A Non-Probabilistic Switching Lemma for the Sipser Function

Sorin Istrail* Dejan Zivkovic†

Department of Mathematics, Wesleyan University
Middletown, CT 06459, USA

Abstract. Valiant [12] showed that the clique function is structurally different than the majority function by establishing the following "switching lemma": Any function f whose set of prime implicants is a large enough subset of the set of cliques (and thus requiring big Σ_2-circuits), has a large set of prime clauses (i.e., big Π_2-circuits). As a corollary, an exponential lower bound was obtained for monotone $\Sigma\Pi\Sigma$-circuits computing the clique function. The proof technique is the only non-probabilistic super polynomial lower bound method from the literature. We prove, by a non-probabilistic argument as well, a similar switching lemma for the NC^1-complete Sipser function. Using this we then show that a monotone depth-3 (i.e., $\Sigma\Pi\Sigma$ or $\Pi\Sigma\Pi$) circuit computing the Sipser function must have super quasipolynomial size. Moreover, any depth-d quasipolynomial size non-monotone circuit computing the Sipser function has a depth-$(d-1)$ gate computing a function with exponentially many both prime implicants and (monotone) prime clauses. These results are obtained by a top-down analysis of the circuits.

1 Introduction

Proving lower bounds on the size or depth of Boolean circuits is a fundamental problem in complexity theory. It is quite remarkable that almost all methods used in deriving super polynomial lower bounds employ—in crucial parts of the argument—probabilistic reasoning. To appreciate the difficulty of obtaining non-probabilistic lower bound proofs, let us mention that there is only one such method in the literature due to Valiant [12]. It is used to show that any monotone $\Sigma\Pi\Sigma$-circuit computing the clique function requires exponential size.

Let us call *clique-like* a function whose prime implicants form a large enough subset of the set of cliques. Certainly such a function requires a big Σ_2-circuit.

*Supported in part by NSF grant CCR-8810074. The author's current address: Sandia National Laboratories, Dept. 1423, Algorithms and Discrete Math., Albuquerque, NM 87185-5800, USA. Email: scistra@cs.sandia.gov.

†The author's current address: Dept. of Math. and Comp. Science, Savannah State College, Savannah, GA 31404, USA. Email: dzivkov%uscn.bitnet@uga.cc.uga.edu.

The Valiant's result shows that any Π_2-circuit for the function is also big. This evidence was presented as a structural difference between the clique function and the majority function. Indeed, majority-like functions may have small Π_2-circuits.

In this paper we present results similar to those of Valiant's. They are stronger in the sense that they apply to a much easier function. Instead of the NP-complete clique function we show analogous results for the NC^1-complete Sipser function $S_{\log n}$, i.e., the alternating OR-AND complete binary tree of depth $\log n$. This means that the above type of structural difference, now between $S_{\log n}$ and majority, is present in NC^1 and therefore is not necessarily related to the fact that the clique function belongs to a higher complexity class.

Let us call *Sipser-like* a function whose set of prime implicants is a subset of at least $2^{\Theta(\sqrt{n})-\log^{O(1)} n}$ prime implicants of $S_{\log n}$. We establish

1. A structural characterization of the self-reducibility of $S_{\log n}$;
2. *Switching Lemma*: Any Sipser-like function has an exponential size set of prime clauses.
3. Any Sipser-like function requires super quasipolynomial size monotone depth-3 circuits;
4. Every non-monotone depth-d circuit computing a Sipser-like function has a depth-$(d-1)$ gate computing a function whose both the number of prime implicants and the number of (monotone) prime clauses is exponential.

Our results are obtained through an extensive combinatorial analysis of the self-reducibility of the Sipser function. This is of an independent interest and, together with the NC^1-completeness, may speak in favor of the use of $S_{\log n}$ as a target function in other lower bound proofs (e.g., attacking the separation of TC^0 and ACC from NC^1).

