

Lower bound techniques for Formula Size and monotone Circuit Depth

*A thesis submitted in partial fulfillment of the requirements
for the award of the degree of*

Master of Science

by

Karteek Sreenivasaiah

Junior Research Fellow
Theoretical Computer Science
The Institute of Mathematical Sciences
Chennai - 600113

Homi Bhabha National Institute



April, 2010

Certificate

Certified that the work contained in the thesis entitled LOWER BOUND TECHNIQUES FOR FORMULA SIZE AND MONOTONE CIRCUIT DEPTH , by **KartEEK Sreenivasaiah**, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Meena Mahajan

Theoretical Computer Science Group
The Institute of Mathematical Sciences, Chennai

ACKNOWLEDGEMENTS

I first thank my advisor Prof. Meena Mahajan for her continuous support, guidance and impeccable proof-reading throughout the course of my master's degree. I thank all the faculty of the theory group here at IMSc for having been helpful and encouraging. I take this opportunity to also thank my friends and family for having made my stay here at IMSc lots of fun.

Abstract

The circuit depth required to compute a function corresponds directly to the running time required by a parallel computer to compute it. So researchers have always been interested in lower bounding circuit depth. When trying to get lower bounds on circuits seemed difficult, it became natural to look at formulas. In this thesis, we first look at different techniques for lower bounding Formula Size, each of which attack different aspects of how a boolean function works. In the process, we study a super quadratic lower bound for an explicit function. Further, we focus our attention on some results proved in the monotone world. We look at a way to lower bound monotone Formula Size using rank arguments. We explore how Communication Complexity relates to circuit depth and can be used as a tool to give lower bounds for circuit depth. And finally, we study three main results which use Communication Complexity to prove non-trivial lower bounds to monotone circuit depth.

Contents

| | | |
|----------|--|-----------|
| 1 | Formula Size | 1 |
| 1.1 | Introduction | 1 |
| 1.2 | Introduction to the model | 1 |
| 1.2.1 | Circuit | 1 |
| 1.2.2 | Formula | 2 |
| 1.3 | Definitions and notation | 2 |
| 1.4 | Lower Bound Techniques | 4 |
| 1.4.1 | Using Random Restrictions | 4 |
| 1.4.2 | Partitioning variables | 8 |
| 1.4.3 | Behaviour of function on close inputs | 12 |
| 2 | Monotone Formula Size | 16 |
| 2.1 | Introduction | 16 |
| 2.2 | Rank Lower Bound | 16 |
| 2.3 | Lower bound for dense graphs of girth at least 5 | 17 |
| 3 | Communication Complexity and Circuit Depth | 21 |
| 3.1 | Introduction | 21 |
| 3.2 | Games on Relations | 22 |
| 3.3 | Karchmer-Wigderson game | 23 |
| 3.4 | Yao's game | 28 |

| | | |
|----------|---|-----------|
| 3.5 | Non-determinism | 32 |
| 4 | Lower Bounds using Communication Complexity | 37 |
| 4.1 | Introduction | 37 |
| 4.2 | Direct Sum | 38 |
| 4.3 | FORK Game | 46 |
| 4.4 | A general technique for a $\log^2 n$ lower bound to monotone circuit depth | 52 |
| | Bibliography | 55 |
| | Appendix | 58 |

Chapter 1

Formula Size

1.1 Introduction

In this chapter, we introduce two computation models: Circuits and Formulas. Both of these are parallel models, meaning, they give us a way to parallelize computation of a function by recognizing those sub-functions that can be computed simultaneously. There are many resources related to circuits and formulas which raise natural questions in complexity. Some of these are size, depth, fan-in. The focus of this chapter, however, will be mainly on the size of formulas. Many of the techniques in this chapter have been covered in [Juk09] and [BS90].

1.2 Introduction to the model

1.2.1 Circuit

A Circuit is a computation model defined as follows:

Let \mathbb{B} be a set of boolean functions. A circuit on basis \mathbb{B} is a sequence of boolean functions g_1, g_2, \dots, g_m such that the first g_1, \dots, g_n are simply the input variables x_1, \dots, x_n and the remaining g_{n+1}, \dots, g_m are functions from

\mathbb{B} whose inputs are from the outputs of previously computed functions. The output of one or more of the g_i 's is called the output of the circuit. The g_i 's are also called gates. Another way of looking at a circuit: A circuit is a Directed Acyclic Graph where the vertices with fan-in 0 are the inputs to the circuit, and each other vertex computes a function from \mathbb{B} on its inputs (which it gets from other vertices). One or more vertices are designated as output vertices. Here, each vertex is called a gate.

