Deterministic Restrictions in Circuit Complexity

Shiva Chaudhuri [*†] Jaikumar Radhakrishnan [‡]

Abstract

We study the complexity of computing Boolean functions using AND, OR and NOT gates. We show that a circuit of depth d with S gates can be made to output a constant by setting $O(S^{1-\epsilon(d)})$ (where $\epsilon(d) = 4^{-d}$) of its input values. This implies a superlinear size lower bound for a large class of functions. Using this, we obtain a function computable by a uniform family of constant depth polynomial size circuits that cannot be computed by constant depth circuits of linear size. We give circuit constructions that show that the bound $O(S^{1-\epsilon(d)})$ is near optimal.

We also study the complexity of computing threshold functions. The function T_r^n has the value 1 iff at least r of its inputs have the value 1. We show that a circuit computing T_r^n has at least $\Omega(r^2(\log n)/\log r)$ gates, for $r \leq n^{1/3}$, improving previous bounds. We also show a trade-off between the number of gates and the number of wires in a threshold circuit, namely, a circuit with G $(< n/2)$ gates and W wires computing T_r^n satisfies $W \geq \Omega(nr(\log n)/(\log(G/\log n)))$, showing that it is not possible to simultaneously optimize the number of gates and wires in a threshold circuit. Our bounds for threshold functions are based on a combinatorial lemma of independent interest.

1 Introduction

A fundamental goal of Boolean circuit complexity theory is to obtain bounds on the size of circuits computing various functions. A counting argument shows that most Boolean functions are hard to compute, specifically, they require circuits of exponential size [21, 18](see [5], pp. 763, Theorem 2.4). However, this yields no information about the complexity of explicit Boolean functions. Few techniques are known that yield lower bounds for explicit functions in NP, and this is an important problem in circuit complexity. One of the most succesful techniques for proving lower bounds on the size of circuits is the method of *random restrictions* [14, 23, 16]. The method consists of randomly setting the value of a number of the inputs to a circuit and showing that the resulting circuit has a simple structure. In this paper, we study a similar technique, *deterministic restrictions*, that yields lower bounds in some situations when random restrictions fail. Here also, the basic technique is to set input values and simplify the circuit, but the inputs to set are chosen deterministically, based on the structure of the circuit. As an application of the method, we obtain a separation between linear and superlinear size circuits of constant depth. We also study the computation of threshold functions and prove better size lower bounds for circuits computing small threshold functions and lower bounds on the number of wires in threshold circuits. We discuss the results in more detail.

Using the random restriction technique, it can be shown (see [5], pp. 772, Corollary 3.7) that a circuit with S gates of depth d can be made to output a constant by setting the values of $n - n/(O(\log S))^d$ inputs. Thus, any polynomial size circuit can be made to output a constant by setting $n - n/\text{polylog}(n)$ inputs. This yields a superpolynomial size lower bound for a constant depth circuit computing any function that cannot be made constant by setting $n - n/\text{polylog}(n)$ input bits. However it does not say anything about functions which *can* be made constant by setting this many bits. We prove the following *Restriction Lemma*: a circuit with S gates and depth d can be made to output a constant by setting $5S^{1-\epsilon(d)}$ input values, where $\epsilon(d) = 1/4^d$. Thus, for appropriate constants α, β, a circuit of constant depth with $O(n^{1+\alpha})$ gates can be made to output a constant by setting $O(n^{1-\beta})$ input values. This yields a superlinear lower bound for any constant depth circuit computing a function that cannot be made constant by setting $O(n^{1-\beta})$ inputs. In other words, this allows us to prove a superlinear lower bound for a much larger class of functions.

Håstad showed that PARITY cannot be computed by a circuit of polynomial size and depth d for any fixed d [16]. Since any Boolean function can be computed by an explicit depth two circuit of exponential size, the result can be interpreted as saying that exponential size circuits are strictly more powerful than polynomial, for constant depth. We would also like to see such a separation between polynomial sizes of diferent degrees, i.e. between n^k and n^{k+1}. Eric Allender (personal communication) observed that using a counting argument one can show that there exist functions computable by depth two circuits of size n^{k+1} than cannot be computed by circuits of size n^k (of any depth). However, no explicit function with this property is known. We consider this question for the case $k = 1$.

The class AC^0 is the set of functions computable by uniform polynomial size circuits of constant depth. The class LC^0 is the set of functions computable by uniform *linear* size circuits of constant depth [19, 20]. While LC^0 appears to be a severely restricted class, it turns out to contain surprisingly complex functions. An open question posed in [19, 20] is whether LC^0 is properly contained in AC^0.

We answer this question in the affirmative. A 1/4-*approx-*

*The work of Shiva Chaudhuri was partially supported by the EC ESPRIT Basic Research Action No. 7141 (ALCOM II).

†Max-Planck Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. E-mail: shiva@mpi-sb.mpg.de

‡Tata Institute of Fundamental Research, Bombay, India. E-mail: jaikumar@tcs.tifr.res.in

imate selector is any function whose value is 0 if the number of 1's in the input is at most $n/4$, 1 if the number of 1's is at least $3n/4$, and can be either 0 or 1 otherwise. Such a function provides a rough estimate of the number of 1's and is extremely useful in parallel computation [19, 15, 10]. The existence of polynomial size constant depth circuits that compute a 1/4-approximate selector function was shown in [1, 3]. The construction used probabilistic arguments and was nonuniform. Recently, Ajtai gave a uniform construction of an AC^0 circuit to compute such a function [2]. It is easy to see that a 1/4-approximate selector function cannot be made constant by setting $n/4$ or fewer input bits. The Restriction Lemma then implies a superlinear lower bound on the number of gates in any constant depth circuit that computes such a function. Thus, no approximate selector function is in LC^0. Thus, polynomial size uniform circuits are strictly more powerful than linear, for constant depth circuits.

