Explicit Lower Bound Of *4.5n - o(n)* For Boolean Circuits

Oded Lachish*
Department of Computer Science
Weizmann Institute
Rehovot 76100, ISRAEL
odedl@wisdom.weizmann.ac.il

Ran Raz†
Department of Computer Science
Weizmann Institute
Rehovot 76100, ISRAEL
ranraz@wisdom.weizmann.ac.il

ABSTRACT

We prove a lower bound of $4.5n - o(n)$ for the circuit complexity of an explicit Boolean function (that is, a function constructible in deterministic polynomial time), over the basis U_2. That is, we obtain a lower bound of $4.5n - o(n)$ for the number of $\{and, or\}$ gates needed to compute a certain Boolean function, over the basis $\{and, or, not\}$ (where the *not* gates are not counted). Our proof is based on a new combinatorial property of Boolean functions, called *Strongly-Two-Dependence*, a notion that may be interesting in its own right. Our lower bound applies to any Strongly-Two-Dependent Boolean function.

1. INTRODUCTION

Shannon showed that the circuit complexity of almost all Boolean functions is exponential [1]. Lower bounds for explicit Boolean functions were proved for some restricted models of Boolean circuits (e.g., monotone circuits, constant depth circuits, etc'). For the general (non-restricted) model, however, no super-linear lower bound was obtained.

In this paper, we consider Boolean circuits over the basis U_2, which is one of the most common basis for Boolean circuits. The basis U_2 contains all the Boolean functions over two variables, except for the the *xor* function and its complement. That is, any gate over the basis U_2 can be replaced by an *and* gate (or, equivalently, an *or* gate), with the optional addition of *not* gates connected directly to the inputs to the gate and to the output of the gate. Hence, any Boolean circuit over U_2 can be converted into a Boolean circuit over the basis $\{and, or, not\}$, with the exact same number of gates (when the *not* gates are not counted). That is, the circuit complexity of a function over U_2 is equivalent to counting the number of $\{and, or\}$ gates needed to compute the function (when the *not* gates are ignored).

*Research supported by US-Israel BSF grant 98-00349.
†Research supported by US-Israel BSF grant 98-00349, and NSF grant CCR-9987077.

STOC'01, July 6-8, 2001, Hersonissos, Crete, Greece.

We prove a lower bound of $4.5n - o(n)$ for the circuit complexity of an explicit Boolean function with n input variables, over the basis U_2. The best previous lower bound was a bound of $4n - O(1)$, proved by Zwick [2].

For our proof, we define a new property of Boolean functions, called *Strongly-Two-Dependence*. A Boolean function is Two-Dependent if for any choice of two input variables, x_i, x_j, and any choice of two different assignments, σ', σ'', to x_i, x_j, the partial function obtained by fixing x_i, x_j to σ' is different than the partial function obtained by fixing x_i, x_j to σ''. A Boolean function is Strongly-Two-Dependent if each one of its partial functions, obtained by fixing less than $n - o(n)$ input variables to constants, is Two-Dependent.

Our lower bound is proved for any Strongly-Two--Dependent Boolean function. We do not give here an explicit construction of a Strongly-Two-Dependent Boolean function. Nevertheless, we do give a probabilistic construction for such a function F, using a small number of random bits (more specifically, we use $polylog(n)$ random bits). Our lower bound hence follows for the function F, as a function of both, the original input variables and the additional random bits. (Note that as a function of both, the original input variables and the additional random bits, the function F is deterministic and explicit). As in [2], our main method is the method of partial restrictions. That is, in each step we fix several input variables to constants, and obtain a smaller circuit. We use the Strongly-Two-Dependence property to obtain the better lower bound.

2. PRELIMINARIES

2.1 Boolean circuits over U_2

Define the basis U_2 to be the set of all Boolean functions $f : \{0,1\}^2 \to \{0,1\}$ of the sort

$$f(x, y) = ((x \oplus a) \wedge (x \oplus b)) \oplus c,$$

where $a, b, c \in \{0,1\}$. That is, U_2 is the set of Boolean functions (over two variables) that can be derived from the Boolean function $f(x,y) = x \wedge y$ by optionally applying some of the following: negate x, negate y, negate the output.

DEFINITION 2.1. *(Boolean circuit over U_2): A Boolean circuit over the basis U_2 is a directed acyclic graph with nodes of in-degree 0 or 2, such that:*

1. *Nodes of in-degree 0 are called input-nodes, and each one of them is labeled by a variable from $\{x_1, \ldots, x_n\}$ or a constant from $\{0,1\}$. Input-nodes labeled by a constant are called constant-nodes.*

2. *Nodes of in-degree 2 are called gate-nodes, and each one of them is labeled by a function from U_2.*

A specific subset of nodes are called output-nodes. In this paper, we only deal with Boolean functions $F : \{0,1\}^n \rightarrow \{0,1\}$ and hence we assume that our Boolean circuit has only one output-node.

We refer to a Boolean circuit over the basis U_2 as a Boolean circuit (unless stated otherwise). Let C be a Boolean circuit and let v be a node in C. We denote by $OUT_C(v)$ the set of gate-nodes, such that, v is connected directly to each one of them. We denote by $IN_C(v)$ the set of input-variables whose corresponding input-nodes are connected by a path to v. (If v is an input-node labeled by an input-variable then $IN_C(v) = \{v\}$). Given a gate-node v in C, we refer to the two nodes that are connected directly to v by $Right_C(v)$ and $Left_C(v)$.

Let C be a Boolean circuit and let $X = \{x_1, \ldots, x_n\}$ be the set of input-variables. Given an assignment $\sigma \in \{0,1\}^n$ to the variables in X, we denote by $C(\sigma)$ the value of the circuit's output on the assignment $x_i = \sigma_i$ (for every i). We compute $C(\sigma)$ as follows:

1. Label each input-variable x_i (i.e., input-node labeled by x_i) by the constant σ_i.
2. Find a gate-node v, such that, $Left_C(v)$ and $Right_C(v)$ are already labeled by constants $a_1, a_2 \in \{0,1\}$ respectively. Label v by $f(a_1, a_2) \in \{0,1\}$, where f is the Boolean function labeling the gate-node v.
3. Repeat step 2, until the output-node is labeled by a constant $a \in \{0,1\}$.

The value $C(\sigma)$ is the constant a. In the same way, for any node v in the circuit C, we denote by $C_v(\sigma)$ the value computed by v on the assignment $x_i = \sigma_i$.

Any Boolean circuit computes a Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$, where n is the number of input-variables. The other direction is also true: any Boolean function can be computed by a Boolean circuit. We say that two Boolean circuits C_1 and C_2 are equivalent ($C_1 \equiv C_2$) if they both compute the same function.

Note that given a Boolean circuit C, if we unite all input-nodes labeled by the same input-variable the circuit C will still compute the same Boolean function. Therefore, we can assume that for every input-variable x_i, there is only one input-node labeled by x_i. We will sometimes abuse notations and refer to that node by x_i. For example, the expression $x_i = Left_C(v)$ means: in the Boolean circuit C the input-node labeled by x_i is the left hand side input to v. In the same way, the expression $OUT_C(x_i)$ means the set of gate-nodes in C, such that, the input-node labeled by the input-variable x_i is directly connected to each one of them.

The size of a circuit C is the number of gate-nodes in it. We denote this number by $Size(C)$. The circuit complexity of a Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ is the minimal size of a Boolean circuit that computes F. We denote this number by $Size(F)$. Note that $Size(F)$ (i.e., the circuit complexity over U_2) counts the number of *and*,*or* gates needed to compute F over the base $\{and, or, not\}$ (i.e., we work over the standard base $\{and, or, not\}$ but the *not* gates are not counted).

The depth of a node v in a Boolean circuit C is the length of the longest path from v to the output-node. We denote this number by $Depth_C(v)$. The depth of a circuit C is the maximal depth of a node v in the circuit. We denote this number by $Depth(C)$.

The degree of a node v in a Boolean circuit C is the node's out degree. We denote this number by $Degree_C(v)$. We denote by $Degeneracy(C)$ the number of input-variables that have degree one in C. For our lower bound proof, we also need the following measure:

$$\mathsf{SD}(C) = Size(C) - 0.5 \cdot Degeneracy(C).$$

A similar definition was used in [2].

2.2 Blocking constants

The basis U_2 has some properties that are used in our lower bound proof. Recall that every Boolean function $f \in U_2$ can be represented as:

$$f(x,y) = ((x \oplus a) \wedge (x \oplus b)) \oplus c.$$

Define $Bl(f)$ to be the constant a in this expression. Define $Br(f)$ to be the constant b in this expression. Define $Dm(f)$ to be the constant c in this expression.