The paper is organized as follows. After recalling basic definitions and notation we study the Sipser function in more detail. Section 4 contains our switching lemma, and section 5 its application to lower bounds on the size of circuits.

2 Basic definitions and notation

This section contains definitions and notation adopted throughout the paper. We first recall some of the basic notions of the theory of Boolean functions (for more details see, for example, [13]). A *literal* is a variable or a negated variable. A conjunction of literals, p, is an *implicant* of a Boolean function f if $p \leq f$ pointwise. If in addition, no conjunction of any proper subset of the literals comprising p is an implicant, then p is a *prime implicant* of f. By the same token, a disjunction of literals, s, such that $f \leq s$ is a *clause* of f. It is a *prime clause* if, in addition, no disjunction of any proper subset of the literals comprising s is a clause. A (prime) clause or implicant is *monotone* if it has no

negated variables. In the case of monotone (prime) implicants and clauses, we will often regard them as sets of variables.

A Boolean formula $f(x_1, \ldots, x_n)$ determines the unique Boolean function $f : \{0,1\}^n \to \{0,1\}$ in a natural way. For a Boolean function f denote by $PI(f)$ and $PC(f)$, respectively, the set of prime implicants and the set of prime clauses. Likewise, the sets of all implicants and clauses of f are denoted by $I(f)$ and $C(f)$. By $\alpha(f)$ and $\beta(f)$ are meant the sizes of a smallest prime clause and prime implicant of f, i.e., $\alpha(f) = \min\{|s| \mid s \in PC(f)\}$ and $\beta(f) = \min\{|p| \mid p \in PI(f)\}$.

The set of variables that occur in a formula f is denoted by $V(f)$, and a mapping $\rho : V(f) \to \{0,1,*\}$ is referred to as a *restriction*. The function represented by the formula obtained by substituting $\rho(x)$ for each x in f for which $\rho(x) \neq *$ is denoted $f|\rho$. More generally, for a set F of formulas we write $F|\rho = \{f|\rho \mid f \in F\}$.

3 The Sipser function S_d

Definition 3.1 *The* Sipser function S_d *is defined for odd* $d = 1, 3, \ldots$ *such that if* $n = 2^d$ *then* $S_d : \{0,1\}^n \to \{0,1\}$ *as follows. Given the set* $\{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$ *of* n *(distinct) variables, we form the complete binary tree of depth* d *with the root labelled* $\vee$ *and each level thereafter labelled with alternating* $\wedge$ *and* $\vee$ *nodes. If the leaves of the tree are now labelled with the variables* $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$*, the tree represents the Boolean formula* $S_d(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$*, which in turn defines the Sipser function* S_d*.*

In this section we study the Sipser function in more detail. Clearly S_d is a monotone function, hence prime implicants and clauses are made out of positive variables only. Moreover, every prime implicant and clause intersect in exactly one variable. (This is true in general for a monotone function iff it has a formula with no repeated variable [8].)

The number of all prime implicants $|PI(S_d)|$ and clauses $|PC(S_d)|$ of S_d is abbreviated, respectively, to $\Delta(S_d)$ and $\Gamma(S_d)$, as well as Δ_d and Γ_d. Lengths of prime implicants (clauses) of S_d are identical, and the length is denoted, respectively, by $\pi(S_d)$ and $\sigma(S_d)$, or π_d and σ_d for short. It is easy to see that $\Delta(S_d) = 2^{2^{(d+1)/2}-1}$, $\Gamma(S_d) = 2^{2^{(2^{(d-1)/2}-1)}}$, $\pi(S_d) = 2^{(d-1)/2}$, and $\sigma(S_d) = 2^{(d+1)/2}$.