We actually prove a size-depth tradeoff for a general class of functions. The *robustness* of a function is the maximum number of input bits that can be revealed to an adversary without revealing the value of the function. In [22, 9], robustness was called *everywhere-sensitivity.* For example, the robustness of PARITY is $n-1$, of MAJORITY $\lfloor n/2 \rfloor$, and of a 1/4-approximate selector, $n/4$. The Restriction Lemma implies that a circuit of depth d with S gates computing a function of robustness R, has $S = \Omega(R^{1+\epsilon(d)})$, where $\epsilon(d) = 1/4^d$. Thus, a depth d circuit for a 1/4-aproximate selector has size $\Omega(n^{1+\epsilon(d)})$. The lower bound is optimal upto the value of $\epsilon(d)$. We give uniform, constructions of circuits of depth $O(d)$, computing a 1/4-approximate selector, with $O(n^{1+\delta(d)})$ where $\delta(d) = 1/2^d$. Thus, a linear size 1/4-approximate selector circuit has depth $\Theta(\log\log n)$.

The threshold function T_r^n assumes the value 1 iff at least r of its input bits have value 1. It follows from the lower bounds in [16], that if $r = (\log n)^{\omega(1)}$, then T_r^n is not in AC^0. It was shown by Denenberg *et al.* that for $r = (\log n)^k$ for constant k, T_r^n is in AC^0 [11]. Subsequently, constructions using fewer gates were given and finally it was shown that for these values of r, T_r^n is in LC^0 [20, 17]. In [17], circuits of depth d for T_r^n are constructed using $2^{O(r^{1/d})} \log n$ gates. The lower bound in [16] implies that a depth d circuit for T_r^n requires $2^{\Omega(r^{1/d})}$ gates. For unbounded depth circuits, [17] showed that a circuit for T_r^n requires r gates. For small values of r, there is a large gap between the upper and lower bounds, in particular, the lower bounds do not even depend on n. We show that any circuit computing T_r^n, for $2 \leq r < n^{1/3}$, must have $\Omega(r^2(\log n)/\log r)$ gates. The lower bound holds for circuits without any restriction on depth, and for small values of r, considerably improves the previous bounds, both for bounded and unbounded depth circuits.

While the number of gates has been extensively studied as a resource in circuit complexity, much less attention has been given to wires [7]. There is often a trade-off between gates and wires. T_r^n can be computed by a circuit with $O(n)$ gates and wires. On the other hand, for small values of r, the optimal number of gates is much less. The constructions in [17] and [20] use fewer gates ($o(n)$) but more wires ($O(n(r \log n)^2)$ and $O(nr^4 \log n)$ wires respectively). For constant r, the results of Friedman imply T_r^n circuits with $O(\log n)$ gates and $O(n \log n)$ wires [13]. In particular, is it possible to simultaneously optimize the number of gates and the number of wires? We give a partial answer to this question, by showing that a circuit with G ($< n/2$) gates and W wires computing T_r^n, for $2 \leq r \leq n^{1/3}$ satisfies $W \geq \Omega(nr \ln n/\ln(G/\ln n))$ Thus, it is not possible to simultaneously achieve a sublinear number of gates and a linear number of wires, when r is not constant. In particular, this gives tight bounds on complexity of constant threshold functions.

Our lower bounds for threshold functions are based on a combinatorial lemma of independent interest. A family of sets is k-cover-free if no set is contained in the union of k others. Suppose we are given a k-cover-free system of n subsets of a set X. What is the minimum possible cardinality of X? Erdős, Frankl and Füredi [12] showed that $|X| \geq \Omega(k \log n)$. We improve this bound and show that $|X| \geq \Omega(k^2 \log n/\log k)$.

2 Definitions

The model of computation we consider is unbounded fan-in and fan-out circuits with AND, OR and NOT gates. A circuit is modelled by a DAG in which each vertex of indegree greater than 0 is labelled by one of AND, OR, NOT. We refer to the vertices as *gates* and the edges as *wires.* There are n vertices, $x_1, \ldots, x_n$, of indegree 0, called input gates, which correspond to the input variables, and one vertex of outdegree 0, called the output gate.

The *level* of a gate, g, is the length of the longest path from an input gate to g.

By *fixing* or *setting* an input variable, we mean assigning it a value in $\{0,1\}$. A *partial input* is an input in which some of the input variables are fixed. Given a partial input, by *admissible inputs*, we mean the set of inputs consistent with the partial input (i.e. those inputs which agree with the partial input on its fixed variables).

We say a gate or a wire in the circuit is *fixed* by a partial input if, over all admissible inputs, the gate output or the value carried by the wire has the same value. A gate or wire that is not fixed is called *free*, as is an input variable whose value is not set.

We define the notions of *indegree* and *outdegree* of gates in the context of a partial input. Given a partial input, the indegree of a gate g is 0 if it is fixed by th partial input; otherwise it is the number of free gates feeding into g. The outdegree of a gate g is 0 if it is fixed by the partial input; otherwise, it is the number of free gates that g feeds into.