PROPOSITION 2.2. *Let $f(x,y)$ be a function in U_2. Then,*

$$f(Bl(f),0) = f(Bl(f),1) = f(0, Br(f)) =$$

$$= f(1, Br(f)) = Dm(f).$$

That is, fixing x to $Bl(f)$ or y to $Br(f)$ fixes $f(x,y)$ to one specific constant. We call this constant $Dm(f)$.

PROPOSITION 2.3. *Let $f(x,y)$ be a function in U_2. Then,*

$$f(\neg Bl(f), 0) \neq f(\neg Bl(f), 1)$$

and

$$f(0, \neg Br(f)) \neq f(1, \neg Br(f)).$$

That is, fixing x to $\neg Bl(f)$ fixes $f(x,y) = y$ or $f(x,y) = \neg y$. Fixing y to $\neg Br(f)$ fixes $f(x,y) = x$ or $f(x,y) = \neg x$.

Let C be a Boolean circuit and let v be a gate-node in C. Let $f \in U_2$ be the Boolean function labeling v in C. Define, $Bl_C(v) = Bl(f)$, $Br_C(v) = Br(f)$ and $Dm_C(v) = Dm(f)$.

2.3 Restrictions

A restriction θ is a mapping from a set of n variables to $\{0,1,\star\}$. That is, $\theta \in \{0,1,\star\}^n$. Intuitively, a restriction is a partial assignment to the set of input-variables. That is, some input-variables are assigned to a constant from $\{0,1\}$ and all other input-variables remain undetermined. Formally, we apply a restriction θ to a Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ in the following way: For any variable x_i that is mapped by θ to a constant $a_i \in \{0,1\}$, we assign a_i to x_i. We leave all the other variables untouched. We refer to the restricted Boolean function by $F \mid_\theta$. Note that $F \mid_\theta$ is a Boolean function of all the untouched variables. We apply a restriction θ to a Boolean circuit C in the following way: For any input-variable x_i that is mapped by θ to a constant $a_i \in \{0,1\}$, we relabel the input-variable x_i (i.e., the corresponding input-node) by a_i. We leave all the other nodes untouched. We refer to the restricted Boolean circuit by $C \mid_\theta$. In this paper, when we describe a restriction θ, we will only mention the input-variables that are mapped to constants in $\{0,1\}$. The input-variables that we do not mention are mapped to $\star$.

Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function over the set of variables $X = \{x_1, \ldots, x_n\}$. Let θ_1 and θ_2 be two restrictions, such that, θ_1 maps each one of the input-variables in the set X_1 to $\{0,1\}$ and θ_2 maps each one of the input-variables in the set X_2 to $\{0,1\}$. We say that the two restrictions θ_1, θ_2 are orthogonal if $X_1 \cap X_2 = \phi$. The composition of two orthogonal restrictions θ_1, θ_2 is well defined. We denote that composition by $\theta_1\theta_2$. The composition $\theta_1\theta_2$ maps each one of the input-nodes in X_1 according to θ_1 and maps each one of the input-nodes in X_2 according to θ_2.

PROPOSITION 2.4. *Let F be a Boolean function. Let C be a Boolean circuit. Let θ_1, θ_2 be two orthogonal restrictions. Then,*

$$(F \mid_{\theta_1}) \mid_{\theta_2} \equiv F \mid_{\theta_1\theta_2} \equiv F \mid_{\theta_2\theta_1} \equiv (F \mid_{\theta_2}) \mid_{\theta_1}$$

and

$$(C \mid_{\theta_1}) \mid_{\theta_2} \equiv C \mid_{\theta_1\theta_2} \equiv C \mid_{\theta_2\theta_1} \equiv (C \mid_{\theta_2}) \mid_{\theta_1} .$$

The last proposition can be easily generalized to the case of several orthogonal restrictions.

3. STRONGLY TWO DEPENDENCE

3.1 Definitions

DEFINITION 3.1. *(*Two-Dependent *Boolean function): Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function. We say that F is Two-Dependent if for any two different variables x_i,x_j and for any four constants $a, a', b, b' \in \{0,1\}$, such that, $(a,b) \neq (a', b')$, the following is satisfied: Let θ_1 be a restriction that maps x_i,x_j to a,b respectively. Let θ_2 be a restriction that maps x_i,x_j to a',b' respectively. Then,*

$$F \mid_{\theta_1} \neq F \mid_{\theta_2} .$$

DEFINITION 3.2. *(*(n, k)-Strongly-Two-Dependent *Boolean function): Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function. We say that F is (n,k)-Strongly-Two-Dependent if for any k different variables $\{x_{i_1}, \ldots, x_{i_k}\}$ and for any restriction θ that maps $\{x_{i_1}, \ldots, x_{i_k}\}$ to $\star$ and all other variables to constants from $\{0,1\}$, we have that $F \mid_\theta$ is Two-Dependent.*

PROPOSITION 3.3. *Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function. Then, for any n', such that, $n \geq n' > k$, for any set of n' different variables $X' = \{x_{i_1}, \ldots, x_{i_{n'}}\}$ and for any restriction θ that maps each one of the input-variables in X' to $\star$ and maps all other input-variables to $\{0,1\}$. $F \mid_\theta$ is (n', k)-Strongly-Two-Dependent.*

PROPOSITION 3.4. *Any (n,k)-Strongly-Two-Dependent Boolean function is also Two-Dependent.*

3.2 Construction

We present a construction for an explicit Boolean function $G : \{0,1\}^{\tilde{n}} \rightarrow \{0,1\}$, such that, for some restriction ϕ we have that $G \mid_\phi$ is (n,k)-Strongly-Two-Dependent, where $\tilde{n} = n + k^2$ and $k = t \log n$, and t is some big enough constant (say $t > 2^{20}$). We partition the $\tilde{n}$ input-variables of G into two sets: a set of n "regular" input-variables denoted by $(x_1, \ldots, x_n)$, and a set of k^2 "auxiliary" input-variables. In the analysis of the function, we think of the auxiliary input-variables as a string of random bits. We consider random restrictions ϕ that map the regular input-variables to $\star$ and the auxiliary input-variables to random constants. We show that with high probability (over the values of the auxiliary input-variables) $G \mid_\phi$ is (n,k)-Strongly-Two-Dependent. In all that comes bellow, the probability is taken over these random bits. For the sake of simplicity we also assume that $n > 2^8$. Thus, we also have that $k > 2^{23}$.

We define the function G as follows:

1. Use the auxiliary random string to choose n $k-wise$ independent vectors $\bar{c}_1, \ldots, \bar{c}_n$, such that, each $\bar{c}_i$ is a vector of k bits.

2. Define
$$\bar{c} = \bigoplus_{j=1}^{n} x_j \cdot \bar{c}_j.$$
That is, $\bar{c}$ is a vector of k bits, which is the bitwise *xor* of the n vectors $x_j \cdot \bar{c}_j$.

3. Define
$$G = Maj[\bar{c}].$$
That is, the value of G is one if at least half of the bits in $\bar{c}$ are one.

LEMMA 3.5. *There exists a restriction ϕ over the input-variables of G that maps the auxiliary variables to $\{0,1\}$, such that, $G \mid_\phi$ is (n,k)-*Strongly-Two-Dependent.

3.3 Proof of the Lemma

We will first prove that the vectors $\bar{c}_1, \ldots, \bar{c}_n$ satisfy two specific properties, with high probability over the possible assignments to the auxiliary input-variables. Denote by $W[\bar{v}]$ the Hamming weight of a vector $\bar{v}$ (i.e., $W[\bar{v}]$ is the number of ones in $\bar{v}$).

PROPOSITION 3.6. *With probability of at least $1 - \frac{1}{n}$ the following is satisfied: For every $1 \leq i < j \leq n$ and every four constants $a, a', b, b' \in \{0,1\}$, such that, $(a,b) \neq (a',b')$,*

$$W\left[(a \cdot \bar{c}_i) \oplus (b \cdot \bar{c}_j) \oplus (a' \cdot \bar{c}_i) \oplus (b' \cdot \bar{c}_j)\right] > \frac{1}{8}k.$$

PROOF. Observe that the value of $(a \cdot \bar{c}_i) \oplus (b \cdot \bar{c}_j) \oplus (a' \cdot \bar{c}_i) \oplus (b' \cdot \bar{c}_j)$ is equal to one of the following: $\bar{c}_i$, $\bar{c}_j$, $\bar{c}_i \bigoplus \bar{c}_j$.

For each i, the probability that $W[\bar{c}_i] \leq \frac{1}{8}k$ is at most n^{-10} (by the standard Chernoff bound). Since for $i \neq j$ the vectors $\bar{c}_i, \bar{c}_j$ are independent (as random variables), the probability that $W[\bar{c}_i \oplus \bar{c}_j] \leq \frac{1}{8}k$ is also at most n^{-10}.

There are n different possible values for i and $n(n-1)/2$ different possible values for i, j. Therefore, the probability that for every i we have $W[\bar{c}_i] > \frac{1}{8}k$ and for every $i \neq j$ we have $W[\bar{c}_i \oplus \bar{c}_j] > \frac{1}{8}k$ is larger than $1 - \frac{1}{n}$.

