Electronic Colloquium on Computational Complexity, Report No. 49 (2006)

The Complexity of Depth-3 Circuits Computing Symmetric Boolean Functions

Guy Wolfovitz
Dept. of Computer Science
Haifa, Israel

April 8, 2006

Abstract

We give tight lower bounds for the size of depth-3 circuits with limited bottom fanin computing symmetric Boolean functions. We show that any depth-3 circuit with bottom fanin k which computes the Boolean function $\text{Exact}^n_{n/(k+1)}$, has at least $(1 + 1/k)^{n+\mathcal{O}(\log n)}$ gates. We show that this lower bound is tight, by generalizing a known upper bound on the size of depth-3 circuits with bottom fanin 2, computing symmetric Boolean functions.

1 Introduction

One of the most challenging problems in theoretical computer science is to prove lower bounds on the resources needed in order to compute a Boolean function. An important measure for the complexity of a Boolean function is the circuit complexity of the function.

A Boolean function is a function $f : \{0,1\}^n \to \{0,1\}$. A circuit is a directed acyclic graph in which every node of in-degree 0 is labeled with a variable $(x_1, x_2, \ldots)$ or its negation or a constant; the nodes which are labeled with either a variable or its negation are called the *inputs* of the circuit. Every other node v of in-degree $k \geq 1$ in the circuit is labeled with some Boolean function on k variables; furthermore these nodes are called *gates*. A special node with out-degree 0 is called the *output* of the circuit. A circuit with n inputs computes a Boolean function $f : \{0,1\}^n \to \{0,1\}$ in a natural way. If $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function and Ω is a set of Boolean functions then we denote by $C_\Omega(f)$ the minimal integer m such that there exists a circuit C over n variables and m gates that computes f and such that each gate in C computes a function from Ω. Every circuit has an obvious *representation* by a directed acyclic graph, where edges are directed from the inputs towards the output of the circuit. The *depth* of a circuit is the length of the longest directed path in its graph representation.

Why study circuit complexity? First, there is the practical interest: circuits for computing various Boolean functions are used within the hardware of digital computers, so it is important to understand, for optimization reasons, how large a circuit does a Boolean function requires. Secondly, studying circuit complexity has important applications to the theory of algorithms; one such application is this: proving sufficiently high lower bounds for the circuit complexity of a Boolean function f will imply a lower bound on the running time of algorithms computing f. This relation between circuit complexity and the theory of algorithms comes from efficient simulations of Turing machines by Boolean circuits. Efficient simulations are due to Schnorr [14] and Pippenger and Fischer [13], which shows how to simulate a Turing machine with time resources $t(n)$ and

ISSN 1433-8092

space resources $s(n)$ by a circuit of size $\mathcal{O}(t(n)\log(s(n)))$ over the Boolean basis B_2, consisting of all Boolean functions over 2 variables.

Shannon [15] shows that for a random Boolean function f on n variables, $C_{B_2}(f) = \Omega(2^n/n)$ is valid with high probability. Lupanov [8] shows that this lower bound is best possible, i.e., for every Boolean function f on n variables, $C_{B_2}(f) \leq \mathcal{O}(2^n/n)$. In contrast, for explicit Boolean functions f (i.e., functions computable in nondeterministic polynomial time,) the best lower bounds on their circuit complexity over the basis B_2 are merely linear. The best circuit lower bound over the full binary basis B_2 is due to Blum [1], who gives a lower bound of $3n - o(n)$ for an explicit Boolean function. For the basis U_2, which is B_2 without the XOR and its negation, the best circuit lower bound for an explicit Boolean function is $5n - o(n)$, due to Iwama and Morizumi [5] and Lachish and Raz [7].

1.1 Depth-3 Circuits

In this paper we consider the well studied, restricted model of depth-3 circuits. A formal definition of depth-3 circuits will be given at Section 2. For now it will suffice to think about depth-3 circuits over the variable set X, as circuits which are the OR (resp. AND) of an unbounded number of Boolean formulas over the variable set X, each written in conjunctive normal form (resp. disjunctive normal form.) We shall write CNF (DNF) for conjunctive normal form (resp. disjunctive normal form.) We also consider depth-3 circuits with bottom fanin k, which are the OR (resp. AND) of an unbounded number of Boolean formulas, each being a k-CNF (resp. k-DNF.)