Say an input variable *influences* a gate, g, if there is a path from the input gate to g passing through only free gates and wires. It is not hard to see that whenever all the inputs that influence a gate are fixed, the gate is also fixed. Define $\delta(g)$ to be the set of variables that influence gate g. (Note that influence and $\delta(g)$ are defined with respect to a partial input.)

We say a function has *robustness* r if it is not possible to fix less than or equal to r of its input bits and fix the value of the function. An equivalent definition considers the function, f, as giving a 2-colouring of the vertices of the boolean cube $\{0,1\}^n$, i.e. for $x \in \{0,1\}^n$, the colour of the vertex x is $f(x)$. If the largest monochromatic subcube has dimension d, then the robustness of the function is $n-d$.

3 Small size, small depth circuits

In this section, we obtain a bound on the number of inputs that need to be fixed in order to make a circuit output a constant value. In the course of our proof, we will fix input variables so that for the resulting partial input the indegree and outdegree of gates are small.

Definition 3.1 (Regular Circuits) *Let d and M be positive integers. Let $d_0 = 1$, $d_1 = d$ and $d_{i+1} = d_i^4$. A circuit $\mathcal{C}$ is (d, M)-regular if*

(a) the indegree of every gate at level i is at most d_i; and

(b) the outdegree of every gate g at level i is at most $M \cdot \delta(g)$.

If with respect to a partial input σ, the circuit $\mathcal{C}$ satisfies conditions (a) and (b) above, then we say that $\mathcal{C}|_\sigma$ is (d, M)-regular. ($\mathcal{C}|_\sigma$ is the circuit obtained by applying the partial input to $\mathcal{C}$.)

Suppose $\mathcal{C}$ is a circuit of depth k. Let the size of $\mathcal{C}$ be S (including the input gates). We wish to obtain a partial input σ so that $\mathcal{C}|_\sigma$ is (d, M)-regular. To do this, we set some input variables carefully so that gates with high indegree or outdegree are eliminated. The method for setting the input variables is described in the procedures below. Let $d \geq 2$, M and $d_0, d_1, \ldots$ be as in Definition 3.1.

FixIndegree(i) Repeat as long as there is a free gate g in level i with indegree more than d_i. Since g is free, there must be a free gate g' at some lower level that feeds into g. Fix g' so that g is fixed. That is if g is an OR then fix g' at 1 and if g is an AND then fix g' is at 0. *For this, we need to set at most $\delta(g')$ input variables.*

FixOutdegree(i) Repeat as long as there is a free gate g in level i with outdegree$(g) > M \cdot \delta(g)$. If at least half the gates that g feeds into are ORs then fix g at value 1 by suitably setting the $\delta(g)$ input variables that influence g; otherwise, fix g at value 0 by setting the $\delta(g)$ input variables suitably. *Note that the number of gates fixed in this step is at least $M/2$ times the number of input variables set.*

To obtain the required partial input, we start with the partial input with no input variables set, and refine it successively by applying the operations FixIndegree and FixOutdegree in the order FixOutdegree(0), FixIndegree(1), FixOutdegree(1), FixIndegree(2), ..., FixIndegree(k). Let the partial input obtained using this process be $\sigma(d, M)$.

Claim 3.1 *$\mathcal{C}|_{\sigma(d,M)}$ is (d, M)-regular.*

Proof. Immediate from the definitions of FixIndegree and FixOutdegree. Details omitted. ∎

Claim 3.2 *The number of inputs set during calls to* FixOutdegree *is at most $2S/M$.*

Proof. The number of gates fixed is at least $M/2$ times the number of input variables set. Since there are only S gates in all, we set at most $2S/M$ inputs. ∎

Claim 3.3 *The number of inputs set during calls to* FixIndegree *is at most $2SM/d$.*

Proof. FixIndegree(i) is invoked after FixIndegree(1), FixIndegree(2), ..., FixIndegree($i-1$). This ensures that the indegrees of all gates at levels j less than i are at most d_j. Thus $\delta(h) \leq d_1 d_2 \ldots d_{i-1}$, for gates at h at levels less than i. It follows that the number of input variables set while fixing any gate at level i is at most $d_1 d_2 \ldots d_{i-1}$.

Since, FixIndegree(i) is invoked after FixOutdegree(0), ..., FixOutdegree($i-1$), the outdegree of any gate in levels j less than i is at most $M \cdot (d_1 d_2 \ldots d_{i-1})$. Since there are at most S gates in these levels, the number of gates in level i with degree more than d_i is at most $SM(d_1 d_2 \ldots d_{i-1})/d_i$.

Thus, the number of variables set during FixIndegree(i) is at most

$$SM(d_1 d_2 \ldots d_{i-1})^2/d_i.$$

Summing over all levels i, we get that the total number of input variables set is at most

$$SM \sum_{i=1}^{k} \frac{(d_1 d_2 \ldots d_{i-1})^2}{d_i}.$$

Using the conditions $d_1 \geq 2$ and $d_{i+1} = d_i^4$, it can be verified that $(d_1 d_2 \ldots d_{i-1})^2/d_i \leq 1/(2^{i-1}d)$; thus the sum above is bounded by

$$\frac{SM}{d} \sum_{i=1}^{k} 2^{-i+1} \leq \frac{2SM}{d}.$$

∎

We can now prove the main result of this section.