A boolean function f on n variables is a monotone boolean function if $\forall x, y \in \{0, 1\}^n, x \leq y$ (pointwise) $\implies f(x) \leq f(y)$. A circuit on the basis $\{\wedge, \vee\}$ is called a monotone circuit. It can be seen that a monotone circuit can only compute monotone functions.

1.2.2 Formula

A Formula over a basis \mathbb{B} is simply a circuit over basis \mathbb{B} where each gate has a fan-out exactly 1. Hence a formula can be seen as a Tree, where the leaves are the input gates. A formula over the basis $\{\wedge, \vee\}$ is called a monotone formula.

The only difference between Circuits and Formulas is that in Circuits, we can reuse a value computed at a gate several times by feeding it into other gates. On the other hand, a formula does not allow for such reuse.

1.3 Definitions and notation

In the whole of this document, we will use the following definitions and notation.

\wedge_n and \vee_n denote boolean AND and boolean OR respectively of fan-in n . \wedge is short for \wedge_2 and \vee is short for \vee_2 .

$L_{\mathbb{B}}(f)$ denotes the leaf-size of the smallest leaf-size formula on basis \mathbb{B} computing the function f . $L(f)$ is short for $L_{\{\wedge, \vee, \sim\}}(f)$. $L^+(f)$ is short for $L_{\{\wedge, \vee\}}(f)$. $L_{\mathcal{U}}(f)$ is defined similarly on the universal binary basis: $\mathcal{U} = \{g \mid g : \{0, 1\}^2 \rightarrow \{0, 1\}\}$.

$d(f)$ denotes the depth of the minimum depth formula computing the boolean function f . $d^+(f)$ denotes the depth of the minimum depth monotone formula computing monotone boolean function f .

\mathcal{R}_k denotes the set of all restrictions that leave k variables unassigned. A random restriction $\rho \in \mathcal{R}_k$ is a restriction picked uniformly at random from the set \mathcal{R}_k . Here, any $\rho \in \mathcal{R}_k$ is assumed to work as follows: Chose k variables uniformly at random, and assign values to the remaining variables uniformly at random. $f \upharpoonright_{\rho}$ denotes the function obtained by restricting the variables of f using ρ .

A rectangle is just a Cartesian product $X \times Y$ of two disjoint sets of binary strings X and Y . $S \times T$ is called a sub-rectangle if $S \subseteq X$ and $T \subseteq Y$. A monochromatic rectangle $X \times Y$ is a rectangle such that there exists an i with the property that $\forall x \in X, \forall y \in Y, x_i \neq y_i$. Here i is also called the color of the rectangle. A rectangle is called monotone-monochromatic if $\exists i$, such that $\forall x \in X, \forall y \in Y, x_i = 0$ and $y_i = 1$.

For a boolean function f , S_f denotes the rectangle $f^{-1}(0) \times f^{-1}(1)$. $D(S_f)$ denotes the minimum number of parts required to partition S_f into monochromatic sub-rectangles. $D^+(S_f)$ denotes the minimum number of parts required to partition S_f into monotone monochromatic sub-rectangles. For a boolean function f over $2n$ elements, M_f denotes the matrix $\{0, 1\}^n \times \{0, 1\}^n$ where the entry corresponding the row x and column y is 1 if $f(x, y) = 1$, and 0 otherwise.

For a matrix A and a rectangle R , A_R denotes the matrix obtained by setting to 0 the entries $(i, j) \notin R$ of the matrix A .

A sub-function of a boolean function $f(X)$ on $Y \subseteq X$ is a function obtained from f by setting all the variables in $X - Y$ to constants. $S_Y(f)$ is the set of all functions g such that either g , or \bar{g} , or both are a sub-function of f on Y . $\text{Size}_Y(F)$ denotes the number of leaves of the formula F that are labelled with a variable in Y .

1.4 Lower Bound Techniques

In what follows, we will look at techniques to lower bound the leaf size of a formula computing a boolean function f .