Why study depth-3 circuits? Leslie Valiant [16] have showed that sufficiently large lower bounds on the size of depth-3 circuits with limited bottom fanin would imply super linear lower bounds on the size of logarithmic depth circuits over the basis B_2; in more detail, if f is a Boolean function over n variables which for some $\epsilon > 0$, cannot be computed by depth-3 circuits with bottom fanin n^ϵ and $2^{\mathcal{O}(n/\log\log n)}$ gates, then f cannot be computed by a linear sized logarithmic depth circuit over the basis B_2. What Valiant has observed is that the graph representation of a linear sized Boolean circuit over B_2 has a relatively small set of edges E, whose removal separates the graph representation into components of small depth; thus, by trying all possible truth assignments to the edges in E, while keeping valid the constraints implied by the circuit, Valiant have constructed a depth-3 circuit with relatively small bottom fanin which computes the same Boolean function as the original circuit. Valiant's reduction gives a good motivation for a large line of work done on the complexity of depth-3 circuits [6, 2, 3, 12, 9, 10, 4, 11]. For an explicit Boolean function, the largest lower bound for the number of gates in the depth-3 circuit model is due to Paturi, Pudlák, Saks and Zane [11], who showed that sufficiently dense codes cannot be accepted by depth-3 circuits of size less than $2^{\pi\sqrt{n}/6} > 2^{1.281\sqrt{n}}$; this lower bound was the first to give a lower bound for depth-3 circuits of the kind $2^{c\sqrt{n}}$, with $c > 1$. The technique in [11] also gives the best known known lower bounds for depth-3 circuits with bottom fanin k.

For symmetric Boolean functions, which are functions that essentially depend on the hamming weight of their input, the best known lower bound for depth-3 circuits with bottom fanin k was given by Paturi, Pudlák and Zane [9], who showed that $2^{n/k+o(n)}$ gates are necessary and sufficient in this model, in order to compute the parity function. Paturi, Saks and Zane [12] showed nonuniform constructions of depth-3 circuits with bottom fanin 2 for symmetric Boolean function. In Section 2 we extend the construction from [12] and show that any symmetric Boolean function can be computed by a depth-3 circuit with bottom fanin k and $(1 + 1/k)^{n+\mathcal{O}(\log n)}$ gates. We also show that this upper bound is best possible: we show that for any k there exists an explicit Boolean function which cannot be computed by depth-3 circuits with bottom fanin k and less than

$(1+1/k)^{n+\mathcal{O}(\log n)} \gg 2^{n/k+o(n)}$ gates. As a byproduct of the generalization of the construction of depth-3 circuits for symmetric Boolean functions we also answer an open question of Håstad, Jukna and Pudlák [3], regarding the complexity of depth-3 circuits that compute the Majority function: we improve the upper bound of Klawe *et. al.* [6] for the size of depth-3 circuits for Majority from $2^{\mathcal{O}(\sqrt{n \log n})}$ to $2^{\mathcal{O}(\sqrt{n})}$. This, together with the lower bound for Majority from [3] implies that the complexity of depth-3 circuits computing the Majority function is $2^{\Theta(\sqrt{n})}$.

2 Tight Bounds for Σ_3^k Circuits Computing Symmetric Boolean Functions

Say that a circuit is leveled if its gates can be arranged such that the inputs are at level 0 and gates at level $l > 0$ depend only on gates (or inputs) from level $l-1$. Say that a circuit is an alternating circuit over the De Morgan basis if it is a leveled circuit, the input nodes are labeled with $x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}, 0, 1$, gates at the same level are either all AND gates or all OR gates and lastly, OR gates (AND gates) at level l are being followed by AND gates (OR gates) at level $l+1$, if such a level exists. The depth of an alternating circuit over the De Morgan basis is the number of levels consisting of AND or OR gates. Denote by Σ_d^k the class of all alternating circuits over the De Morgan basis such that the depth of the circuit is d, the top level is an OR gate, the fanin of the gates at the first level is k and the fanin of all other gates is unbounded. Define the size of a circuit to be the number of gates of the circuit.

A Boolean function f over the variables $x_1, \ldots, x_n$ is symmetric if for every permutation σ in the group S_n of permutations on n letters, the following holds:

$$f(x_1, \ldots, x_n) = f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}).$$

For an integer k, define $\mathrm{sym}(k,d,n)$ to be the minimal integer m such that if f is a symmetric Boolean function over n variables then there exists a Σ_d^k circuit computing f, having m gates. In order for $\mathrm{sym}(k,d,n)$ to be well defined we assume that $d \geq 3$. Paturi, Saks and Zane [12] have showed that $\mathrm{sym}(2,3,n) \leq 1.5^{n+\mathcal{O}(\log n)}$. In this section we prove that this upper bound is best possible, i.e., $\mathrm{sym}(2,3,n) = 1.5^{n+\mathcal{O}(\log n)}$. In fact, we give a more general result and obtain a tight bound for $\mathrm{sym}(k,3,n)$ for all k.