Lemma 3.1 (Restriction Lemma) *Let $\mathcal{C}$ be a circuit of size S and depth k. Then there exists a partial input ρ which sets at most $5S^{1-\epsilon(k)}$ inputs, where $\epsilon(k) = 4^{-k}$, such that $\mathcal{C}|_\rho$ is a constant.*

Proof. Let $d = S^{2\epsilon(k)}$ and $M = S^{\epsilon(k)}$. By the Claim 3.1 above, $\mathcal{C}|_{\sigma(d,M)}$ is (d, M)-regular. Now, we may fix the output of $\mathcal{C}|_{\sigma(d,M)}$ by fixing at most $d_1 d_2 \ldots d_{k-1}$ additional inputs. It can be verified that

$$d_1 d_2 \ldots d_{k-1} \leq d_k = d^{4^{k-1}} \leq S^{1/2} \leq S^{1-\epsilon(k)}.$$

[if $k = 0$ then $S = 1$.] Hence, by setting at most (using Claims 3.2 and 3.3 to bound the number of variables set in $\sigma(d, M)$)

$$\frac{2S}{M} + \frac{2SM}{d} + S^{1-\epsilon(k)} = 5S^{1-\epsilon(k)}$$

input variables, we have obtained the partial input ρ that fixes $\mathcal{C}$ to a constant. ∎

Clearly, if a circuit of depth k with S gates computes a function of robustness R, then $5S^{1-\epsilon(k)} \leq R$, because otherwise, the function can be made constant by setting less than R inputs, a contradiction. This proves

Theorem 3.1 *If a circuit of depth k with S gates computes a function of robustness at least R, then*

$$S \geq \frac{1}{10} R^{1+\frac{1}{4^k}}.$$

The following is now easy.

Corollary 3.1 *$LC^0 \subset AC^0$.*

Proof. Consider a 1/4-approximate selector function. Suppose we could fix less than $n/4$ inputs and fix the value of the function to 0. By setting the remaining bits to 1, we have an input with more than $3n/4$ 1's but the value of the function is 0, a contradiction. A dual argument applies when the function is fixed to 1. Hence, the robustness of the function is at least $n/4$. By Theorem 3.1, any constant depth circuit computing a 1/4-approximate selector function has superlinear size. Thus all such functions do not belong to LC^0. Since, by [2], there is such a function in AC^0, the claim is proved. ∎

4 Upper bounds for approximate selectors

We describe a circuit that computes a 1/4-approximate selector function. Note that it is enough to construct an approximate addition circuit, i.e. one that takes n bits as input, and computes $\log n$ bits representing an integer, satisfying the following condition: If S is the integer represented by the output bits and E is the sum of the input bits, then $E - n/4 \leq S \leq E + n/4$. It is not hard to verify that a function which outputs 0 when $S < n/2$ and 1 when $S \geq n/2$ computes a 1/4-approximate selector. Thus we can simply output the high order bit of S.The rest of this section describes the construction of an approximate addition circuit.

Ajtai [2] describes a circuit of constant depth with a polynomial number of gates that behaves as follows: The inputs are $x_1, \ldots, x_n, r_1, \ldots, r_{\log n}$. The circuit outputs 0 if $\sum x_i < r(1 - 1/(\log n)^2)$ and 1 if $\sum x_i > r(1 + 1/(\log n)^2)$, where r is the number between 1 and n represented by the bits $r_1, \ldots, r_{\log n}$. Construct a circuit by placing n copies of the above circuit in parallel, each recieving the same inputs $x_1, \ldots, x_n$, but with the ith copy using the value $r = i$. Let y_i be the output of the ith circuit. Let $z_i = y_i \wedge \neg(y_{i-1} \vee \ldots \vee y_1)$. Then, z_i is 1 iff i is the smallest value of r for which the circuits computed 1. It follows that the z_i's are a unary representation of a number, s such that $\sum x_i(1 - 1/(\log n)^2) < s \leq \sum x_i(1 + 1/(\log n)^2)$. Convert this into the binary representation of the number. A polynomial size, constant depth circuit for unary to binary conversion is given in [8]. For numbers $a_1, \ldots, a_n$, define an ϵ-approximate sum to be any number, r, satisfying $(1 + \epsilon)\sum_{i=1}^{n} a_i \geq r \geq (1 - \epsilon)\sum_{i=1}^{n} a_i$. Then, the above discussion proves

Lemma 4.1 *We can construct circuits of polynomial size and constant depth with n inputs $x_1, \ldots, x_n$ and n outputs $y_1, \ldots, y_{\log n}$ such that*

$$\sum_{i=1}^{n} x_i(1 - 1/(\log n)^2) < s \leq \sum_{i=1}^{n} x_i(1 + 1/(\log n)^2),$$

where s is the integer between 1 and n represented by the bits $y_1, \ldots, y_{\log n}$.

We call the circuits constructed above *approximate bit addition* circuits.