1.4.1 Using Random Restrictions

Theorem 1.1 (Subbotovskaya). : *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$. $\forall k, 1 \leq k \leq n$, $\exists \rho \in \mathcal{R}_k$, such that*

$$L(f \upharpoonright_\rho) \leq 4 \cdot \left(\frac{k}{n}\right)^{3/2} \cdot L(f) \quad (1.1)$$

Proof. In this proof, whenever we use the word “formula”, we refer to a formula over the basis $\{\wedge, \vee, \sim\}$. Let F be a formula for f which is optimal in leaf-size. Let s denote $L(f)$. We prove the existence of a random restriction with the required inequality by carefully observing how the random restriction affects the leaf-size at each stage. Pick a random restriction $\rho \in \mathcal{R}_k$. ρ picks k variables to leave unassigned, and chooses a variable randomly from the remaining variables and assigns a 0 or 1 to it uniformly and independently at random. ρ does this one by one to each of the remaining variables. Let the process of picking a variable and assigning a value to it be called a “stage”. Since $\rho \in \mathcal{R}_k$, there are totally $n - k$ stages.

Suppose ρ chooses the variable x_i to assign 0 or 1, then the input gates labelled by x_i or \bar{x}_i will now disappear. The expected number of input gates that disappear is $\left(\frac{s}{n}\right)$.

Setting the above input gates to constants would reduce the leaf-size even further because these constants can be propagated upwards to eliminate more gates. i.e., for each input gate g labelled by x_i which was fixed to a constant, suppose g was connected to an \wedge gate $g \wedge H$, where H is a sub-formula of F . H does not contain a leaf labelled x_i or \bar{x}_i because if it did, F would not be an optimal formula for f . Now, since x_i was assigned 0 or 1 with a probability of $1/2$, the expected number of leaves that disappear due to this kind of propagation is $\left(\frac{1}{2} \cdot \frac{s}{n}\right)$.

Hence the expected number leaves that disappear due to the two reasons above is at least:

$$\frac{s}{n} + \frac{s}{2n} = \frac{3s}{2n}$$

So after the first stage the formula size decreases by this amount yielding a new formula of expected leaf-size:

$$\mathbb{E} [L(f \upharpoonright_{\rho(1 \text{ stage})})] = s \cdot \left(1 - \frac{3}{2n}\right) \leq s \cdot \left(1 - \frac{1}{n}\right)^{\frac{3}{2}}$$

Hence after $(n - k)$ stages, the formula size would be:

$$\begin{aligned} \mathbb{E} [L(f \upharpoonright_{\rho})] &\leq s \cdot \left(1 - \frac{1}{n}\right)^{\frac{3}{2}} \cdot \left(1 - \frac{1}{n-1}\right)^{\frac{3}{2}} \cdot \left(1 - \frac{1}{n-2}\right)^{\frac{3}{2}} \cdots \left(1 - \frac{1}{n-k}\right)^{\frac{3}{2}} \\ &= s \cdot \left(\frac{k}{n}\right)^{3/2} \end{aligned}$$

Now we use Markov's inequality taking $L(f \upharpoonright_{\rho})$ as a random variable. By Markov's inequality, $\Pr \left[L(f \upharpoonright_{\rho}) \geq 4 \cdot s \cdot \left(\frac{k}{n}\right)^{3/2} \right] < 1/4$. So with probability at least $3/4$, a random restriction from \mathcal{R}_k will satisfy the inequality in the claim. Hence there exists a random restriction satisfying the required

inequality. □

To get a non-trivial lower bound using this technique, the approach would be to find a random restriction ρ such that $L(f \upharpoonright_\rho)$ can be easily lower-bounded by a value greater than 0. We will see one simple example of this, followed by a more involved example proved by Andreev in the year 1987.