■

PROPOSITION 3.7. *With probability of at least $1 - \frac{1}{n}$ the following is satisfied: for every set of k different indices $i_1, \ldots, i_k$, the vectors $\bar{c}_{i_1}, \ldots, \bar{c}_{i_k}$ span a linear space of dimension at least $(1 - \frac{1}{256})k$.*

PROOF. Denote $k' = (1 - \frac{1}{256})k$. For the sake of simplicity we assume that k' is an integer.

The property is not satisfied only if there exists a set S of vectors $\{\bar{c}_{i_1}, \ldots, \bar{c}_{i_k}\}$ and a subset $S' \subset S$ of size k', such that, the vectors in S' span all the vectors in S. Recall that the vectors $\bar{c}_{i_1}, \ldots, \bar{c}_{i_k}$ are independent as random variables. Thus, the probability that all the $k - k'$ vectors in $S \setminus S'$ are spanned by S' is at most

$$\left(\frac{2^{k'}}{2^k}\right)^{k-k'} = 2^{-2^{-16}k^2},$$

(since the dimension of the linear space spanned by the vectors in S' is at most k', and the length of the vectors is k). The number of possibilities for choosing S,S' is

$$\binom{n}{k} \cdot \binom{k}{k'},$$

which is less than n^{2k}. Hence, by the union bound, the probability that there exist such S, S' is smaller than $\frac{1}{n}$ (since $k > 2^{20}$).
■

We are now ready to prove Lemma 3.5.

PROOF. By Proposition 3.6, Proposition 3.7, and the union bound, we have that with probability of at least $1 - \frac{2}{n}$:

1. For every $1 \leq i < j \leq n$ and every four constants $a, a', b, b' \in \{0,1\}$, such that, $(a,b) \neq (a',b')$,
$$W\left[(a \cdot \bar{c}_i) \oplus (b \cdot \bar{c}_j) \oplus (a' \cdot \bar{c}_i) \oplus (b' \cdot \bar{c}_j)\right] > \frac{1}{8}k.$$
2. For every set of k different indices $i_1, \ldots, i_k$, the vectors $\bar{c}_{i_1}, \ldots, \bar{c}_{i_k}$ span a linear space of dimension at least $(1 - \frac{1}{256})k$.

Hence, there exists a restriction ϕ, such that these two properties are satisfied. We will now show that $G|_\phi$ is (n,k)-*Strongly-Two-Dependent*.

Let θ be a restriction which is the composition of ϕ and a restriction that maps the k variables $\{x_1, \ldots, x_k\}$ to $\star$ and all other variables to the constants $h_{k+1}, \ldots, h_n \in \{0,1\}$. Let θ_1 be a restriction which is the composition of θ and a restriction that maps the variables $\{x_1, x_2\}$ to the constants $h_1^1, h_2^1 \in \{0,1\}$. Let θ_2 be a restriction which is the composition of θ and a restriction that maps the variables $\{x_1, x_2\}$ to the constants $h_1^2, h_2^2 \in \{0,1\}$, such that, $(h_1^1, h_2^1) \neq (h_1^2, h_2^2)$. We will prove that

$$G|_{\theta_1}(x_3, \ldots, x_k) \neq G|_{\theta_2}(x_3, \ldots, x_k)$$

(the proof for arbitrary k variables $x_{i_1}, \ldots, x_{i_k}$ is done in the same way).

Let $\bar{m}_1, \bar{m}_2$ be the two vectors defined as follows:

$$\bar{m}_1 = \left(\bigoplus_{r=k+1}^{n} h_r \cdot \bar{c}_r\right) \oplus (h_1^1 \cdot \bar{c}_1) \oplus (h_2^1 \cdot \bar{c}_2).$$

$$\bar{m}_2 = \left(\bigoplus_{r=k+1}^{n} h_r \cdot \bar{c}_r\right) \oplus (h_1^2 \cdot \bar{c}_1) \oplus (h_2^2 \cdot \bar{c}_2).$$

Let $\bar{D} : \{0,1\}^{k-2} \to \{0,1\}^k$ be the function defined over $\{x_3, \ldots, x_k\}$ as follows:

$$\bar{D}(x_3, \ldots, x_k) = \bigoplus_{r=3}^{k} x_r \cdot \bar{c}_r.$$

Thus, we have

$$G|_{\theta_1}(x_3, \ldots, x_k) = Maj\left[\bar{m}_1 \oplus \bar{D}(x_3, \ldots, x_k)\right]$$

$$G|_{\theta_2}(x_3, \ldots, x_k) = Maj\left[\bar{m}_2 \oplus \bar{D}(x_3, \ldots, x_k)\right]$$

By the choice of ϕ and since $(h_1^1, h_2^1) \neq (h_1^2, h_2^2)$, we have

$$W\left[\bar{m}_1 \oplus \bar{m}_2\right] > \frac{1}{8}k$$

and the vectors $\bar{c}_3, \ldots, \bar{c}_k$ span a linear space of dimension at least $(1 - \frac{1}{256})k - 2$. Thus, there is some specific set of $(1 - \frac{1}{256})k - 2$ coordinates of the vector $\bar{D}(x_3, \ldots, x_k)$, such that, the values of these $(1 - \frac{1}{256})k - 2$ bits can be determined to anything we want (by choosing appropriate values for $x_3, \ldots, x_k$).

We denote the set of indices of these bits by J. We choose an assignment of constants $h_3, \ldots, h_k \in \{0,1\}$ to $x_3, \ldots, x_k$, such that, the following two properties are satisfied:

1. For every index $j \in J$, such that, $\bar{m}_1^j \neq \bar{m}_2^j$ (where $\bar{m}_1^j$ is the j^{th} bit of $\bar{m}_1$ and $\bar{m}_2^j$ is the j^{th} bit of $\bar{m}_2$) the bit $\bar{D}^j(h_3, \ldots, h_k)$ satisfies $\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_1^j = 0$ and $\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_2^j = 1$ (where $\bar{D}^j(h_3, \ldots, h_k)$ is the j^{th} bit of $\bar{D}(h_3, \ldots, h_k)$).
2. For the indices $j \in J$, such that, $\bar{m}_1^j = \bar{m}_2^j$, we choose values, such that,
$$\sum_{j \in J}\left(\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_1^j\right) = \frac{7}{16}k.$$

Hence,

$$\frac{7}{16}k < \sum_{j=1}^{k}\left(\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_1^j\right) \leq \frac{7}{16}k + \frac{1}{256}k + 2 < \frac{1}{2}k.$$

Thus,

$$G|_{\theta_1}(h_3, \ldots, h_k) = 0.$$

On the other hand,

$$\sum_{j=1}^{k}\left(\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_2^j\right) - \sum_{j=1}^{k}\left(\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_1^j\right) \geq$$

$$\geq \frac{1}{8}k - 2\left(\frac{1}{256}k + 2\right).$$

Hence,

$$\sum_{j=1}^{k}\left(\bar{D}^j(h_3, \ldots, h_k) \oplus \bar{m}_2^j\right) \geq \frac{9}{16}k - 2\left(\frac{1}{256}k + 2\right) > \frac{1}{2}k.$$

Thus,

$$G|_{\theta_2}(h_3, \ldots, h_k) = 1.$$

■

3.4 Some easy propositions

The following lemma is the motivation behind the definitions of Two-Dependent and (n,k)-Strongly-Two-Dependent.

LEMMA 3.8. *Let $F:\{0,1\}^n \to \{0,1\}$ be a Two-Dependent Boolean function over the set of variables $X=\{x_1,\dots,x_n\}$. Let C be a Boolean circuit that computes F. Then, the following is never satisfied in C: There exist two input-variables x_i,x_j, such that, $OUT_C(x_i)=OUT_C(x_j)$ and $|OUT_C(x_i)|=|OUT_C(x_j)|=2$ (i.e., x_i,x_j are connected directly to the same two gate-nodes).*

PROOF. Without loss of generality, assume that $i=1$ and $j=2$. Let v_1,v_2 be the two different gate-nodes, such that, $OUT_C(x_1)=OUT_C(x_2)=\{v_1,v_2\}$. Without loss of generality, assume that $x_1=Left_C(v_1)$, $x_1=Left_C(v_2)$, $x_2=Right_C(v_1)$, $x_2=Right_C(v_2)$.