In [8], it is shown how to construct a circuit that adds m q-bit numbers using a polynomial number of subcircuits that compute exact bit addition. This construction has the property that if each of the subcircuits used has polynomial size and constant depth, then so does the whole circuit. The construction has two stages. Let $A_1, \ldots, A_m$ be the numbers to be added. Let S denote the sum of these numbers. The first stage produces $\log m$ numbers $B_1, \ldots, B_{\log m}$, each of $q + \log m$ bits, such that $\sum_{i=1}^{\log m} B_i = S$. This stage uses exact bit addition subcircuits that add m bits and produced their $\log m$-bit sum. The second stage computes S from $B_1, \ldots, B_{\log m}$. This stage uses subcircuits that add at most $\log m$ bits and produce their sum. Since there exist circuits of size polynomial in m and constant depth that add $\log m$ bits, we can implement this stage exactly as in [8].

Our construction replaces each of the exact bit addition subcircuits in the first stage by an approximate bit addition circuit from Lemma 4.1. This will produce $\log m$ numbers $C_1, \ldots, C_{\log m}$ such that

$$S(1 + \frac{1}{(\log m)^2}) \geq \sum_{i=1}^{\log m} C_i \geq S(1 - \frac{1}{(\log m)^2}).$$

The second stage then computes the sum of the B_i's. We give the details of the first stage.

Let $a_{i,q}, \ldots, a_{i,1}$ be the binary representation of A_i. Let $S_j = \sum_{i=1}^{m} a_{i,j}$. Note that S_j can be represented in $\log m$ bits. Define

$$B_i = \sum_{j=0}^{\lfloor q/\log m \rfloor} S_{i+j\log m} 2^{i+j\log m}$$

for $1 \leq i \leq \log m$. Notice that computing B_i corresponds to "packing" every $(\log m)$th S_j (starting with the ith) into a single word, and can be done in constant depth. Clearly, $\sum_{i=1}^{\log m} B_i = \sum_{j=1}^{m} A_j$.

In our construction, we use the approximate bit addition circuits from Lemma 4.1 with inputs $a_{1,j}, \ldots, a_{m,j}$ to produce a $\log m$ bit number R_j, for each j, such that

$$S_j(1 + \frac{1}{(\log m)^2}) \geq R_j \geq S_j(1 + \frac{1}{(\log m)^2}).$$

We then compute

$$C_i = \sum_{j=0}^{\lfloor q/\log m \rfloor} R_{i+j\log m} 2^{i+j\log m}$$

by packing the appropriate values into single words. Because of the bounds on R_j in terms of S_j, it follows that $B_i(1 + 1/(\log m)^2) \geq C_i \geq B_i(1 - 1/(\log m)^2)$. Summing up these inequalities for each j yields

Lemma 4.2 *We can construct polynomial size (in m), constant depth circuits to compute the $1/(\log m)^2$-approximate sum of m numbers of q bits each, $q \leq m$.*

Let the circuits in the previous lemma have depth d and size m^{1+c} for some constant c. Write $f(m, i, \epsilon(m, i))$ for the minimum size circuit of depth id that computes the $\epsilon(m, i)$-approximate sum of m numbers. Given such circuits of depth id, we can construct circuits of depth $(i + 1)d$ as follows: Divide the input numbers into a groups, each of m/a numbers. Compute the approximate sums of each of the groups using circuits of depth id. Compute the approximate sum of the sums computed using the circuit of the previous lemma. Then, we have

$$f(m, i + 1, \epsilon(m, i + 1)) \leq af(m/a, i, \epsilon(m/a, i)) + a^{1+c}$$

By choosing a such that $a(m/a)^{1+c/i} = a^{1+c}$, we can show, by induction, that $f(m,i) \leq m^{1+c/i}$. Since the first phase computes $\epsilon(m/a, i)$-approximate sums and the second phase $1/(\log m)^2$-approximate sums, we have

$$1 + \epsilon(m, i+1) \leq [1 + \epsilon(m,i)][1 + \frac{1}{(\log m)^2}]$$

from which $\epsilon(m,i) \leq 3^i/(\log m)^2$ can be proved by induction. Thus, $f(m, c, 3^c/(\log m)^2) \leq m^2$. Note that the circuit has depth cd, a constant.

Write $g(m, i, \delta(m,i))$ for the smallest circuit of depth icd computing a $\delta(m,i)$-approximate sum. Then we have $g(m, 1, 3^c/(\log m)^2) \leq m^2$. Call this the *basic circuit.* We construct circuits of depth $(i+1)cd$, given circuits of depth icd as before, except that we now choose the size of each group to be $\sqrt{m}$ and use the basic circuit to compute the sums of the groups. The recurrence relation for the sizes is

$$g(m, i+1, \delta(m, i+1)) \leq \sqrt{m} g(\sqrt{m}, i, \delta(\sqrt{m}, i)) + (\sqrt{m})^2$$

It can be shown, by induction, that $g(m,i) \leq m^{1+1/2^{i-1}} + (i-1)m$. As before, it can be shown that

$$\delta(m,i) \leq 3^{i+c}/(\log m)^2,$$

which is at most $1/4$ whenever $i \leq \log\log m$ with m sufficiently large.

Given n bits and a depth k, we can compute, in k stages, the sums of disjoint groups of bits of size 2^k, by using pairwise addition. A circuit for pairwise addition with constant depth and a linear number of gates is given in [6]. Then we can use the circuit of depth kcd described above with $m = n/2^k$, to compute the approximate sum of the bits. The discussion above gives us the following

Theorem 4.1 *For any $k \leq \log\log n$, we can construct a $1/4$-approximate selector of depth $O(k)$ with $O(n^{1+1/2^{k-1}})$ gates. Thus selecting $k = \log\log n$ yields a circuit with $O(n)$ gates and $O(\log\log n)$ depth.*

5 k-cover-free systems

In this section, we prove bounds on the size of certain set systems. The bounds we prove will be used in proving lower bounds for threshold circuits.