Lemma 3.2 *For* $x \in V(S_d)$ *denote by* $\Delta_d(x)$ *and* $\Gamma_d(x)$ *the number of prime implicants and prime clauses of* S_d *containing variable* x*. Then* $\Delta_d(x) = \Delta_d/\sigma_d$ *and* $\Gamma_d(x) = \Gamma_d/\pi_d$*.*

Proof: See [6]. ■

Lemma 3.3 (Window Lemma) *Let* S_d *be the Sipser function given by the formula* $S_d(x_{i_1}, \ldots, x_{i_n})$ *and* $X = \{x_{i_1}, \ldots, x_{i_n}\}$*. If* Y *is a non-empty subset of* X *and* $A_d(Y) = \{s \in PC(S_d) \mid Y \subseteq s\}$*, then* $|A_d(Y)| \leq \Gamma_d/2^{|Y|-2}$*.*

Proof: See [6]. ■

Let $\{L_r(S_d)\}_{r=1,2,...}$ denote a sequence of subfunctions of S_d, called *layers*, which are defined as follows. Each layer will have the form $L_r(S_d) = T_0^r \wedge T_1^r \wedge \cdots \wedge T_{\tau_r}^r$, where $T_0^r, T_1^r, \ldots, T_{\tau_r}^r$ are subfunctions of S_d called *triangles*, each triangle being itself a Sipser function. Informally, the sequence is constructed iteratively so that next layer is obtained in the following way. First, we fix one variable from every triangle of the current layer. Let us call the fixed variables *pivots*. Secondly, we build a suitable restriction dependent on the choice of pivot variables. Finally, the next layer is defined to be the previous layer under the restriction.

Formally, to start with we think of S_d as a triangle in itself. Now, fixing a pivot $x \in V(S_d)$, the layer $L_1(S_d)$ is formed as follows. Take any $s \in PC(S_d)$ with $s = x \vee t$ and define the restriction ρ_x on $V(S_d)$ by

$$\rho_x(y) = \begin{cases} 0\,, & \text{if } y \in t \\ *\,, & \text{if } y \notin t \end{cases}\,.$$

Then we define $L_1(S_d) = S_d|\rho_x$. Observe that we can write $L_1(S_d) = T_0^1 \wedge T_1^1 \wedge T_2^1 \wedge \cdots \wedge T_{\tau_1}^1$, where $T_0^1, T_1^1, T_2^1, \ldots, T_{\tau_1}^1$ are the triangles of depth $0, 1, 3, \ldots, d-2$ respectively.

The next layer $L_2(S_d)$ is dependent on the choice of pivot variables $x_i^1 \in V(T_i^1)$, $(i = 0, 1, \ldots, \tau_1)$. Given such a choice, we define $L_2(S_d) = L_1(T_0^1) \wedge L_1(T_1^1) \wedge L_1(T_2^1) \wedge \cdots \wedge L_1(T_{\tau_1}^1)$. Put another way, we have $L_2(S_d) = T_0^2 \wedge T_1^2 \wedge \cdots \wedge T_{\tau_2}^2$, where $T_0^2, T_1^2, \ldots, T_{\tau_2}^2$ are all the triangles from $L_1(T_0^1), \ldots, L_1(T_{\tau_1}^1)$.

In general, each time choosing one pivot from every triangle of the current layer we proceed in the same way to get the next layer. Namely, for $r = 1, 2, \ldots$, $L_r(S_d) = T_0^r \wedge T_1^r \wedge T_2^r \wedge \cdots \wedge T_{\tau_r}^r$, and the choice of pivot variables $x_i^r \in V(T_i^r)$, $(i = 0, 1, \ldots, \tau_r)$, we define $L_{r+1}(S_d) = L_1(T_0^r) \wedge L_1(T_1^r) \wedge L_1(T_2^r) \wedge \cdots \wedge L_1(T_{\tau_r}^r)$, or equivalently $L_{r+1}(S_d) = T_0^{r+1} \wedge T_1^{r+1} \wedge \cdots \wedge T_{\tau_{r+1}}^{r+1}$, where $T_0^{r+1}, T_1^{r+1}, \ldots, T_{\tau_{r+1}}^{r+1}$ are all the triangles from $L_1(T_0^r), \ldots, L_1(T_{\tau_r}^r)$.