Let us partition the possibilities for values of $Bl_C(v_1)$, $Bl_C(v_2)$, $Br_C(v_1)$, $Br_C(v_2)$ into the following cases:

1. $Bl_C(v_1)=Bl_C(v_2)$ or $Br_C(v_1)=Br_C(v_2)$
2. $Bl_C(v_1)\neq Bl_C(v_2)$ and $Br_C(v_1)\neq Br_C(v_2)$

Let us analyze separately each one of these cases:

Assume that either $Bl_C(v_1)=Bl_C(v_2)$ or $Br_C(v_1)=Br_C(v_2)$. Without loss of generality, assume that $Bl_C(v_1)=Bl_C(v_2)$. Let θ_1,θ_2 be two restrictions, such that, θ_1 maps x_1 to $Bl_C(v_1)$ and x_2 to 0, and θ_2 maps x_1 to $Bl_C(v_1)$ and x_2 to 1. In $C\mid_{\theta_1}$ and in $C\mid_{\theta_2}$ the gate-nodes v_1,v_2 compute the same constant functions $Dm_C(v_1)$, $Dm_C(v_2)$, respectively. Hence, $C\mid_{\theta_1}\equiv C\mid_{\theta_2}$. Thus, $F\mid_{\theta_1}=F\mid_{\theta_2}$ in contradiction to the fact that F is Two-Dependent.

Assume that $Bl_C(v_1)\neq Bl_C(v_2)$ and $Br_C(v_1)\neq Br_C(v_2)$. Let θ_3,θ_4 be two restrictions, such that, θ_3 maps x_1 to $Bl_C(v_1)$ and x_2 to $Br_C(v_2)$, and θ_4 maps x_1 to $Bl_C(v_2)$ and x_2 to $Br_C(v_1)$. In $C\mid_{\theta_3}$ and in $C\mid_{\theta_4}$ the gate-nodes v_1,v_2 compute the same constant functions $Dm_C(v_1)$, $Dm_C(v_2)$, respectively. Hence, $C\mid_{\theta_3}\equiv C\mid_{\theta_4}$. Thus, $F\mid_{\theta_3}=F\mid_{\theta_4}$ in contradiction to the fact that F is Two-Dependent. ■

PROPOSITION 3.9. *Let $F:\{0,1\}^n\to\{0,1\}$ be a Two-Dependent Boolean function over the set of variables $X=\{x_1,\dots,x_n\}$. Let C be a Boolean circuit that computes F, and let v be the output-node of C. Then, $IN_C(v)$ contains all the input-variables in X.*

PROPOSITION 3.10. *Let $F:\{0,1\}^n\to\{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function over the set of variables $X=\{x_1,\dots,x_n\}$. Let C be a Boolean circuit that computes F. Then, the following is never satisfied in C: There exists an input-variable x_i, and a set of less than $n-k$ other input-variables X' and a restriction θ that maps each input-variable in X' to a constant in $\{0,1\}$, such that, in $C\mid_\theta$ every path that connects x_i to the output-node contains a gate-node that computes a constant function.*

PROPOSITION 3.11. *Let $F:\{0,1\}^n\to\{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function and let C be a Boolean circuit that computes F. Let v be a gate-node in C and let v' be the node, such that, $v'=Right_C(v)$. Assume that $Left_C(v)$ is an input-variable x_i, such that, $Degree_C(x_i)=1$. Then, one of the following is satisfied:*

1. *The node v' computes the constant function $\neg Br_C(v)$. (That is, $C_{v'}(\sigma)=\neg Br_C(v)$, for any assignment $\sigma\in\{0,1\}^n$).*
2. *The node v' computes a non constant function, and $|IN_C(v)|\geq n-k$.*

PROOF. In the first case all the paths between x_i and the output-node contain a gate-node that computes a constant Boolean function. That is a contradiction to Proposition 3.10. In the second case there exists a restriction θ that maps the input-variables in $|IN_C(v)|$ to constants, such that, v' computes the constant function $\neg Br_C(v)$, since the node v' computes a non constant function. By Proposition 3.3, $F\mid_\theta$ is $(n-|IN_C(v)|,k)$ Strongly-Two-Dependent, since $|IN_C(v)|\geq n-k$. This is a contradiction to the first case. ■

PROPOSITION 3.12. *Let $F:\{0,1\}^n\to\{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function and let C be a Boolean circuit that computes F. Denote by v the output-node of C and let $v'=Right_C(v)$. Assume that v' does not compute a constant function. Then, $IN_C(v')\geq n-k$.*

In the same way, assume that $Left_C(v)$ does not compute a constant function. Then, $IN_C(Left_C(v))\geq n-k$.

PROOF. Assume for the sake of contradiction that $IN_C(v')<n-k$. Let $n'=|IN_C(v')|$. Without loss of generality, assume that $IN_C(v')=\{x_1,\dots,x_{n'}\}$. Then, v' is a node that computes a non constant Boolean function $F':\{0,1\}^{n'}\to\{0,1\}$ over the the set $\{x_1,\dots,x_{n'}\}$. Therefore, there exists an assignment $\sigma\in\{0,1\}^{n'}$, such that, $F'(\sigma)=Br_C(v)$. Let θ be a restriction that maps each input-variable $x_j\in\{x_1,\dots,x_{n'}\}$ to σ_j. Then, in $C\mid_\theta$ the node v computes a constant function, since v' computes the constant function $Br_C(v)$. This is a contradiction to Definition 3.2, since by Proposition 3.3, $F\mid_\theta$ is $(n-n',k)$-Strongly-Two-Dependent (because $n'<n-k$). ■

4. THE LOWER BOUND

4.1 Circuit manipulation propositions

The following proposition supplies the method of removing degenerate gate-nodes (i.e., gate-nodes that do not contribute to the computation process of the Boolean circuit) from a Boolean circuit. It is widely used in the lower bound proof.

PROPOSITION 4.1. *Let C be a Boolean circuit. Assume that C contains one of the following degenerate cases:*

1. *A gate-node v, such that, for some constant $a\in\{0,1\}$ and any assignment σ, we have $C_v(\sigma)=a$ (that is, the function computed by the node v is a constant function).*
2. *A gate-node v, such that, a constant-node is connected directly to v and v computes a non constant Boolean function.*
3. *A non output gate-node v, such that, $Degree_C(v)=0$.*
4. *A gate-node v, such that, $Left_C(v)=Right_C(v)$.*

Then, there exists a Boolean circuit $C \equiv C'$, such that,

$$Size(C) \geq Size(C') + 1.$$

Proof. The proof is trivial. Nevertheless, we will present it here, since the exact argument will be important for the rest of the paper. Let C, v be as above. Let $f \in U_2$ be the Boolean function labeling v. Take C' to be identical to C and modify it as follows:

1. If for some constant $a \in \{0,1\}$ and any assignment σ, we have $C_v(\sigma) = a$ then relabel v by a and remove the two edges connected to v. That is, v is modified to be a constant-node.

2. Assume that a constant-node is connected directly to v and v computes a non constant Boolean function. Without loss of generality, assume that $Left_C(v)$ is a constant-node labeled by 1 and that v is labeled by a Boolean function $f \in U_2$, such that, either $f(1,y) = y$ or $f(1,y) = \neg y$. Then, remove v and the edges connected to it. Instead, connect the node $Right_{C'}(v)$ directly to each one of the nodes in $OUT_{C'}(v)$. If $f(1,y) = \neg y$ we also have to relabel each one of the gate-nodes $u \in OUT_{C'}(v)$ as follows: Let $g \in U_2$ be the Boolean function labeling u in C. If $v = Left_C(u)$ relabel u in C' by the Boolean function $g' \in U_2$, such that, $g'(x,y) = g(\neg x, y)$. If $v = Right_C(u)$ relabel u in C' by the Boolean function $g'' \in U_2$, such that, $g''(x,y) = g(x, \neg y)$.

3. If v is a non-output gate-node, such that, $Degree_C(v) = 0$. Then, remove v and the two edges connected to it.

4. If v is a gate-node labeled by a Boolean function $f \in U_2$, such that, $Left_C(v) = Right_C(v)$ and $f(x,x) = x$. Then, connect the node $Right_{C'}(v)$ directly to each one of the nodes in $OUT_{C'}(v)$. Remove v and the edges connected to it.

5. If v is a gate-node labeled by a Boolean function $f \in U_2$, such that, $Left_C(v) = Right_C(v)$ and $f(x,x) = \neg x$. Then, connect $Right_{C'}(v)$ directly to each one of the nodes in $OUT_{C'}(v)$. Relabel each one of the gate-nodes $u \in OUT_{C'}(v)$ as follows: Let $g \in U_2$ be the Boolean function labeling u in C. If $v = Left_C(u)$ relabel u in C' by the Boolean function $g' \in U_2$, such that, $g'(x,y) = g(\neg x, y)$. If $v = Right_C(u)$ relabel u in C' by the Boolean function $g'' \in U_2$, such that, $g''(x,y) = g(x, \neg y)$.

Note that a gate-node is removed if it was physically removed from the Boolean circuit or if it was modified into a constant-nodes. Therefore, in all the cases we have removed one gate-node. ∎

The following proposition is designed to remove as many degenerate cases as possible from a Boolean circuit that computes an (n,k)-Strongly-Two-Dependent Boolean function. We use it before we apply the random restriction technique in order to avoid dealing with the mentioned degenerate cases.