A family of sets $\mathcal{F}$ is k-cover-free if $F_0 \not\subseteq F_1 \cup \ldots \cup F_k$ for all $\mathcal{F}_0, \ldots, \mathcal{F}_k \in \mathcal{F}$ ($F_i \neq F_0$ if $i \neq 0$). Let $f_k(m)$ denote the maximum cardinality of a k-cover-free family $\mathcal{F} \subseteq 2^{[m]}$. Erdős, Frankl and Füredi [12, Theorem 3.1] showed that

$$\left(1 + \frac{1}{4k^2}\right)^m < f_k(m) < \exp(\frac{(1+o(1))m}{k}). \tag{1}$$

In our application, we will be given a k-cover-free family $\mathcal{F}$ of cardinality n and will need a lower bound on $m = |\bigcup_{F\in\mathcal{F}} F|$. The second inequality in (1) gives $m = \Omega(k \ln n)$. We improve this bound and show that $m = \Omega(k^2 \ln n/\ln k)$. In other words, we strengthen the second inequality in (1) to $f_k(m) = \exp(O(m \ln k/k^2))$.

Assume that k is a large number and $n \geq k^3$. We say that $\mathcal{F}$ is an (n,k)-family if $|\mathcal{F}| \geq n$ and $\mathcal{F}$ is k-cover-free. Let $m(\mathcal{F}) = |\bigcup_{F\in\mathcal{F}} F|$.

Lemma 5.1 *Let $\mathcal{F}$ be a (n,r)-family such that $m(\mathcal{F}) \leq 2n/3$. Then there is a set in $\mathcal{F}$ of size at least*

$$\frac{r \ln n}{20 \ln(m(\mathcal{F})/\ln n)}.$$

Proof. Since $m(\mathcal{F}) \leq 2n/3$, at most $2n/3$ sets have a private element (an element not in any other set in the family). Then, every other sets in $\mathcal{F}$ has at least $r+1$ elements. Let $\mathcal{F}'$ be the family of these at least $n/3$ sets that do not have any private element (with respect ot $\mathcal{F}$). Then $\mathcal{F}'$ is a $(n/3, r)$-family, where each set has at least $r+1$ elements.

For each set $F \in \mathcal{F}'$ there is a subset S_F of size $\lceil |F|/r \rceil$ that is not included in any other set in the family $\mathcal{F}$ (for otherwise we could cover F by writing it as a union of r such sets). Then, the family of sets $\mathcal{S} = \{S_F : F \in \mathcal{F}'\}$ is an $(n/3, 1)$-family. Let t be the size of the largest set in $\mathcal{F}'$. Then the size of largest set in $\mathcal{S}$ is at most $\lceil t/r \rceil$. Since $t \geq r+1$, we have $\lceil t/r \rceil \leq 2t/r$. Since there are n sets in $\mathcal{F}$, we have $m(\mathcal{F}) \geq \log n$.

If $2t/r \geq m(\mathcal{F})/2$, then the lemma follows easily.

Otherwise, using the LYM inequality [4, page 11], we have

$$\left(\frac{em(\mathcal{F})}{2t/r}\right)^{2t/r} \geq \left(\frac{em(\mathcal{F})}{\lceil t/r \rceil}\right)^{\lceil t/r \rceil} \geq \binom{m(\mathcal{F})}{\lceil t/r \rceil} \geq n/3.$$

To justify the first inequality, note that $(a/x)^x$ is an increasing function of x, whenever $a \geq ex$.

To complete the proof, we need to show that

$$\left(\frac{em}{u}\right)^u \geq n/3 \Longrightarrow u \geq \frac{\ln n}{10 \ln(m/\ln n)}.$$

That is,

$$u \ln\left(\frac{em}{u}\right) \geq \ln(n/3) \Longrightarrow u \geq \frac{\ln n}{10 \ln(m/\ln n)}. \tag{2}$$

Suppose

$$u < \frac{\ln n}{10 \ln(m/\ln n)}.$$

Then the leftmost quantity in (2) is at most

$$\frac{\ln n}{10 \ln(m/\ln n)} \cdot \ln\left[\frac{10em \ln(m/\ln n)}{\ln n}\right].$$

That is,

$$\frac{\ln n}{10 \ln(m/\ln n)} \cdot \left[\ln(m/\ln n) + \ln 10e + \ln\ln(m/\ln n)\right].$$

We shall show that this quantity is less than $0.9 \ln n$ (we assume n is large). That would contradict the left hand side of (2). Note that $m \geq \log n \geq 1.442 \ln n$, because the universe must be big enough to accommodate n sets.

Case 1: Suppose $\ln(m/\ln n) \geq (\ln 10e)/7$. Then this quantity is at most

$$\frac{\ln n}{10}\left[1 + 7 + \frac{\ln\ln(m/\ln n)}{\ln(m/\ln n)}\right].$$

And the claim is correct since the last term inside the brackets is less than 1.

