-Verlag, 2001.
7. D. Chaum, C. Crépeau, and I. Damgård. Multi-party unconditionally secure protocols. In *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pages 11–19, 1988.
8. E. E. Kummer. Über die Ergänzungssätze zu den allgemeinen Reciprocitätsgesetzen. *J. Reine Angew. Math.*, 44:93–146, 1852.
9. J.B. Nielsen and M. Hirt. Upper bounds on the communication complexity of optimally resilient cryptographic multiparty computation. In *ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 79–99. Springer-Verlag, 2005.
10. M. V. Mihaľljuk. On the complexity of calculating the elementary symmetric functions over finite fields. *Sov. Math. Dokl.*, 20:170–174, 1979.
11. R. Mirwald and C. Schnorr. The multiplicative complexity of quadratic Boolean forms. *Theoretical Computer Science*, 102(2):307–328, 1992.
12. W. J. Paul. A $2.5n$ lower bound on the combinational complexity of boolean functions. In *Proceedings of the 7th ACM Symposium on the Theory of Computing*, pages 27–36, 1975.
13. R. Rueppel and J. Massey. The knapsack as a nonlinear function. In *Abstracts of papers, IEEE Int. Symp. on Information Theory*, page 46, 1985.
14. C. P. Schnorr. The multiplicative complexity of Boolean functions. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 6th International Conference*, volume 357 of *Lecture Notes in Computer Science*, pages 45–58, 1989.
15. L. Stockmeyer. On the combinational complexity of certain symmetric Boolean functions. *Mathematical Systems Theory*, 10:323–336, 1977.
16. B. L. van der Waerden. *Algebra*. Frederick Ungar Publishing.