Proposition 4.2. *Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function and let C be a Boolean circuit that computes F. Then, there exists a Boolean circuit $C' \equiv C$, such that, $\mathsf{SD}(C) \geq \mathsf{SD}(C')$ and $Degeneracy(C') \leq k$ and C' does not contain any of the following degenerate cases:*

1. *Any of the degenerate cases of Proposition 4.1.*

2. *An input-variable x_i of degree greater than one, such that, $|OUT_{C'}(x_i)| \geq 2$ and one of the gate-nodes in $OUT_{C'}(x_i)$ is directly connected to a different gate-node in $OUT_{C'}(x_i)$.*

Proof. Let C,x_i be as in case 2. Without loss of generality, assume that $i = 1$. Let v_1,v_2 be two different gate-nodes , such that, $v_1, v_2 \in OUT_C(x_1)$ and v_1 is directly connected to v_2. Without loss of generality, assume that $v_1 = Left_C(v_2)$, $x_1 = Right_C(v_2)$ and $x_1 = Left_C(v_1)$. Denote by A the function computed by $Right_C(v_1)$. Note that the function computed by v_2 depends only on the values of x_1 and A. Denote by $\hat{f}(x_1, A)$ the function computed by v_2 (over x_1 and A). We can easily prove that $\hat{f} \in U_2$ by simply checking all the possible cases. For example, assume that both v_1 and v_2 are labeled by $\hat{f}(x,y) = x \wedge y$. Then, the function computed by v_2 over x_1,A is $\hat{f}(x_1, A) = (x_1 \wedge A) \wedge x_1 = x_1 \wedge A$.

We will take C' to be identical to C except for the following modifications: Remove the edge between v_1 and v_2. Instead, connect $Right_{C'}(v_1)$ directly to v_2 and relabel v_2 by $\hat{f}$. By Proposition 3.11, $Right_{C'}(v_1)$ cannot be an input-variable of degree one. Since no other input-variable of degree one was possibly effected, $\mathsf{SD}(C) \geq \mathsf{SD}(C')$.

We now apply the argument of Proposition 4.1. Observe that by Proposition 4.1, we removed one gate-node v from the circuit C'. This can only effect the degrees of $Left_{C'}(v)$ and $Right_{C'}(v)$, and may decrease the degeneracy of the circuit by at most two. Hence, the SD measure is not increased. We apply the described process iteratively until we exhaust all mentioned cases.

We will now show that $Degeneracy(C') \leq k$. Assume for the sake of contradiction that $Degeneracy(C') > k$. Let $k' = Degeneracy(C')$. Without loss of generality, let $X' = \{x_1, \ldots, x_{k'}\}$ be the set of input-variables of degree one in C'. Let $x_j \in X'$ be an input-variable, such that, for every $x_i \in X'$

$$Depth_{C'}(x_j) \geq Depth_{C'}(x_i).$$

Let v be the gate-node, such that, x_j is directly connected to v. Without loss of generality, assume that $x_j = Left_{C'}(v)$. Note that according to the definition of *Depth*, no other input-variable in X' is connected by an indirect path to v. By Proposition 3.11, no input-variable is directly connected to v. Hence, $Right_{C'}(v)$ is a gate-node that computes a Boolean function $F' : \{0,1\}^{n-k'} \rightarrow \{0,1\}$ over the set of input-variables $X \setminus X'$. Since we exhausted all the above mentioned cases, F' is not a constant function. This contradicts Proposition 3.11. ∎

4.2 The Lower Bound

Our lower bound proof is based on the gate elimination technique. For any Boolean circuit C that computes an (n,k)-Strongly-Two-Dependent Boolean function

$F : \{0,1\}^n \rightarrow \{0,1\}$ (for certain values of the parameters n, k), we use the properties of F to prove: There exists a specific restriction θ, such that, by using the argument of Proposition 4.1, we can remove specific gate-nodes from $C|_\theta$. We will actually work with the SD measure of the circuit (rather than the *Size* measure). We will show that the SD measure is decreased when we apply the restriction θ. The following Lemma captures this idea and is the major building block in our lower bound proof.

LEMMA 4.3. *Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function and assume that $n-k \geq 5$. Let C be a Boolean circuit that computes F. Then, there exists a set of one or two input-variables X' (i.e., $|X'| \leq 2$), and there exists a constant $c_i \in \{0,1\}$ for each $x_i \in X'$, such that, for the restriction θ that maps each variable $x_i \in X'$ to c_i, the following is satisfied: There exists a Boolean circuit $C' \equiv C|_\theta$, such that,*

$$\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5 \cdot |X'|.$$

Before proving Lemma 4.3, let us show how it is used to prove our lower bound.

LEMMA 4.4. *Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function, such that, $k = o(n)$. Then,*

$$Size(F) \geq 4.5 \cdot n - o(n).$$

PROOF. Let C be a Boolean circuit that computes F. We generate a sequence of Boolean circuit $C_0, \ldots, C_l$ by iteratively applying Lemma 4.3 to C. (Note that this is possible by Proposition 3.3). More formally, we have $C_0 = C$ and C_{i+1} is obtained from C_i by applying Lemma 4.3. We stop when the number of input-variables remaining is smaller than $k+5$. By Lemma 4.3,

$$\mathsf{SD}(C) \geq \mathsf{SD}(C_l) + 4.5 \cdot n - o(n).$$

By Proposition 4.2, we can assume that $Degeneracy(C) \leq k$. Therefore,

$$Size(C) \geq 4.5 \cdot n - o(n).$$

■

Recall that in Lemma 3.5, we proved that for the Boolean function $G : \{0,1\}^{\tilde{n}} \rightarrow \{0,1\}$ (defined in Section 3), there exists a restriction φ, such that, $G|_\varphi$ is (n,k)-Strongly-Two-Dependent where $\tilde{n} = n + O((\log n)^2)$ and $k = O(logn)$.

COROLLARY 4.5. *$Size(G) \geq 4.5 \cdot n - o(n) = 4.5 \cdot \tilde{n} - o(\tilde{n})$.*

4.3 Proof of Lemma 4.3

The proof of Lemma 4.3 is quite long, since it requires analysis of many different cases (and sub-cases). Nevertheless, the proof for each one of the different cases will be quite similar. More specifically, we partition the different possibilities for connections in the circuit into cases. In each case, we map one or two input-variables to constants in $\{0,1\}$. We then use the argument of Proposition 4.1 to remove some specific gate-nodes from the circuit.

We then calculate the difference in the SD measure between the original circuit and the modified restricted circuit. We call this difference: the number of SD units removed. The number of SD units removed is the number of gate-nodes removed minus half the change in the degeneracy measure. We will show in each of the different cases that the number of SD units removed is at least 4.5 times the number of input-variables mapped to a constant. The degeneracy measure might have changed in the following cases: (1) if an input-variable of degree one was mapped to a constant, (2) if the degree of an input-variable was changed from one, (3) if the degree of an input-variable was changed to one. More specifically, when applying Proposition 4.1 we count the change in the SD measure as follows:

1. Let v be a gate-node that was removed. We count v as one SD unit that was removed.
2. If we know that the degree of an input-variable x_i was changed from a number greater than one to one, we count it as 0.5 SD unit that was removed (since the degeneracy of the circuit was increased by one).
3. If the degree of an input-variable x_i was possibly changed from a number greater than one to one, we count it as -0.5 SD unit that was removed (i.e., 0.5 SD unit that was added), (since the degeneracy of the circuit might have decreased by one).
4. If we mapped to a constant an input-variable x_i, whose degree was possibly one, we count it as -0.5 SD unit that was remove (since the degeneracy of the circuit might have decreased by one).

We only count the change in the SD measure caused by the application of Proposition 4.1. Note that the circuit obtained may contain some of the degenerate cases of Proposition 4.2. Nevertheless, Proposition 4.2 removes all such cases without increasing the SD measure.

We are now ready to prove **Lemma 4.3**.

PROOF. Let $F : \{0,1\}^n \rightarrow \{0,1\}$ be an (n,k)-Strongly-Two-Dependent Boolean function and assume that $n-k \geq 5$. Let C be a Boolean circuit that computes F. By Proposition 4.2, let us assume that C does not contain any of the degenerate cases of Proposition 4.2. Let v_1 be a gate-node, such that, $Depth(v_1) = Depth(C) - 1$. That is, the depth of v_1 is the maximal possible depth for a gate-node in C. Therefore, $Right_C(v_1)$, $Left_C(v_1)$ are both input-variables. Without loss of generality, assume that x_1,x_2 are the two input-variables, such that, $x_1 = Left_C(v_1)$ and $x_2 = Right_C(v_1)$. By Proposition 3.11, $Degree_C(x_1) \geq 2$ and $Degree_C(x_2) \geq 2$. Let us partition all the possibilities for connections of x_1, x_2 into the following cases:

1. $Degree_C(x_1) \geq 4$ or $Degree_C(x_2) \geq 4$. That is, either $|OUT_C(x_1)| \geq 4$ or $|OUT_C(x_2)| \geq 4$.
2. $Degree_C(x_1) = 3$ or $Degree_C(x_2) = 3$. That is, either $|OUT_C(x_1)| = 3$ or $|OUT_C(x_2)| = 3$.
3. $Degree_C(x_1) = 2$ and $Degree_C(x_2) = 2$.