There are three aspects of the layer $L_r(S_d)$ which are of special interest to us: the number of prime implicants, the number of depth-non-zero triangles, and the product of the lengths of prime clauses of all triangles of the layer. They are denoted, respectively, $\ell_r(d)$, $\tau_r(d)$, $\lambda_r(d)$. Since their exact values are rather cumbersome, we compute only tight bounds for them which suffice for our purposes.

Theorem 3.4 *For every $r = 1, 2, \ldots$, if $d = 1, 3, \ldots, 2r-1$ then $\ell_r(d) = 1$, $\tau_r(d) = 0$, $\lambda_r(d) = 1$; and if $d \geq 2r+1$ then*

$$\frac{2^{2^{\frac{d+1}{2}}-1}}{2^{\frac{d^r}{r!}}} \leq \ell_r(d) \leq \frac{2^{2^{\frac{d+1}{2}}-1}}{2^{(\frac{d}{8})^r \cdot \frac{1}{r!}}}\,,$$

$$\tau_r(d) \leq \left(\frac{d}{2}\right)^r \cdot \frac{1}{r!}\,,$$

$$2^{(\frac{d}{16})^{r+1}\cdot\frac{1}{(r+1)!}} \leq \lambda_r(d) \leq 2^{(\frac{d}{2})^{r+1}\cdot\frac{1}{(r+1)!}} .$$

Proof: See [6]. ■

4 The switching lemma for S_d

It is well known that even a monotone Boolean function can have exponentially many prime implicants while only polynomially many prime clauses (or vice versa). In this section we deal with a similar problem of switching between prime implicants and clauses for the Sipser function S_d. More precisely, let F be a nonempty subset of the set of prime implicants of S_d and for $n = 2^d$ let $f : \{0,1\}^n \to \{0,1\}$ be the function defined by $f(x_1, \ldots, x_n) = \bigvee_{p \in F} p$. The question is: what is a lower bound on the size of $PC(f)$? We show that, roughly, if $|F|$ is big enough, then $|PC(f)|$ is also big. (Our argument can be easily modified for the corresponding dual case.) The proof's outline in major steps is as follows.

- first important observation is the relationship between $|PC(f)|$ and $\alpha(f)$, namely $|PC(f)| \geq 2^{\alpha(f)-2}$;
- since $\alpha(f)$ can be small, we show that if $|F|$ is big enough, then there is a restriction ρ such that $S_d|\rho = S_{d'}, f|\rho = f'$ where $f' = \bigvee_{q \in F'} q$ for some $F' \subseteq PI(S_{d'})$, and $\alpha(f')$ is amplified;
- using these and showing $|PC(f)| \geq |PC(f')|$ we get $|PC(f)|$ is big if $|F|$ is.

Lemma 4.1 *Let S_d be the Sipser function and $F \subseteq PI(S_d)$ be nonempty. If $f = \bigvee_{p \in F} p$, then $|PC(f)| \geq 2^{\alpha(f)-2}$.*

Proof: For each $t \in PC(f)$ thought of as the set of variables, we let $A_d(t) = \{s \in PC(S_d) \mid t \subseteq s\}$. (Note that $PC(f) \neq \emptyset$ since f is a non-constant function, hence it makes sense to talk about $A_d(t)$.) We claim that

$$PC(S_d) = \bigcup_{t \in PC(f)} A_d(t) . \tag{1}$$

The inclusion $\supseteq$ is trivial, and to see the other part take $s \in PC(S_d)$. Then s is a clause for f and so there is a $\hat{t} \in PC(f)$ such that $\hat{t} \subseteq s$, i.e., $s \in A_d(\hat{t})$.