Case 2: Suppose $\ln(m/\ln n) < (\ln 10e)/7 < 3.303$. Thus,

$$\ln\ln(m/\ln n) < \ln((\ln 10e)/7) < -0.75.$$

Then the quantity is at most

$$\frac{\ln n}{10}\left[1 + \frac{3.303 - 0.75}{\ln(m/\ln n)}\right].$$

Since $m \geq 1.442 \ln n$, and $(3.303 - 0.75)/\ln 1.442 < 7$, this quantity is less than $0.8 \ln n$.

∎

Lemma 5.2 *Let $r > 1$ and $\mathcal{F}$ be a (n, r)-family such that $m(\mathcal{F}) \leq n/2$. Then the sum of the sizes of the sets in $\mathcal{F}$ is at least*

$$\frac{nr \ln n}{50 \ln m(\mathcal{F})/\ln n}.$$

Proof. Consider the $n/4$ largest sets in $\mathcal{F}$. It follows from Lemma 5.1 that they each have size at least

$$\frac{r \ln n'}{10 \ln(m(\mathcal{F})/\ln n')},$$

where $n' = 3n/4$. The lemma follows from this. ∎

Theorem 5.1 *If $\mathcal{F}$ is an (n, k)-family then $m(\mathcal{F}) \geq \dfrac{k^2 \ln n}{100 \ln k}$.*

Proof. We construct a sequence of family of sets $\mathcal{F}_1 = \mathcal{F}, \ldots, \mathcal{F}_{k/2}$ and a sequence of sets $F_1, \ldots, F_{k/2}$ as follows. F_i will be the set in $\mathcal{F}_i$ of the largest size and

$$\mathcal{F}_{i+1} = \{F - F_i : F \in \mathcal{F}_i \text{ and } F \neq F_i\}.$$

It can be verified that each $\mathcal{F}_i$ is an $(n - k/2, k/2)$-family. Hence, by Lemma 5.1 (we may assume that $m(\mathcal{F}_i) \leq k^2 \ln n$),

$$|F_i| \geq \frac{(k/2)\ln(n - k/2)}{20 \ln k}.$$

Thus,

$$m(\mathcal{F}) \geq \bigcup_{i=1}^{k/2} |F_i| \geq \frac{k}{2} \cdot \frac{(k/2)\ln(n - k/2)}{20 \ln k}.$$

The theorem follows from this. ∎

Corollary 5.1 $f_k(m) = \exp(O(m \ln k/k^2))$.

6 Threshold Circuits

In this section we prove lower bounds on the number of gates in a circuit computing T_k^n, and show a tradeoff between the number of gates and the number of wires in such a circuit. For both the above bounds, we use the following property of threshold functions. The value of the function is critically dependent on every input variable, in the sense that, for each variable, there is an input in which the value of the function changes when the value of that input is changed. Therefore, in the circuit, one cannot block the effect of any input variable on the output by fixing a small number of input variables. This implies that the family of sets formed by the gates connected to each input is a k-cover-free system.

The following lemma is folklore, and easy to prove. The proof is omitted from this abstract.

Lemma 6.1 *Any circuit C, may be converted into a circuit C', in which the negations are connected to only input gates. The number of gates in C' is at most twice the number of gates in C.*

Theorem 6.1 *A circuit computing T_k^n, with $k \leq n^{1/3}$ has $\Omega(k^2(\ln n)/\ln k)$ gates.*

Proof. By Lemma 6.1, we may assume that all the negations in the circuit are connected to the input gates. Let g be the number of gates in the circuit. If any AND (respectively, OR) gate is connected to a non-negated (respectively, negated) input gate, then set this input to 0. This ensures that this gate outputs a constant value, and we may delete it from the circuit. Repeat this until no remaining AND gates is connected to a non-negated input and no remaining OR gate is connected to a non-negated input.

The number of inputs whose values have been set is at most g, since each time an input is set to a fixed value, a gate is deleted from the circuit. If $g \geq n/2$ then the theorem holds. Otherwise, set an additional $n/2 - g$ inputs to 0. The circuit now computes $T_k^{n/2}$ and satisfies the conditions above.

Let $x_1, \ldots, x_m$, $m = n/2$, be the inputs to the circuit and let S_i be the set of gates that input x_i is connected to, either directly, or through one negation. Suppose there is a set S_1 that is contained in the union of $S_2, \ldots, S_k$. Then set inputs $x_2, \ldots, x_k$ to 1. Now, all the gates in the union of $S_2, \ldots, S_k$ output a constant, because all the AND gates are connected to only negated inputs and all the OR gates to only non-negated inputs. Thus, the gates in S_1 all output a constant, therefore the output of the circuit is independent of the value of x_1. However, the circuit must compute T_1^{m-k+1}, and hence should depend on the values of all inputs except $x_2, \ldots, x_k$, which is a contradiction.

Hence, the collection of sets $S_1, \ldots, S_m$ satisfies the condition that no set is contained in the union of $k - 1$ other sets. Applying Theorem 5.1 now yields the desired result.

∎

Theorem 6.2 *If a circuit with W wires and G $(< n/2)$ gates computes T_k^n, then*

$$W \geq \frac{nk \ln n}{100 \ln(G/\ln n)}$$

Proof. As in the previous theorem, we may assume that there are $m = n/2$ input variables $x_1, \ldots, x_m$, such that x_i is connected to the set of gates S_i and that the sets S_i form a $k - 1$-cover-free system. Since the number of wires in the circuit is at least the sum of the cardinalities of the S_i's, applying Lemma 5.2 yields the claimed bounds. ∎

7 Remarks

We have shown that uniform constant depth circuits with n^2 gates are more powerful than constant depth circuits with $O(n)$ gates. It would be interesting to show that for each $k \geq 1$, uniform constant depth circuits with n^{k+1} gates are more powerful than those with $O(n^k)$ gates. However, in order to do this, it seems new techniques are necessary.