Note that in all cases we will never map an input-variable of degree one to a constant. Let us analyze separately each one of these cases.

CASE 1. *$Degree_C(x_1) \geq 4$ or $Degree_C(x_2) \geq 4$.*

Without loss of generality, assume that $Degree_C(x_1) \geq 4$. That is, $|OUT_C(x_1)| \geq 4$. Recall that x_1 is directly connected to the node v_1 and $x_1 = Left_C(v_1)$. Hence, $v_1 \in OUT_C(x_1)$. Let v_2, v_3, v_4 be three other different gate-nodes in $OUT_C(x_1)$. Without loss of generality, assume that $x_1 = Left_C(v_2)$, $x_1 = Left_C(v_3)$, $x_1 = Left_C(v_4)$. By Proposition 3.12 and by Proposition 4.2, $|OUT_C(v_1)| \geq 1$ (since x_1 is directly connected to v_1). Let v_5 be a gate-node, such that, v_2 is directly connected to v_5. By Proposition 4.2, $v_5 \notin OUT_C(x_1)$. Hence, v_1, v_2, v_3, v_4, v_5 are five different gate-nodes.

Let θ be a restriction that maps x_1 to $Bl_C(v_1)$. We take C' to be identical to $C\mid_\theta$ and modify it according to the argument of Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1,v_2,v_3,v_4. Since x_1 was mapped to $Bl_C(v_1)$, v_1 is now a constant-node, and we can apply the argument of Proposition 4.1 to v_5. That is, we removed the gate-nodes v_1,v_2,v_3,v_4,v_5.

We will now count the number of SD units removed. Recall that in C the gate-nodes v_1, v_2, v_3, v_4, v_5 are all different. Hence, we count their removal as 5 SD units removed. Recall that $Degree_C(x_1) > 1$ (and hence its removal doesn't increase the SD measure). By Proposition 3.11, no input-variable of degree one is directly connected to v_1, v_2, v_3, v_4, since x_1 is directly connected to each one of them. Only $Right_C(v_5)$ might be an input-variable of degree 1. Therefore, we count this as -0.5 SD unit removed (since the degeneracy of the circuit might have decreased by one). Hence, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$.

Case 2. *Either $Degree_C(x_1) = 3$ or $Degree_C(x_2) = 3$.*

Without loss of generality, assume that $Degree_C(x_1) = 3$. That is, $|OUT_C(x_1)| = 3$. Recall that x_1 is directly connected to v_1 and $x_1 = Left_C(v_1)$. Hence, $v_1 \in OUT_C(x_1)$. Let v_2,v_3 be the other two different gate-nodes in $OUT_C(x_1)$. Without loss of generality, assume that $x_1 = Left_C(v_2)$, $x_1 = Left_C(v_3)$. By Proposition 3.12 and Proposition 4.2, $|OUT_C(v_1)| \geq 1$, $|OUT_C(v_2)| \geq 1$, $|OUT_C(v_3)| \geq 1$ (since x_1 is directly connected to v_1,v_2,v_3). Let v_4 be a gate-node, such that, $v_4 \in OUT_C(v_1)$. Without loss of generality, assume that $v_1 = Left_C(v_4)$. By Proposition 4.2, $v_4 \notin OUT_C(x_1)$. Hence, v_1,v_2,v_3,v_4 are four different gate-nodes.

Let us partition all the possibilities of connections of v_1,v_2, v_3 into the following cases:

1. $OUT_C(v_1) \cap OUT_C(v_2) \neq \phi$ or
   $OUT_C(v_1) \cap OUT_C(v_3) \neq \phi$ or
   $OUT_C(v_2) \cap OUT_C(v_3) \neq \phi$.

2. $OUT_C(v_1) \cap OUT_C(v_2) = \phi$ and
   $OUT_C(v_1) \cap OUT_C(v_3) = \phi$ and
   $OUT_C(v_2) \cap OUT_C(v_3) = \phi$.

Let us analyze separately each one of these cases.

Case 2.1. *Either $OUT_C(v_1) \cap OUT_C(v_2) \neq \phi$ or $OUT_C(v_1) \cap OUT_C(v_3) \neq \phi$ or $OUT_C(v_2) \cap OUT_C(v_3) \neq \phi$.*

We can assume this, because in this case we will not use the variable x_2 at all, and we will not use the fact that v_1 is of maximal depth (that is, the three gate-nodes v_1,v_2,v_3 are totally symmetric). Recall that $v_4 \in OUT_C(v_1)$. Without loss of generality, assume that v_4 is a gate-node, such that, $v_4 \in OUT_C(v_1) \cap OUT_C(v_2)$. Recall that $v_1 = Left_C(v_4)$. Hence, $v_2 = Right_C(v_4)$. Recall that by Proposition 3.12 and Proposition 4.2, $|OUT_C(v_3)| \geq 1$. Let v_5 be a gate-node, such that, $v_5 \in OUT_C(v_3)$. Without loss of generality, assume that $v_3 = Left_C(v_5)$. Note that v_5 is different from v_4, because of the way v_4 is connected. By Proposition 4.2, $v_5 \notin OUT_C(x_1)$. Recall that the gate-nodes v_1, v_2, v_3, v_4 are different. Hence, the gate-nodes v_1, v_2, v_3, v_4, v_5 are different.

We partition the possibilities for the values of $Bl_C(v_1)$, $Bl_C(v_2)$, $Bl_C(v_3)$ into the following two cases:

1. $Bl_C(v_2) = Bl_C(v_3)$ or $Bl_C(v_1) = Bl_C(v_3)$.

2. $Bl_C(v_1) = Bl_C(v_2)$.

Let us analyze separately each one of these cases.

Case 2.1.1. *$Bl_C(v_2) = Bl_C(v_3)$ or $Bl_C(v_1) = Bl_C(v_3)$.*

Without loss of generality, assume that $Bl_C(v_2) = Bl_C(v_3)$. Let θ be a restriction that maps x_1 to $Bl_C(v_2)$. We take C' to be identical to $C\mid_\theta$ and modify it according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1, v_2, v_3. Since x_1 was mapped to $Bl_C(v_2)$ and $Bl_C(v_2) = Bl_C(v_3)$, the gate-nodes v_2,v_3 are now constant-nodes. We now apply the argument of Proposition 4.1 to the gate-nodes v_4,v_5. That is, we removed the gate-nodes v_1,v_2,v_3,v_4,v_5.

We will now count the number of SD units removed. Recall that the gate-nodes v_1, v_2, v_3, v_4, v_5 are different. Hence, we count their removal as 5 SD units removed. Recall that $Degree_C(x_1) > 1$. By Proposition 3.11, no input-variable of degree one is directly connected to v_1, v_2, v_3, since x_1 is directly connected to each one of them. Recall that $v_1 = Left_C(v_4)$, $v_2 = Right_C(v_4)$, $v_3 = Left_C(v_5)$. Hence, only $Right_C(v_5)$ might be an input-variable of degree 1. Therefore, we count this as -0.5 SD unit removed (since the degeneracy of the circuit might decreased by one). Thus, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$.

Case 2.1.2. *$Bl_C(v_1) = Bl_C(v_2)$.*

Recall that $v_1 = Left_C(v_4)$. We first show that $OUT_C(v_4) \neq \phi$. Assume for the sake of contradiction that $OUT_C(v_4) \neq \phi$. Then, by Proposition 4.2, v_4 is the output-node. Let θ' be a restriction that maps x_1 to $Bl_C(v_1) = Bl_C(v_2)$. In $C\mid_{\theta'}$ the gate-nodes v_1,v_2 compute a constant function, since x_1 is now a constant-node labeled by $Bl_C(v_1) = Bl_C(v_2)$. Therefore, the gate-node v_4 computes a constant function, since the value of the function that v_4 computes depends only on the values of the functions that v_2,v_3 compute. This is a contradiction to the fact that by Proposition 3.3, $F\mid_{\theta'}$ is $(n-2,k)$-Strongly-Two-Dependent, since $n-k \geq 5$.

Let us analyze this case according to the possible content of $OUT_C(v_4)$.