Now, by Lemma 3.3 and definition of $\alpha(f)$ we have $|A_d(t)| \leq \Gamma_d/2^{|t|-2} \leq \Gamma_d/2^{\alpha(f)-2}$ for every $t \in PC(f)$, i.e.,

$$\max\{|A_d(t)| \mid t \in PC(f)\} \leq \frac{\Gamma_d}{2^{\alpha(f)-2}} . \tag{2}$$

Combining (1) and (2), one obtains

$$\Gamma_d = |PC(S_d)| \leq \sum_{t \in PC(f)} |A_d(t)| \leq |PC(f)| \cdot \max\{|A_d(t)| \mid t \in PC(f)\}$$
$$\leq |PC(f)| \cdot \frac{\Gamma_d}{2^{\alpha(f)-2}} .$$

Therefore, $|PC(f)| \geq 2^{\alpha(f)-2}$ as desired. ■

Lemma 4.2 *Let S_d be the Sipser function and $F \subseteq PI(S_d)$ be nonempty. For each $x \in V(S_d)$ define the set $F(x) = \{p \in F \mid x \in p\}$. Then there is an $\hat{x} \in V(S_d)$ such that $|F(\hat{x})| \geq |F|/\sigma_d$.*

Proof: See [6]. ■

Lemma 4.3 (Amplification Lemma) *Let S_d be the Sipser function and $F \subseteq PI(S_d)$ be nonempty. For $n = 2^d$ define $f : \{0,1\}^n \to \{0,1\}$ by $f(x_1, \ldots, x_n) = \bigvee_{p \in F} p$. If $|F| \geq \ell_r(d)$ and $d \geq 2^{5(r+1)}$ for some $r \geq 1$, then there is a restriction ρ on $V(S_d)$ such that*

a) *$S_d|\rho = S_{d'}$ with $d' = \Omega(d)$,*

b) *$F|\rho = F'$ where $F' \subseteq PI(S_{d'})$, and*

c) *if $f' = f|\rho = \bigvee_{q \in F'} q$, then $\alpha(f') = 2^{\Omega(d)}$ and $\beta(f') = 2^{\Omega(d)}$.*

Remark: The constants hidden in the Ω-notation depend on r.

Proof: Fix d and suppose $F \subseteq PI(S_d)$ is nonempty. For the sake of simplicity, in what follows we will usually omit notation for the depth d as a variable in quantities that depend upon it. Also, we put $\lambda_0 = \lambda_0(d) = \sigma_d$.

Now take $r \geq 1$ for which $|F| \geq \ell_r$ and $d \geq 2^{5(r+1)}$. We first construct a layer L_r, restriction η, and set $F_r \subseteq F \cap PI(L_r)$ such that $F_r = F|\eta$ and

$$|F_r| \geq \frac{|F|}{\lambda_0 \cdot \lambda_1 \cdots \lambda_{r-1}} .$$

To this end, for each $x \in V(S_d)$ define the set $F(x) = \{p \in F \mid x \in p\}$ and by Lemma 4.2 choose the pivot x_0^1 for L_1 such that $|F(x_0^1)| \geq |F|/\sigma_d = |F|/\lambda_0$. Building the layer L_1 in this way (i.e., using restriction $\eta_1 = \rho_{x_0^1}$), if $F_1 = F(x_0^1)$ we get $F_1 \subseteq F \cap PI(L_1)$, $F_1 = F|\eta_1$, and $|F_1| \geq |F|/\lambda_0$.