Theorem 1. *$sym(k,3,n) = (1+1/k)^{n+\mathcal{O}(\log n)}$.*

We establish Theorem 1 by generalizing the upper bound of Paturi, Saks and Zane [12] and by proving a lower bound on the size of any Σ_3^k circuit computing the Boolean function $\mathrm{Exact}_{n/c}^n$, for $c = k+1$, where $\mathrm{Exact}_{n/c}^n$ is the Boolean function which accepts an input iff its hamming weight is exactly $\lfloor n/c \rfloor$. For clarity of exposition we will drop the use of the floor function in the remaining of the paper. We let $\mathcal{N}$ denote the set of all non-negative integers. For the lower bound we would need the following definition.

Definition 1. *Let $k,d,n \in \mathcal{N}$ such that $d \leq n$. Say that a k-CNF over n variables is a (k,d,n)-CNF if it accepts only assignments whose hamming weight is exactly d. Note that a (k,d,n)-CNF does not necessarily need to accept all, or any, assignments of hamming weight d.*

The lower bound part of the proof of Theorem 1 on Section 2.1 is based on a top-down approach which has been already used by others [2, 9, 11, 12] in order to give lower bounds for Σ_3^k circuits. This approach, in our context, can be sketched as follows. We first upper bound the possible number of satisfying assignments a (k,d,n)-CNF can have. We then show that if a Σ_3^k circuit computes Exact_d^n and has too small a size then one of its sub-circuits must accept a large set of assignments, which will lead to a contradiction given the upper bound on the number of assignments accepted by a (k,d,n)-CNF. For the upper bound on $\text{sym}(k,3,n)$ a straightforward application of the probabilistic method is used in Section 2.2. We stress the fact that the upper bound we give on $\text{sym}(k,3,n)$ is merely a generalization of the upper bound on $\text{sym}(2,3,n)$ from [12].

Lastly, we'd like to note the following. Let Π_d^k denote the class of all alternating circuits over the De Morgan basis such that the depth of the circuit is d, the top level is an AND gate, the fanin of the gates at the first level is k and the fanin of all other gates is unbounded. It is easy to see that if a symmetric Boolean function f cannot be computed by a Σ_d^k circuit with m gates then there is a symmetric Boolean function g which cannot be computed by a Π_d^k circuit with m gates; simply take $g = \neg f$. Hence, the lower bound on the size of Σ_3^k circuits computing symmetric Boolean functions in Section 2.1 applies also to Π_3^k circuits. Also, it would not be hard to see that the upper bound in Section 2.2 also applies to Π_3^k circuits.

2.1 The Lower Bound

We now give a lower bound on $\text{sym}(k,3,n)$. Let us ask the following question. Suppose that ϕ is a (k,d,n)-CNF. How large is the set of satisfying assignments of ϕ, $|\phi^{-1}(1)|$?

Lemma 1. *Let ϕ be a (k,d,n)-CNF. Then*

$$|\phi^{-1}(1)| \leq k^d.$$

Proof. For $t \in \mathcal{N}$, denote by $[t]$ the set $\{1,2,\ldots,t\}$. Suppose for now that given a (k,d',n')-CNF, with $d' > 0$, we can construct at most k formulas, $\phi_1,\ldots,\phi_t$ which satisfy the following two properties:

1. $\bigcup_{i=1}^t \phi_i^{-1}(1) = \phi^{-1}(1)$, and
2. For all $i \in [t]$, ϕ_i is a $(k,d'-1,n'-1)$-CNF.

Clearly, such a construction would imply by induction that the number of satisfying assignments of a (k,d,n)-CNF is at most k^d, as a $(k,0,n-d)$-CNF has at most one satisfying assignments. It is thus left for us to show how to construct the at most k formulas satisfying the two conditions above.