Case 2.1.2.1. *Assume that $v_3 \in OUT_C(v_4)$.*

Let θ be a restriction that maps x_1 to $Bl_C(v_1)$. We take C' to be identical to $C\mid_\theta$ and modify it according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1, v_2. Since x_1 was mapped to $Bl_C(v_1)$ and $Bl_C(v_1) = Bl_C(v_2)$, the gate-nodes v_1,v_2 are now constant-nodes. Each one of the gate-nodes v_3,v_4 computes a function that depends only on the values of x_1,v_1,v_2.

Recall that x_1,v_1,v_2 are now constant-nodes. Therefore, we apply the argument of Proposition 4.1 to gate-node v_3 and then to v_4. That is, we modified v_3,v_4 into constant-nodes. Therefore, we apply the argument of Proposition 4.1 to gate-node v_5. That is, we removed the gate-nodes v_1,v_2,v_3,v_4,v_5.

We will now count the number of SD units removed. Recall that in C the gate-nodes v_1, v_2, v_3, v_4, v_5 are different. Hence, we count their removal as 5 SD units removed. By Proposition 3.11, no input-variable of degree one is directly connected to v_1, v_2, v_3, since x_1 is directly connected to each one of them. Recall that $v_1 = Left_C(v_4)$, $v_2 = Right_C(v_4)$, $v_3 = Left_C(v_5)$. Hence, only $Right_C(v_5)$ might be an input-variable of degree 1. Therefore, we count this as -0.5 SD unit removed (since the degeneracy of the circuit might have decreased by one). Thus, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$.

CASE 2.1.2.2. *Assume that* $v_3 \notin OUT_C(v_4)$.

Let v_6 be a gate-node , such that, $v_6 \in OUT_C(v_4)$. Without loss of generality, assume that $v_4 = Left_C(v_6)$. Note that v_6 is different from v_1,v_2 because a Boolean circuit is acyclic. Recall that the gate-nodes v_1, v_2, v_3, v_4 are different. Hence, the gate-nodes v_1, v_2, v_3, v_4, v_6 are different.

Let θ be a restriction that maps x_1 to $Bl_C(v_2)$. We take C' to be identical to $C\mid_\theta$ and modify according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1, v_2, v_3. Since x_1 was mapped to $Bl_C(v_1)$ and $Bl_C(v_1) = Bl_C(v_2)$, v_1,v_2 are now constant-nodes. We apply the argument of Proposition 4.1 to v_4. Since v_1,v_2 are constant-nodes v_4 is now a constant-node. We apply the argument of Proposition 4.1 to v_6. That is, we removed the gate-nodes v_1,v_2,v_3,v_4,v_6.

We will now count the number of SD units removed. Recall that in C v_1, v_2, v_3, v_4, v_6 are five different gate-nodes. Hence, we count their removal as 5 SD units removed. By Proposition 3.11, no input-variable of degree one is directly connected to v_1, v_2, v_3, since x_1 is directly connected to each one of them. Recall that $v_1 = Left_C(v_4)$, $v_2 = Right_C(v_4)$, $v_4 = Left_C(v_6)$. Hence, only $Right_C(v_6)$ might be an input-variable of degree 1. Therefore, we count this as -0.5 SD unit removed (since the degeneracy of the circuit might decreased by one). Thus, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$.

CASE 2.2. $OUT_C(v_1) \cap OUT_C(v_2) = \phi$ *and* $OUT_C(v_1) \cap OUT_C(v_3) = \phi$ *and* $OUT_C(v_2) \cap OUT_C(v_3) = \phi$.

Recall that $v_1 = Left_C(v_4)$. By Proposition 3.12 and Proposition 4.2, $|OUT_C(v_2)| \geq 1$ and $|OUT_C(v_3)| \geq 1$, since x_1 is directly connected to v_2,v_3. Let v_8,v_9 be gate-nodes, such that, $v_8 \in OUT_C(v_2)$, $v_9 \in OUT_C(v_3)$. Without loss of generality, assume that $v_2 = Left_C(v_8)$, $v_3 = Left_C(v_9)$. By Proposition 4.2, $OUT_C(v_2) \cap OUT_C(x_1) = \phi$ and $OUT_C(v_3) \cap OUT_C(x_1) = \phi$. Recall that v_1,v_2,v_3,v_4 are four different gate-nodes. Hence, v_1,v_2,v_3,v_4 and the gate-nodes in $OUT_C(v_2)$,$OUT_C(v_3)$ are all different gate-nodes.

Note there exist two different constants $i_1, i_2 \in \{1, 2, 3\}$, such that, $h = Bl_C(v_{i_1}) = Bl_C(v_{i_2})$. Let θ be a restriction that maps x_1 to h. We take C' to be identical to $C\mid_\theta$ and modify it according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1, v_2, v_3. Since x_1 was mapped to $Bl_C(v_{i_1})$ and $Bl_C(v_{i_1}) = Bl_C(v_{i_2})$, the gate-node v_{i_1},v_{i_2} are now constant-nodes. We apply the argument of Proposition 4.1 to the gate-nodes in $OUT_C(v_{i_1})$,$OUT_C(v_{i_2})$. That is, we removed the gate-nodes v_1,v_2,v_3 and any other gate-node in $OUT_C(v_{i_1})$,$OUT_C(v_{i_2})$.

We will now count the number of SD units removed. Recall that in C the gate-nodes v_1, v_2, v_3 and the gate-nodes in $OUT_C(v_{i_1})$,$OUT_C(v_{i_2})$ are different. Hence, we count their removal as 5 SD units removed. Recall that $Degree_C(x_1) > 1$. By Proposition 3.11, no input-variable of degree one is directly connected to v_1, v_2, v_3. By Proposition 3.11, no input-variable of degree one is directly connected to v_4, since v_1 is directly connected to v_4 and $|IN_C(v_1)| < n - k$ (because $IN_C(v_1) = \{x_1, x_2\}$ and $n - k > 5$). Recall that $v_1 = Left_C(v_4)$, $v_2 = Left_C(v_8)$, $v_3 = Left_C(v_9)$. Hence, only $Right_C(v_8)$,$Right_C(v_9)$ might be input-variables of degree one. Assume that $|OUT_C(v_2)| \geq 2$ or $|OUT_C(v_3)| \geq 2$ or at least one of the nodes $Right_C(v_8)$,$Right_C(v_9)$ is not an input-variable of degree one. Then, for all possible values of i_1,i_2, we have that $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$, since if $i_1 = 2$,$i_2 = 3$ and we removed -1 SD unit because $Right_C(v_8)$,$Right_C(v_9)$ are both input-variables of degree one then we removed at least another 0.5 SD unit, since $|OUT_C(v_2) \cup OUT_C(v_3)| \geq 3$.

We will now prove that it cannot be the case that $|OUT_C(v_2)| = 1$ and $|OUT_C(v_3)| = 1$, and $Right_C(v_8)$, $Right_C(v_9)$ are input-variables of degree one. For the sake of contradiction assume that the above happens. Without loss of generality, assume that x_3,x_4 are input-variables, such that, $x_3 = Right_C(v_8)$,$x_4 = Right_C(v_9)$. Let θ' be a restriction that maps x_2 to $Br_C(v_1)$, x_3 to $Br_C(v_8)$ and x_4 to $Br_C(v_9)$. Observe that in $C\mid_{\theta'}$ all the paths from x_1 contain the gate-nodes v_1, v_8, v_9 (since $Degree_C(x_1) = 3$), and by the choice of θ' we know that v_1,v_8,v_9 all compute a constant function. This is a contradiction to Proposition 3.10, since θ' maps less than $n - k$ input-variables to constants, (because $n - k \geq 5$).

CASE 3. $Degree_C(x_1) = 2$ *and* $Degree_C(x_2) = 2$.

By Lemma 3.8, it cannot be the case that $OUT_C(x_1) = OUT_C(x_2)$ (i.e., and x_1,x_2 are not directly connected to the same two gate-nodes). Therefore, $OUT_C(x_1) \cap OUT_C(x_2) = \{v_1\}$. Recall that $x_1 = Left_C(v_1)$, $x_2 = Right_C(v_1)$. Let v_2, v_3 be the other two different gate-nodes, such that, $v_2 \in OUT_C(x_1)$ and $v_3 \in OUT_C(x_2)$. That is, v_1,v_2,v_3 are three different gate-nodes. Without loss of generality, assume that $x_1 = Left_C(v_2)$, $x_2 = Left_C(v_3)$. By Proposition 3.12 and by Proposition 4.2, $|OUT_C(v_1)| \geq 1$, since x_1 is directly connected to v_1. Let v_4 be a gate-node, such that, $v_4 \in OUT_C(v_1)$ and $Depth_C(v_4) = Depth(C) - 2$. Without loss of generality, assume that $v_1 = Left_C(v_4)$. By Proposition 4.2, $v_4 \notin OUT_C(x_1)$ and $v_4 \notin OUT_C(x_2)$. Hence, v_1,v_2,v_3,v_4 are four different gate-nodes.