Example 1.2 (Parity). *Let $f = \oplus(x_1, x_2, \dots, x_n)$. By above theorem, $\exists \rho \in \mathcal{R}_1$ such that*

$$L(\oplus_n \upharpoonright_\rho) \leq 4 \cdot \left(\frac{1}{n}\right)^{3/2} \cdot L(\oplus_n)$$

Now, we observe that $L(\oplus_n \upharpoonright_\rho) \geq 1$. And hence

$$L(\oplus_n) = \Omega(n^{3/2})$$

Example 1.3 (Andreev's function). *Now we look at a function constructed by Andreev for which we can show an $n^{5/2}$ lower bound using Subbotovskaya's technique. Fix $b = \log n$ and $c = n/b$. Let $g : \{0, 1\}^b \rightarrow \{0, 1\}$, Now we define a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ as the composition of g with \oplus_n . If we look at the variable set $X = \{x_1, x_2, \dots, x_n\}$ as a $b \times c$ matrix:*

$$\begin{bmatrix} x_{11} & x_{12} & \cdot & \cdot & \cdot & x_{1c} \\ x_{21} & x_{22} & \cdot & \cdot & \cdot & x_{2c} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{b1} & x_{b2} & \cdot & \cdot & \cdot & x_{bc} \end{bmatrix}$$

then, define function f as follows:

$$f = g \left(\bigoplus_{i=1}^c x_{1i}, \bigoplus_{i=1}^c x_{2i}, \dots, \bigoplus_{i=1}^c x_{bi} \right)$$

Let $\rho \in \mathcal{R}_k$. We first want to find the value of k which would make ρ leave at least one variable in each of the b rows unassigned with a probability greater than $3/4$. We observe: For any fixed variable x ,

$$\Pr[\rho \text{ leaves } x \text{ unassigned}] = \frac{\binom{n-1}{k-1}}{\binom{n}{k}} = \frac{k}{n}$$

Therefore, $\Pr[\rho \text{ fixes } x] = 1 - \frac{k}{n}$. And so, for any fixed row r ,

$$\Pr[\rho \text{ fixes all } x \text{ in row } r] = \left(1 - \frac{k}{n}\right)^c$$

By Union bound,

$$\Pr[\text{all variables in some row are fixed by } \rho] \leq b \left(1 - \frac{k}{n}\right)^c \leq b \cdot e^{-kc/n}$$

So if we take $k = \lceil b \ln 4b \rceil$, then the above probability will be $< 1/4$. And hence the probability that ρ leaves at least one variable in each row unassigned $\geq 3/4$.

Now define a boolean function $A_n : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ as follows: Let the first n variables be X . And let the remaining n variables be Y . Then, $A_n(X, Y) = f_g(Y)$ where g is the function on $\log n$ whose truth table is given by X .

Lemma 1.4 (Andreev). $L(A_n) = \Omega(n^{5/2})$

Proof. By definition of A_n , we have $\forall g, L(A_n) \geq L(f_g)$. We know from Subbotovskaya's theorem that there exists a restriction which will satisfy (1.1), with a probability of at least $3/4$. Hence, there exists a restriction

which satisfies (1.1) and leaves at least one variable in each row of matrix X unassigned. Let this restriction be ρ . We have:

$$L(f_g) \geq \frac{1}{4} \cdot \left(\frac{bc}{k}\right)^{\frac{3}{2}} \cdot L(f_g \upharpoonright_\rho) \geq \frac{1}{4} \cdot \left(\frac{bc}{k}\right)^{\frac{3}{2}} \cdot L(g) \quad (1.2)$$

By Shannon's argument, we know that complicated functions that require exponential size formulas exist (See the appendix for Shannon's theorem statement. For full proof, see [Sha49], [Vol99]). We take g to be one such complicated function. So $L(g) = \Omega\left(\frac{2^b}{\log(b)}\right)$. And hence, we get:

$$L(f_g) \geq \frac{1}{4} \left(\frac{bc}{k}\right)^{3/2} \cdot \frac{2^b}{\log b} = \Omega\left(\frac{n^{5/2}}{\log n \cdot \log \log n}\right)$$

□

A tighter analysis by Hastad in [Hås98] shows that $L(A_n) = \Omega(n^{3-o(1)})$.

1.4.2 Partitioning variables

The following version of Nechiporuk is from [BS90].

Theorem 1.5 (Nechiporuk). *: Let $f(X)$ be a boolean function on a set of n variables $X = \{x_1, x_2, \dots, x_n\}$. Fix a partition Y_1, Y_2, \dots, Y_m of X .*

$$L_u(f) \geq \sum_{i=1}^m \log_5(2c_i(f) + 1) \quad (1.3)$$

where $c_i(f)$ denotes the number of distinct subfunctions of f on Y_i .