In order to obtain the layer L_2 let us introduce the following notation. For every variable $y_i \in V(T_i^1)$, where T_i^1, $(i = 0, 1, \ldots, \tau_1)$, are the triangles of the layer L_1, let

$$F_1(y_0) = \{p \in F_1 \mid y_0 \in p\},$$
$$F_1(y_0, y_1) = \{p \in F_1(y_0) \mid y_1 \in p\},$$
$$\vdots$$
$$F_1(y_0, y_1, \ldots, y_{\tau_1}) = \{p \in F_1(y_0, y_1, \ldots, y_{\tau_1 - 1}) \mid y_{\tau_1} \in p\}.$$

The pivots x_i^2, $(i = 0, 1, \ldots, \tau_1)$, for L_2 are now chosen as $x_0^2 = x_0^1$ (i.e., $F_1(x_0^2) = F_1$), and the others by repeated application of Lemma 4.2 such that for every $i = 1, \ldots, \tau_1$ we have $|F_1(x_0^2, x_1^2, \ldots, x_i^2)| \geq |F_1(x_0^2, x_1^2, \ldots, x_{i-1}^2)|/\sigma_i^1$, where $\sigma_i^1 = \sigma(T_i^1)$. It follows that

$$|F_1(x_0^2, x_1^2, \ldots, x_{\tau_1}^2)| \geq \frac{|F_1|}{\sigma_1^1 \cdot \sigma_2^1 \cdots \sigma_{\tau_1}^1} = \frac{|F_1|}{\lambda_1} \geq \frac{|F|}{\lambda_0 \cdot \lambda_1} .$$

Thus, if η_2 is the restriction yielding the layer L_2 and $F_2 = F_1(x_0^2, x_1^2, \ldots, x_{\tau_1}^2)$, then $F_2 \subseteq F \cap PI(L_2)$, $F_2 = F_1|\eta_2$, and $|F_2| \geq |F|/(\lambda_0 \cdot \lambda_1)$.

The reader should have no difficulty to see that, proceeding in this way for subsequent layers, we will eventually get the layer L_r with corresponding restriction η_r and the set $F_r = F_{r-1}(x_0^r, x_1^r, \ldots, x_{\tau_{r-1}}^r)$ such that $F_r \subseteq F \cap PI(L_r)$, $F_r = F_{r-1}|\eta_r$, and $|F_r| \geq |F|/(\lambda_0 \cdot \lambda_1 \cdots \lambda_{r-1})$. Therefore, for $\eta = \eta_1 \circ \eta_2 \circ \cdots \circ \eta_r$ we have $F|\eta = F_r$, and so our first goal is achieved.

Next, define the restrictions ξ_i, $(i = 0, 1, \ldots, \tau_r)$, on $V(L_r)$ by

$$\xi_i(y) = \begin{cases} *, & \text{if } y \in V(T_i^r) \\ 1, & \text{if } y \notin V(T_i^r) \end{cases} ,$$

and let P_i, $(i = 0, 1, \ldots, \tau_r)$, be the sets given by $P_i = F_r|\xi_i$. If we define $\Delta_i^r = \Delta(T_i^r)$, $\mu_i = |P_i|/(\Delta_i^r/\sigma_i^r)$, and $\mu = \prod_{i=0}^{\tau_r} \mu_i$, then

$$|F_r| \leq |P_0| \cdot |P_1| \cdots |P_{\tau_r}| = \mu_0 \frac{\Delta_0^r}{\sigma_0^r} \cdot \mu_1 \frac{\Delta_1^r}{\sigma_1^r} \cdots \mu_{\tau_r} \frac{\Delta_{\tau_r}^r}{\sigma_{\tau_r}^r} = \mu \cdot \frac{\ell_r}{\lambda_r} .$$

On the other hand $|F_r| \geq |F|/(\lambda_0 \cdot \lambda_1 \cdots \lambda_{r-1}) \geq \ell_r/(\lambda_0 \cdot \lambda_1 \cdots \lambda_{r-1})$, and so $\mu \geq \lambda_r / \prod_{i=0}^{r-1} \lambda_i$. Since $\mu = \prod_{i=0}^{\tau_r} \mu_i$, it follows that there is a $k \in \{0, 1, \ldots, \tau_r\}$ such that

$$\mu_k \geq \left(\frac{\lambda_r}{\prod_{i=0}^{r-1} \lambda_i} \right)^{1/\tau_r} . \tag{3}$$

Using the assumption $d \geq 2^{5(r+1)}$ and Theorem 3.4, it requires straightforward algebra to check that the right-hand side of (3) is at least $2^{d/2^{5(r+1)}}$, which implies $\mu_k = 2^{\Omega(d)}$.