It was conjectured that polylog threshold functions are candidate functions to separate AC^0 and LC^0, but this was shown to be false [20]. WLC^0 is the class of functions that have circuits of constant depth with a linear number of wires. Addition of two n bit numbers is known to be in LC^0 but not in WLC^0. There is no single output function known that separates LC^0 and WLC^0. We conjecture that polylog threshold functions are not in WLC^0.

The bounds of [16] yield a lower bound of $2^{k^{1/d}}$ for T_k^n and we prove a lower bound of $(k^2/\log k)\log n$. It would be nice to combine the two bounds and prove a lower bound of $2^{k^{1/d}}\log n$.

Acknowledgment

We are grateful to Mike Paterson and Uri Zwick for helpful discussions and for pointing us to the work in [2]. We thank Eric Allender for describing to us the counting argument that shows that there are functions computable by depth two circuits of size n^{k+1} but not by circuits of size n^k.

References

[1] M. Ajtai. $\sum_1^1$- formulae on finite structures. *Ann. Pure Appl. Logic* 24 (1983), pp. 1-48.

[2] M. Ajtai. Approximate counting with uniform constant depth circuits. In J.-Y Cai, ed. *Advances in Computational Complexity Theory*, DIMACS Series in Disc. Math. and Theoret. Comp. Sci., American Math. Society, (1993) pp. 1-20.

[3] M. Ajtai and M. Ben-Or. A theorem on probabilistic constant depth computations. In *Proc. 16th STOC*, (1984), pp. 471-474.

[4] B. Bollobás. Combinatorics. Cambridge University Press, 1986.

[5] R.B. Boppana and M. Sipser. The complexity of finite functions. Handbook of Theoret. Comp. Sci., Vol A, Algorithms and Complexity, Elsevier Science Publishers, 1990.

[6] A. K. Chandra, S. Fortune and R. J. Lipton, "Unbounded Fan-in Circuits and Associative Functions", *Proc. of the 15th ACM STOC*, 1983.

[7] A. K. Chandra, S. Fortune and R. J. Lipton, "Lower bounds for Constant Depth Circuits for Prefix Problems", *Proc. of the 10th Intl. Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer-Verlag, 1983.

[8] A.K. Chandra, L. Stockmeyer and U. Vishkin. Constant Depth Reducibility. *SIAM J. Comput.* 13, 2, (1984), pp. 423-439.

[9] S. Chaudhuri, "Sensitive Functions and Approximate Problems", *Proc. of 34th IEEE FOCS*, (1993), pp. 186-193.

[10] S. Chaudhuri, T. Hagerup and R. Raman. Approximate and Exact Deterministic Parallel Selection. In *Proc. 18th Math. Fdtns. of Comp. Sci.*, (1993), LNCS 711, Springer-Verlag, pp. 352-361.

[11] L. Denenberg, Y. Gurevich and S. Shelah. Definability by constant depth polynomial size circuits. Information and Control, 70 (1986), pp. 216-240.

[12] P. Erdős, P. Frankl and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. Israel Journal of Mathematics, 51 (1985), pp. 79–89.

[13] J. Friedman. Constructing $O(n \log n)$ size monotone formulae for the kth elementary symmetric polynomial of n Boolean variables. In *Proc. 25th Symp. on Found. of Comp. Sci.*, (1984), pp. 506-515.

[14] M. Furst, J. Saxe and M. Sipser. Parity, circuits and the polynomial time hierarchy. Mathematical Systems Theory, 17, (1984), pp. 13-27.

[15] T. Goldberg and U. Zwick. Optimal Deterministic Approximate Parallel Prefix Sums and Their Applications. In *Proc. Israel Symp. on Theory and Computing Systems (ISTCS'95)*, (1995), pp. 220-228.

[16] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proc. of18th STOC*, (1986), pp. 6-20.

[17] J. Håstad, I. Wegener, N. Wurm and S-Z. Yi. Optimal Depth, VerySmall Size Circuits for Symmetric Functions in AC^0. Information and Computation 108, (1994) pp. 200-211.

[18] D.E. Muller. Complexity in electronic switching circuits. IRE Trans. Electronic Computers, 5, (1956),pp. 15-19.

[19] I. Newman, P. Ragde and A. Wigderson, "Perfect Hashing, Graph Entropy and Circuit Complexity", *Proc. of 5th Ann. Conf. on Structure in Complexity Theory*, 1990, 91–99.

[20] P. Ragde and A. Wigderson. Linear-size constant-depth polylog-threshold circuits. *Information Processing Letters*, 39 (1991), pp. 143-146.

[21] C.E. Shannon. The synthesis of two-terminal switching circuits. Bell Systems Tech. Journal, 28 (1949), pp. 59-98.

[22] U. Vishkin and A. Wigderson. Trade-offs between depth and width in parallel computation. *SIAM Journal on Computing.*, 14 (1985) pp. 303-314.

[23] A.C. Yao. Separating the polynomial -time hierarchy by oracles. *Proc. of 26th FOCS*, (1985), pp. 1-10.