Proof. Let F be an optimal leaf-size formula for f , then, $L_u(f) = \text{Size}_X(F)$. Since Y_1, Y_2, \dots, Y_m is a partition of X , we have:

$$L_u(f) = \sum_{i=1}^m \text{Size}_{Y_i}(F)$$

and by definition of $c_i(f)$, we have $|S_{Y_i}(F)| \geq c_i(f)$. So, it is sufficient to prove:

Lemma 1.6. *For any formula F and every variable set $Y \subseteq X$,*

$$2 \cdot S_Y(F) + 1 \leq 5^{\text{Size}_Y(F)}$$

Proof. Induction on leaf-size of formula F .

Base case: $F = x_i$ or $F = \bar{x}_i$. If $x_i \in Y$, then $S_Y(F) = 2$ (the two distinct sub-functions being x_i and \bar{x}_i), else $S_Y(F) = 2$ (the two distinct sub-functions being 0 and 1) and the claim holds.

Step: Let $F = F_1 * F_2$ where $*$ is a binary operation. By induction hypothesis, F_1 and F_2 satisfy the claim. Consider the following sets of boolean functions:

$$\begin{aligned} T &= \{g_1 * g_2 \mid g_1 \in S_Y(F_1) \text{ and } g_2 \in S_Y(F_1)\} \\ T' &= \{g \mid \bar{g} \in T\} \end{aligned}$$

Now, we observe that:

$$S_Y(f) \subseteq T \cup T' \cup S_Y(F_1) \cup S_Y(F_2)$$

And hence:

$$\begin{aligned} 2|S_Y(f)| + 1 &\leq 2 \cdot |T \cup T' \cup S_Y(F_1) \cup S_Y(F_2)| + 1 \\ &\leq 4 \cdot |S_Y(F_1)||S_Y(F_2)| + 2|S_Y(F_1)| + 2|S_Y(F_2)| + 1 \\ &= (2|S_Y(F_1)| + 1) \cdot (2|S_Y(F_2)| + 1) \\ &\leq 5^{\text{Size}_Y(F_1)} \cdot 5^{\text{Size}_Y(F_2)} \text{ from inductive hypothesis} \\ &= 5^{\text{Size}_Y(F)} \end{aligned}$$

□

□

Example 1.7 (Element Distinctness). ED_n is the *Element Distinctness* function where the input is m numbers from a range $\{1, 2, 3, \dots, m^2\}$. The function evaluates to a 1 if the m numbers are all distinct from one another, 0 otherwise. Each of the m numbers can be represented with $2 \log m$ bits. Hence the input size $n = 2m \log m$.

Lemma 1.8. $L(ED_n) = \Omega\left(\frac{n^2}{\log n}\right)$

Proof. Look at the input vector x representing m numbers as strings s_1, s_2, \dots, s_m of $2 \cdot \log m$ bits each. The partition is simply $Y_i = s_i$.

Claim. $c_i(ED_n) = \binom{m^2}{m-1} \geq \left(\frac{m^2}{m-1}\right)^{m-1} = 2^{\Theta(m \cdot \log m)}$

Proof. Consider the case when $i = 1$. Every way of setting Y_2, \dots, Y_m to distinct numbers (without importance to ordering) gives a different sub-function of ED_n . To see this, take any two different sets of $m - 1$ distinct numbers $A = \{a_2, a_3, \dots, a_m\}$ and $B = \{b_2, b_3, \dots, b_m\}$ as assignments to Y_2, \dots, Y_m (without importance to order). Since A and B are different, there is at least one i such that $a_i \in A$ and $a_i \notin B$. On such an a_i , the sub-function derived from A would output 0 when Y_1 is set to a_i , whereas the sub-function derived using B would output 1. Hence each way of choosing $m - 1$ numbers out of a range of m^2 numbers gives us a different sub-function. Since there was nothing special about the choice of $i = 1$, the same argument applies to all i , and the claim holds. □

Applying Nechiporuk's technique on this variable partition, and using the above claim, we get:

$$L_U(ED_n) \geq m \cdot \log \left(2 \binom{m^2}{m-1} + 1 \right) = \Omega \left(m \cdot \log \left(2^{\Theta(m \cdot \log m)} \right) \right) = \Omega \left(\left(\frac{n}{\log n} \right) \cdot n \right)$$