Clearly, if we take $F' = P_k$, $\rho = \eta \circ \xi_k$, and $S_{d'} = T_k^r$ where d' is the depth of T_k^r, then $F' = F|\rho$ and $F' \subseteq PI(S_{d'})$. Moreover, it is easy to see that $S_d|\rho = S_{d'}$ and $f|\rho = \bigvee_{q \in F'} q$. Thus, if $f' = f|\rho$ it remains to prove $\alpha(f')$ and d' are as claimed.

For that, for each $x \in V(S_{d'})$ define the set $F'(x) = \{q \in F' \mid x \in q\}$. Since one variable $x \in V(S_{d'})$ occurs by Lemma 3.2 in at most $\Delta_{d'}/\sigma_{d'}$ prime implicants of $S_{d'}$, we have $|F'(x)| \leq \Delta_{d'}/\sigma_{d'}$. Now, take any $s \in PC(f')$ and observe $F' = \bigcup_{x \in s} F'(x)$, hence $|F'| \leq \sum_{x \in s} |F'(x)| \leq |s| \cdot \Delta_{d'}/\sigma_{d'}$, i.e.,

$$|s| \geq \frac{|F'|}{\Delta_{d'}/\sigma_{d'}} = \frac{|P_k|}{\Delta_k^r/\sigma_k^r} = \mu_k .$$

Because $s \in PC(f')$ is arbitrary, this implies $\alpha(f') \geq \mu_k = 2^{\Omega(d)}$. Finally, $d' = \Omega(d)$ easily follows from $2^{\Omega(d)} = \alpha(f') \leq 2^{d'}$, and $\beta(f') = 2^{(d'-1)/2} = 2^{\Omega(d)}$. ■

Lemma 4.4 *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function represented by a formula $f(x_1, \ldots, x_n)$. If ρ is any restriction on $\{x_1, \ldots, x_n\}$, then $|PC(f)| \geq |PC(f|\rho)|$.*

Proof: Straightforward. ■

Lemma 4.5 (Switching Lemma) *Let S_d be the Sipser function, and let $F \subseteq PI(S_d)$ be nonempty. For $n = 2^d$ define $f : \{0,1\}^n \to \{0,1\}$ by $f = \bigvee_{p \in F} p$. If $|F| \geq \ell_r(d)$ and $d \geq 2^{5(r+1)}$ for some $r \geq 1$, then $|PC(f)| = 2^{2^{\Omega(d)}}$.*

Remark: The constant hidden in the Ω-notation depends on r.

Proof: Let ρ be the restriction from Amplification Lemma such that $f' = f|\rho = \bigvee_{q \in F'} q$ for $F' \subseteq PI(S_{d'})$ and $\alpha(f') = 2^{\Omega(d)}$. Then

$$\begin{aligned} |PC(f)| &\geq |PC(f')| \text{ , by Lemma 4.4} \\ &\geq 2^{\alpha(f')-2} \text{ , by Lemma 4.1} \\ &= 2^{2^{\Omega(d)}} \text{ .} \blacksquare \end{aligned}$$

5 On depth-3 circuits computing S_d

In this section we use the previous results to easily obtain a lower bound on the size of depth-3 monotone circuits computing S_d. To this end we consider only $\Sigma\Pi\Sigma$-circuits since similar dual argument handles the case of $\Pi\Sigma\Pi$-circuits. Through the rest of the section we let $n = 2^d$.