Let ϕ be a (k,d',n')-CNF, with $d' > 0$. Consider the assignment $\alpha = 0^{n'}$ to the variables of ϕ and observe that there must be some clause C of length $t \leq k$ in ϕ, which is not satisfied by α. Write $C = (x_1 \vee \ldots \vee x_t)$. For $i \in [t]$, define $\phi_i = \phi_{|x_i}$, where $\phi_{|x}$ denotes the formula ϕ restricted to the case where the literal x assumes the truth value 1. We claim that this construction satisfies the two properties mentioned above. To see that the first property hold, first observe that if β is a satisfying assignment for ϕ then it must satisfy at least one literal in C and hence, it must also satisfy at least one of the formulas ϕ_i. Also, it is clear that any satisfying assignment for ϕ_i, for all i, is also a satisfying assignment for ϕ. We thus conclude that $\bigcup_{i=1}^t \phi_i^{-1}(1) = \phi^{-1}(1)$. To see that the second property hold, note that ϕ_i has $n'-1$ variables and that any satisfying assignment for ϕ_i must have hamming weight $d'-1$, for all $i \in [t]$. This concludes the proof. □

Having Lemma 1 at hand we are now ready to prove the lower bound on $\text{sym}(k,3,n)$. Fix k and let $c \leq n$, be an integer whose value will be determined soon. We consider the Boolean function $\text{Exact}_{n/c}^n$. Let C be a Σ_3^k circuit computing $\text{Exact}_{n/c}^n$. Let s be the number of gates in C. Since C is a circuit of depth-3 with the top level gate being an OR gate, we can view C as a disjunction of at most s k-CNFs $\phi_1, \phi_2, \ldots, \phi_s$. This implies, together with the fact that the function computed by C is $\text{Exact}^n_{n/c}$, that ϕ_i is a $(k, n/c, n)$-CNF for all $i \in [s]$. Since the number of satisfying assignments of $\text{Exact}^n_{n/c}$ is $\binom{n}{n/c}$ we have by the pigeon hole principle that there exists $i \in [s]$ for which ϕ_i has at least $\binom{n}{n/c}/s$ satisfying assignments. By Lemma 1, since ϕ_i is a $(k, n/c, n)$-CNF, we have that

$$\binom{n}{n/c}\Big/ s \leq k^{n/c}. \tag{1}$$

For constant c we can approximate $\binom{n}{n/c}$ by $2^{H(1/c)n}$ where $H(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy function. It then follows from (1) that

$$\begin{aligned} s &\geq \binom{n}{n/c} k^{-n/c} \\ &\approx 2^{H(1/c)n} k^{-n/c} \\ &= (c^{1/c}(c/(c-1))^{1-1/c})^n k^{-n/c} \\ &= \left(\left(\frac{c}{k}\right)^{1/c}\left(\frac{c}{c-1}\right)^{1-1/c}\right)^n. \end{aligned} \tag{2}$$

It is easily seen that (2) is maximized whenever $c = k+1$. We hence conclude the following.

Lemma 2. *$sym(k, 3, n) \geq (1 + 1/k)^n$.*

2.2 The Upper Bound

We now turn to give the upper bound on $\text{sym}(k, 3, n)$. Paturi, Saks and Zane [12] gave an upper bound on $\text{sym}(k, 3, n)$, for $k = 2$. We give this upper bound for completeness. More accurately, we generalize the upper bound from [12] to give an upper bound on $\text{sym}(k, 3, n)$, for all $k \geq 2$.

To give the desired upper bound on $\text{sym}(k, 3, n)$ we note the following property of symmetric Boolean functions: if f is a symmetric Boolean function then it is the disjunction of at most $n+1$ functions from the set $\{\text{Exact}^n_{n/c} : n/c \in \{0, 1, \ldots, n\}\}$. Hence, in order to upper bound $\text{sym}(k, 3, n)$ it is suffice to construct Σ^k_3 circuits for $\text{Exact}^n_{n/c}$, for all $n/c \in \{0, 1, \ldots, n\}$. The plan for constructing a Σ^k_3 circuit for $\text{Exact}^n_{n/c}$ is as follows. First, we consider a probabilistic construction of a Σ^k_2 circuit of size at most $2^{2k}n$, which will be constructed in a way so that it will be in fact a $(k, n/c, n)$-CNF. Secondly, we show that constructing a sufficient number s of such $(k, n/c, n)$-CNFs and taking their disjunction (which is a Σ^k_3 circuit) will result, with positive probability, in a circuit computing $\text{Exact}^n_{n/c}$. From this we will conclude the existence of a depth-3 circuit over at most $2^{2k}sn$ gates computing $\text{Exact}^n_{n/c}$.