Let us partition the possibilities for connections of v_1,v_2,v_3, v_4 into the following cases:

1. $Degree_C(v_1) \geq 2$ (i.e., $|OUT_C(v_1)| \geq 2$).
2. $Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and either $v_2 = Right_C(v_4)$ or $v_3 = Right_C(v_4)$.
3. $Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and $Right_C(v_4)$ is an input-variable.
4. $Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and $Right_C(v_4)$ is a gate-node different than v_2,v_3.

Let us analyze separately each one of these cases.

CASE 3.1. $Degree_C(v_1) \geq 2$ *(i.e.,* $|OUT_C(v_1)| \geq 2$*).*

Let $v_5 \in OUT_C(v_1)$ be a gate-node that is different from v_4. Without loss of generality, let $v_1 = Left_C(v_5)$. By the same reasoning as for v_4 (i.e., by Proposition 4.2), the gate-node v_5 is different from v_2,v_3. Thus, v_1,v_2,v_3,v_4,v_5 are five different gate-nodes.

Let θ be a restriction that maps x_1 to $Bl_C(v_1)$. We take C' to be identical to $C \mid_\theta$ and modify it according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 to each one of the gate-nodes v_1,v_2. Since x_1 was mapped to $Bl_C(v_1)$, v_1 is now a constant-nodes. Hence, we apply the argument of Proposition 4.1 to v_4,v_5.

We will now count the number of SD units removed. Recall that v_1,v_2,v_4,v_5 are four different gate-nodes. Hence, we count their removal as 4 SD units removed. Recall that $Degree_C(x_1) > 1$. By Proposition 3.11, no input-variable of degree one is directly connected to v_1,v_2, since x_1 is directly connected to each one of them. Recall that $v_1 = Left_C(v_4)$, $v_1 = Left_C(v_5)$. Hence, by Proposition 3.11, no input-variable of degree one is directly connected to v_1,v_2, since v_1 is directly connected to each one of them and $IN_C(v_1) < n - k$ (since $IN_C(v_1) = \{x_1, x_2\}$ and $n - k \geq 5$). Thus, the degree of any input-variable did not change from one by the applying the argument of Proposition 4.1. Recall that v_1,v_2,v_3,v_4,v_5 are five different gate-nodes. Hence, v_3 was not removed. Recall that in C' the node v_1 is a constant-node. Therefore, in C' the input-variable x_2 is directly connected only to v_3. We count the decrease in the degree of x_2 as 0.5 SD unit removed. Hence, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$

Case 3.2. *$Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and either $v_2 = Right_C(v_4)$ or $v_3 = Right_C(v_4)$.*

Without loss of generality, assume that $v_2 = Right_C(v_4)$. Let us prove that this case cannot occur in C. Recall that $Depth_C(v_1) = Depth(C') - 1$. Therefore, since $Degree_C(v_1) = 1$, we have $Depth_C(v_4) = Depth(C') - 2$. Hence, $Depth_C(v_2) = Depth(C') - 1$. Implying, that $Right_C(v_2)$ is an input-variable. Without loss of generality, assume that x_3 is the input-variable , such that, $x_3 = Right_C(v_2)$. Let θ' be a restriction that maps x_2 to $Br_C(v_1)$ and x_3 to $Br_C(v_2)$. Observe that in $C \mid_{\theta'}$ all the paths from x_1 contain the gate-nodes v_1, v_2 and by the choice of θ', the gate-nodes v_1,v_2 both compute a constant function. This is a contradiction to Proposition 3.10, since θ' maps less than $n - k$ input-variables to constants (because $n - k \geq 5$).

Case 3.3. *$Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and $Right_C(v_4)$ is an input-variable.*

Without loss of generality, let x_4 be the input-variable, such that, $x_4 = Right_C(v_4)$. By Proposition 3.11 and Proposition 3.12, $|OUT_C(x_4)| \geq 2$, since $|IN_C(v_1)| < n - k$ (because $IN_C(v_1) = \{x_1, x_2\}$ and $n - k \geq 5$). Without loss of generality, assume that for each gate-node $v_i \in OUT_C(v_4)$, we have $v_4 = Left_C(v_i)$. Then, by Proposition 3.11, for each $v_i \in OUT_C(v_4)$, $Right_C(v_i)$ is not an input-variable of degree one, since $|IN_C(v_4)| < n - k$ (because $IN_C(v_4) = \{x_1, x_2, x_4\}$ and $n - k \geq 5$). By Proposition 4.2 x_4 is directly connected to a gate-node v_6, such that, $v_6 \notin OUT_C(v_4)$. By Proposition 3.11, no input-variable of degree one is directly connected to v_6, since x_4 is directly connected to v_6. Recall that v_1,v_2,v_3,v_4 are four different gate-nodes, v_6 is different than v_1, because it is connected differently.

Let us prove that v_6 is different from v_2,v_3. Assume for the sake of contradiction that v_6 and v_2 are the same gate-node. Recall that $x_1 = Left_C(v_2)$. Hence, $x_4 = Right_C(v_2)$. Let θ' be a restriction that maps x_2 to $Br_C(v_1)$ and x_4 to $Br_C(v_2)$. Recall that $x_2 = Right_C(v_1)$. Observe that in $C \mid_{\theta'}$ all the paths from x_1 contain the gate-nodes v_1,v_2 and by the choice of θ', v_1,v_2 both compute a constant function. This is a contradiction to Proposition 3.10, since θ' maps less than $n - k$ input-variables to constants (because $n-k \geq 5$). Thus, v_1,v_2,v_3,v_4,v_6 are five different gate-nodes. By Proposition 3.12 and Proposition 4.2, $|OUT_C(v_4)| \geq 1$, since $|IN_C(v_1)| < n - k$ (because $IN_C(v_1) = \{x_1, x_2\}$ and $n - k \geq 5$). By the way v_1 is connected $v_1 \notin OUT_C(v_4)$. Thus, v_1,v_4,v_6 and the gate-nodes in $OUT_C(v_4)$ are all different.

Let θ be a restriction that maps the input-variable x_4 to $Br_C(v_4)$. We take C' to be identical to $C \mid_\theta$ and modify according to Proposition 4.1. That is, we apply the argument of Proposition 4.1 on v_4. Since x_4 was mapped to $Bl_C(v_4)$, v_4 is now a constant-node. Therefore, we apply the argument of Proposition 4.1 on v_1,v_6 and on each gate-node in $OUT_C(v_4)$. That is, we removed at least four gate-nodes v_1,v_4,v_6 and the gate-nodes in $OUT_C(v_4)$.

We will now count the number of SD units removed. Recall that v_1,v_4,v_6 and the gate-nodes in $OUT_C(v_4)$ are all different and that $OUT_C(v_4)$ is not empty. Hence, we count their removal as 4 SD units removed if $v_2 \notin OUT_C(v_4)$ or $v_3 \notin OUT_C(v_4)$, and as 5 SD units removed if both v_2,v_3 are in $OUT_C(v_4)$ Recall that no input-node of degree one is directly connected to v_1,v_4. By Proposition 3.11 no input-variable of degree one is directly connected to v_6, since x_4 is connected directly to v_6. By similar reasoning no input-variable of degree one is directly connected to v_2,v_3. Also by Proposition 3.11, no input-variable of degree one is directly connected to a gate-node in $OUT_C(v_4)$,since v_4 is connected directly each it gate-node in $OUT_C(v_4)$ and $|IN_C(v_4)| < n - k$ (because, $IN_C(v_1) = \{x_1, x_2, x_4\}$ and $n - k \geq 5$). Thus, the degree of any input-variable did not change from one, by the applying the argument of Proposition 4.1. Assume $v_2 \notin OUT_C(v_4)$ or $v_3 \notin OUT_C(v_4)$. Without loss of generality, assume that $v_2 \notin OUT_C(v_4)$. Then, since v_1,v_2,v_3,v_4,v_6 are five different gate-nodes , v_2 was not removed. Therefore, in C' the degree of x_1 is one . Hence, we count this as 0.5 SD unit removed (since th the degeneracy increased by one). Hence, $\mathsf{SD}(C) \geq \mathsf{SD}(C') + 4.5$

Case 3.4. *$Degree_C(v_1) = 1$ (i.e., $|OUT_C(v_1)| = 1$) and that $Right_C(v_4)$ is a gate-node.*

This is the most complicated case. It requires the analysis of many subcases. Due to space limitation we omit the analysis.

■

5. REFERENCES

[1] C.E. Shannon. The synthesis of two-terminal switching circuits, *Bell Systems Tech. J.*, vol 28, pages 59–98, 1949.

[2] U. Zwick. A 4n lower bound on the combinatorial complexity of certain symmetric Boolean functions over the basis of unate dyadic Boolean functions., *SIAM Journal on Computing*, vol 20, pages 499–505, 1991.