Theorem 5.1 *Any monotone $\Sigma\Pi\Sigma$-circuit computing $S_{\log n}$ has super quasi-polynomial size.*

Proof: Let C be a monotone $\Sigma\Pi\Sigma$-circuit on n variables that computes $S_{\log n}$ and has size $2^{\log^k n}$, for some constant k. Then there is one AND gate on the middle level computing function, say, h with the following properties: $PI(h) = F \cup G$, where $F \subseteq PI(S_{\log n})$ and $|F| \geq \Delta_{\log n}/2^{\log^k n} = 2^{\Theta(\sqrt{n})-\log^k n}$; moreover, every $q \in G$ has the form $q = q'q''$ with $q' \in PI(S_{\log n}) \backslash F$ and q'' a non-empty product of variables. Thus, we can apply the Switching Lemma to $f = \bigvee_{p \in F} p$ and conclude that there exists $\epsilon > 0$ such that $PC(h) \geq 2^{n^\epsilon}$. Now, it is not hard to see that $|PC(h)| \geq |PC(f)|$ and so $PC(h) \geq 2^{n^\epsilon}$. But then the monotone subcircuit rooted at the AND gate which computes the function h must have exponential size, a contradiction. ■

At the end we note that we can prove a slightly stronger result. Namely, a non-monotone quasipolynomial size circuit computing the Sipser function still has a gate on the next-to-top level which computes a function with exponentially many monotone prime clauses. The details are omitted and can be found in [6].

Acknowledgments

We would like to thank Michael Sipser for insightful comments and discussions, and especially for pointing out an error in an earlier version of this paper. We also thank David Barrington for helpful suggestions about the results of our work.

References

[1] E. Allender, "A note on the power of threshold circuits", *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pp. 580–584, 1989.

[2] D. A. Barrington, "Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1", *Journal of Computer and System Sciences*, Vol. 38, pp. 150–164, 1989.

[3] R. Beigel and J. Tarui, "On ACC", *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pp. 783–792, 1991.

[4] R. B. Boppana and M. Sipser, "The Complexity of Finite Functions", *Handbook of Theoretical Computer Science, Vol. A* (J. van Leeuwen, ed., North-Holland, Amsterdam), pp. 757–804, 1990.

[5] J. Hastad, "Almost optimal lower bounds for small-depth circuits", *Proceedings of the 18th ACM Symposium on Theory of Computing*, pp. 6–20, 1986.

[6] S. Istrail and D. Zivkovic, "A non-probabilistic switching lemma for the Sipser function", *Wesleyan University, CS/TR-92-1*, 1992.

[7] M. Karchmer and A. Wigderson, "Monotone circuits for connectivity require super-logarithmic depth", *Proceedings of the 20th ACM Symposium on Theory of Computing*, pp. 539–550, 1988.

[8] D. Mundici, "Functions computed by monotone Boolean formulas with no repeated variables", *Theoretical Computer Science*, Vol. 66, pp. 113–114, 1989.

[9] A. A. Razborov, "Lower bounds on the monotone complexity of some Boolean functions", *Doklady Akademii Nauk SSSR*, Vol. 281(4), pp. 798–801, 1985 (in Russian). English translation in *Soviet Mathematics Doklady*, Vol. 31, pp. 354–357, 1985.

[10] A. A. Razborov, "Lower bounds on the size of bounded depth networks over a complete basis with logical addition", *Matematicheskie Zametki*, Vol. 41(4), pp. 598–607, 1987 (in Russian). English translation in *Mathematical Notes of the Academy of Sciences of the USSR*, Vol. 41(4), pp. 333–338, 1987.

[11] S. Skyum and L. G. Valiant, "A complexity theory based on Boolean algebra", *Journal of the ACM*, Vol. 22, pp. 484–504, 1985.

[12] L. G. Valiant, "Exponential lower bounds for restricted monotone circuits", *Proceedings of the 15th ACM Symposium on Theory of Computing*, pp. 110–117, 1983.

[13] I. Wegener, *The Complexity of Boolean Functions*, Wiley-Teubner, 1987.