Hence the lemma follows. □

Example: Th_k^n We look at a function where Nechiporuk's technique does not give a good lower bound. The function f that we consider here will be the threshold function: The Threshold function Th_k^n , over n variables is defined as follows:

$$\text{Th}_k^n(x) = \begin{cases} 1 & \text{if number of 1s in } x \text{ is } \geq k \\ 0 & \text{Otherwise} \end{cases}$$

Consider a partition Y_1, Y_2, \dots, Y_m of the variable set with sizes s_1, s_2, \dots, s_m . Clearly $s_1 + s_2 + \dots + s_m = n$.

Claim. For any partition Y_i , $c_i(f) \leq s_i + 1$

Proof. A setting to variables outside a Y_i could force Th_k^n to be 1 by contributing more than $k-1$ 1s, or it could contribute less than k 1s and leave the remaining to be contributed by Y_i .

So, if $s_i \geq k$, then we would have at most k distinct sub-functions on Y_i , and they would be: $\text{Th}_k^{s_i}, \text{Th}_{k-1}^{s_i}, \text{Th}_{k-2}^{s_i}, \dots, \text{Th}_0^{s_i}$. In the other case where $s_i < k$, we would have at most s_i distinct sub-functions, and they are the following: $\text{Th}_0^{s_i}, \text{Th}_1^{s_i}, \text{Th}_2^{s_i}, \dots, \text{Th}_{s_i}^{s_i}$. Hence the proof. □

Lemma 1.9. $\sum_{i=1}^m \log(2c_i(f) + 1) = O(n)$

Proof. From the previous claim, we know that $s_i + 1$ is an over-estimate

of $c_i(f)$ and hence:

$$\begin{aligned}
\sum_{i=1}^m \log(2c_i(f) + 1) &\leq \sum_{i=1}^m \log(2(s_i + 1) + 1) \\
&\leq \sum_{i=1}^m \log(2(s_i + 2)) \\
&= m + \log((s_1 + 2)(s_2 + 2) \cdots (s_m + 2))
\end{aligned}$$

It can be shown that the product of m integers, when their sum is fixed, is the maximum when they are all equal. Since $m = O(n)$, $m + \log\left(\left(\frac{n+2m}{m}\right)^m\right) = m + m \cdot \log\left(\frac{n+2m}{m}\right) = O(n)$. However, The best known upper-bound for threshold over the universal basis is $n^{3.13}$ and was proved in [MSPZ92]. \square

1.4.3 Behaviour of function on close inputs

Theorem 1.10 (Krapchenko). : $\forall f : \{0, 1\}^n \rightarrow \{0, 1\}$ the following holds:

$$L(f) \geq D(S_f) \geq \frac{|Y|^2}{|S_f|} \quad (1.4)$$

where $Y = \{(x, y) \in S_f \mid x, y \text{ differ in exactly one position}\}$

Proof. We prove the theorem in two steps:

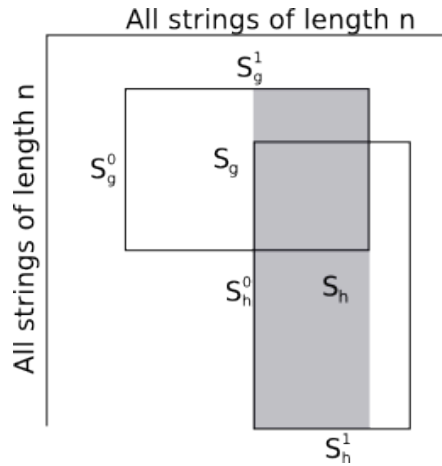
1. $L(f) \geq D(S_f)$
2. $D(S_f) \geq \frac{|Y|^2}{|S_f|}$

Lemma 1.11 (Rychkov's). : $L(f) \geq D(S_f)$

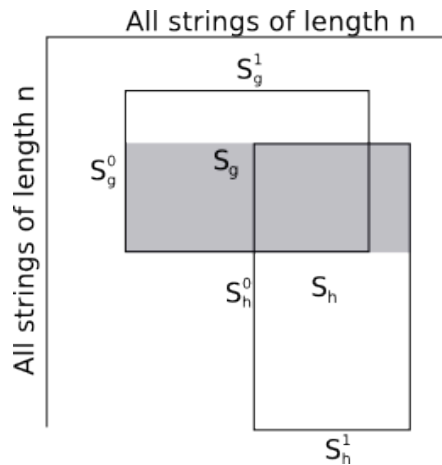
Proof. We need to show that for any boolean function f on n variables, S_f can be partitioned into at most $L(f)$ monochromatic parts.