Let $x_1, \ldots, x_n$ be the variables underlying the Σ^k_2 circuit we are about to construct probabilistically. In fact, we will construct k-CNFs instead of Σ^k_2 circuits, as clearly a k-CNF is a Σ^k_2 circuit. The formula we construct will be the conjunction of two formulas, ϕ_1 and ϕ_2. The formula ϕ_1 will consist of a conjunction of clauses of length 1, and ϕ_2 will consist of clauses of length k. The construction is as follows. First partition uniformly at random the set of n variables to two sets S_1 and S_2 of size $n - kn/c$ and kn/c, respectively. Define ϕ_1 to be the conjunction of the literals $\overline{x_i}$, for all $x_i \in S_1$. Next, to the definition of ϕ_2, which is a formula over the variables in S_2. Partition the variables in S_2 u.a.r. to n/c sets R_i of size k each. Define ϕ_2 to be the conjunction of the Boolean functions ψ_i, where ψ_i is the k-CNF over the variables of R_i, computing the function Exact^k_1. Lastly, let ϕ be the conjunction of ϕ_1 and ϕ_2. Observe that the formula ϕ is a $(k, n/c, n)$-CNF. Also note that each ψ_i has at most 2^{2k} clauses and so the size of ϕ is at most $2^{2k}n$.

Let α be an assignment to n variables having hamming weight exactly n/c. Denote by A_α the event that α is a satisfying assignment for ϕ, where ϕ is as defined in the previous paragraph. We claim that $\mathrm{Prob}(A_\alpha) = k^{n/c}/\binom{n}{n/c}$. Consider the probabilistic construction of ϕ from above. One way to think of the probabilistic construction of ϕ is as follows. We have an ordered list of $n - kn/c + n/c$ bins, the first $n - kn/c$ bins being of size 1 and the last n/c bins are of size k; we then think of the construction of ϕ as first choosing a random permutation of the n variables and entering each variable in order, to the first non-full bin. Then, we use the first $n - kn/c$ bins to construct ϕ_1 and the last bins to construct ϕ_2, as described above, treating the i-th bin of size k as the set R_i. Let A be the set of n/c variables which are assigned 1 under α. Then the probability that α satisfies the resulting formula ϕ is exactly the probability that the variables in A each ends up in one of the last n/c bins, those of size k and there does not exist a bin of size k which do not contain a variable from A. We note that each variable can "fall" in k different places in each bin of size k. Hence, the number of permutations of n variables which lead to a construction of a formula which is satisfied by α is $k^{n/c}(n - n/c)!(n/c)!$. It then holds that $\mathrm{Prob}(A_\alpha) = k^{n/c}/\binom{n}{n/c}$, as claimed.

Let B_α be the event that upon constructing s k-CNFs as described above, independently of each other, α does not satisfy any of the s formulas. Then

$$\mathrm{Prob}(B_\alpha) \leq \left(1 - k^{n/c}\Big/\binom{n}{n/c}\right)^s \leq e^{-sk^{n/c}/\binom{n}{n/c}}.$$

Hence, the expected number of assignments having hamming weight n/c and which are not satisfied by any of the s formulas is $\binom{n}{n/c}e^{-sk^{n/c}/\binom{n}{n/c}}$. Choosing $s = k^{-n/c}\binom{n}{n/c}\ln\binom{n}{n/c}$, the expected number of assignments not satisfied by any of the s formulas is less than 1. We note that s is maximized whenever $c = k+1$, as in the argument for the lower bound. Since every symmetric Boolean function is computed by a disjunction of Σ_3^k circuits computing functions from $\{\mathrm{Exact}^n_{n/c} : n/c \in \{0, 1, \ldots, n\}\}$, we have that

$$\begin{aligned} \mathrm{sym}(k,3,n) &\leq 2^{2k} n(n+1) k^{-n/(k+1)} \binom{n}{n/(k+1)} \ln \binom{n}{n/(k+1)} \qquad (3) \\ &\leq 2^{H(1/(k+1))n - \frac{\log_2 k}{(k+1)} n + \mathcal{O}(\log n)} \\ &= (1 + 1/k)^{n + \mathcal{O}(\log n)}. \end{aligned}$$

We hence have the following.

Lemma 3. *$sym(k,3,n) \leq (1+1/k)^{n+\mathcal{O}(\log n)}$.*

This concludes the proof of Theorem 1.

3 The Complexity of Σ_3 Circuits Computing Majority

We conclude this paper with the following observation. Håstad *et. al.* [3] have showed that the Majority Boolean function could not be computed by depth-3 circuits with less than $2^{0.849\sqrt{n}}$ gates and have asked to determine the asymptotics of the number of gates required for computing the Majority function by depth-3 circuits. Klawe *et. al.* [6] have showed that the Majority function can be realized by depth-3 circuits of size $2^{\mathcal{O}(\sqrt{n \log n})}$. We observe that Lemma 3 determine the asymptotic behavior of the size of depth-3 circuits computing the Majority function. In fact, we show that any symmetric Boolean function can be computed by a depth-3 circuit of size $2^{\mathcal{O}(\sqrt{n})}$.

Let us define the following. Let $\mathrm{sym}(d, n)$ be the minimal integer m such that if f is a symmetric Boolean function over n variables then there exists a Σ_d circuit computing f, having m gates, where Σ_d is defined exactly as Σ_d^k except for the fact that no restriction on the fanin of the first level is imposed. We observe here that the upper bound on $\mathrm{sym}(k, 3, n)$ given by Lemma 3 implies the following.

Theorem 2. *$sym(3, n) = 2^{\Theta(\sqrt{n})}$.*

Proof. Since $\mathrm{sym}(k, 3, n)$ provides an upper bound on $\mathrm{sym}(3, n)$, we have from (3),

$$\begin{aligned}\mathrm{sym}(3,n) &\leq \mathrm{sym}(\sqrt{0.5\cdot n\log_2 e}, 3, n)\\ &\leq 2^{2\sqrt{0.5\cdot n\log_2 e}}n(n+1)\Big(1+\frac{1}{\sqrt{0.5\cdot n\log_2 e}}\Big)^{n+\mathcal{O}(\log n)}\\ &= 2^{2\sqrt{2n\log_2 e}+\mathcal{O}(\log n)}.\end{aligned}$$

By known lower bounds for the size of depth-3 circuits computing the parity function [9], we have $\mathrm{sym}(3, n) \geq 2^{\sqrt{n}}$. This concludes the proof. □

Since Majority is a symmetric Boolean function we can use the upper bound from the proof of Theorem 2 in order to conclude the following, which answers the forth question of Håstad, Jukna and Pudlák, from [3].

Corollary 1. *The asymptotic complexity of depth-3 circuits for Majority is $2^{\Theta(\sqrt{n})}$. More precisely, letting M_n to be the least integer m such that there exists a depth-3 circuit of size m which computes the Majority function over n Boolean variables. Then*

$$2^{0.849\sqrt{n}} \leq M_n \leq 2^{3.399\sqrt{n}}.$$

References

[1] Norbert Blum. A boolean function requiring 3n network size. *Theor. Comput. Sci.*, 28:337–345, 1984.

[2] J. Håstad. Almost optimal lower bounds for small depth circuits. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20, New York, NY, USA, 1986. ACM Press.

[3] J. Håstad, S. Jukna, and P. Pudlák. Top-down lower bounds for depth-three circuits. *Comput. Complex.*, 5(2):99–112, 1995.

[4] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[5] Kazuo Iwama and Hiroki Morizumi. An explicit lower bound of $5n - o(n)$ for boolean circuits. In Krzysztof Diks and Wojciech Rytter, editors, *MFCS*, volume 2420 of *Lecture Notes in Computer Science*, pages 353–364. Springer, 2002.

[6] Maria Klawe, Wolfgang J. Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 480–487, New York, NY, USA, 1984. ACM Press.

[7] Oded Lachish and Ran Raz. Explicit lower bound of $4.5n - o(n)$ for boolean circuits. In *STOC*, pages 399–408, 2001.

[8] O.B. Lupanov. A method of circuit synthesis. *Izv. VUZ Radiofiz 1*, pages 120–140, 1958.

[9] Ramamohan Paturi, Pavel Pudlak, and Francis Zane and. Satisfiability coding lemma. *Chicago Journal of Theoretical Computer Science*, 1999.

[10] Ramamohan Paturi, Pavel Pudlak, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k -SAT. In *IEEE Symposium on Foundations of Computer Science*, pages 628–637, 1998.

[11] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-sat. *J. ACM*, 52(3):337–364, 2005.

[12] Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth three boolean circuits. *Computational Complexity*, 9(1):1–15, 2000.

[13] Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, 1979.

[14] Claus-Peter Schnorr. The network complexity and the turing machine complexity of finite functions. *Acta Inf.*, 7:95–107, 1976.

[15] C.E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.

[16] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In Jozef Gruska, editor, *MFCS*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/