We first make the following observation: Given rectangles $S_g = S_g^0 \times S_g^1$ and $S_h = S_h^0 \times S_h^1$ for two boolean functions g and h on n variables, the

rectangle corresponding to $f = g \wedge h$ is $S_f = \{S_g^0 \cup S_g^1\} \times \{S_h^1 \cap S_h^0\}$. In the following figure, S_f is the shaded area



The rectangle corresponding to S_f when $f = g \vee h$ is $S_f = \{S_g^0 \cap S_h^0\} \times \{S_h^1 \cup S_g^1\}$. The shaded area in the following figure represents S_f when $f = g \vee h$:



To prove the lemma, we use induction on Leaf size $L(f)$. The base case is when the minimal formula F that computes f has only one leaf, that is to say, $F = x_i$ or $F = \bar{x}_i$. In this case, the whole of S_f is monochromatic and hence the claim holds.

Inductive step: Suppose the topmost gate in F is an \wedge gate, then F can be written as $g \wedge h$ where g and h are the sub-functions being computed at either side of the \wedge gate. $L(g) + L(h) = L(f)$. Now we apply inductive hypothesis on g and h to get $D(S_g) \leq L(g)$ and $D(S_h) \leq L(h)$. We know from the observation above that we can get S_f from S_g and S_h . Since S_g can be covered by at-most $L(g)$ monochromatic sub-rectangles, and S_h can be covered by at most $L(h)$ many monochromatic sub-rectangles, we can cover S_f by $L(f) + L(g)$ sub-rectangles. The case when the topmost gate is an \vee is similar. \square

Observation 1.12. *The above proof works for monotone case as well. In the base case, $L^+(f) = 1$, F has just one leaf - x_i . And thus the rectangle is monotone monochromatic and the proof goes through. Induction step works exactly the same way. Hence the above proof also shows that $L^+(f) \geq D^+(S_f)$.*

We now prove the following property about the $|Y|$:

Lemma 1.13. *Let $R = R^0 \times R^1$ be a monochromatic rectangle, then $|R \cap Y|^2 \leq |R^0| \times |R^1|$*

Proof. Since R is monochromatic, there exists one position i such that for each $(x, y) \in R$, x and y differ in position i . x and y could differ in other positions as well, but surely at i , $x_i \neq y_i$. Hence for any given $x \in R^0$, there is exactly one $y \in R^1$ which differs from x in exactly one position, namely $x \oplus e_i$. Hence $|R \cap Y| \leq |R_0|$ and similarly $|R \cap Y| \leq |R_1|$. Hence the lemma follows. \square

To prove the second claim, we start off with a monochromatic partition

R_1, R_2, \dots, R_d of S_f , where $d = D(S_f)$. Since the R_i s cover all of Y , we have:

$$\begin{aligned}
|Y| &= \sum_{i=1}^d |Y \cap R_i| \\
|Y|^2 &= \left(\sum_{i=1}^d |Y \cap R_i| \right)^2 \\
&\leq d \cdot \sum_{i=1}^d |Y \cap R_i|^2 && \text{from Cauchy-Schwarz} \\
&\leq d \cdot \sum_{i=1}^d |R_i^0| |R_i^1| && \text{by lemma 1.10} \\
&= d \cdot |S_f| && \text{since } R_i\text{s cover the } S_f
\end{aligned}$$

□

It has been proved that the above theorem cannot yield anything better than an $\Omega(n^2)$ lower bound. The proof can be found in [KKN95].

Example 1.14 (\oplus_n). *A very natural function to use Krapchenko's method on would be the parity function because \oplus_n differs on inputs that are just one bit apart.*

Lemma 1.15. $L(\oplus_n) \geq n^2$

Proof. For each string $x \in \{0, 1\}^n$ with k 1s, there are n strings with either $k-1$ or $k+1$ number of 1s. Hence $|Y| = n \cdot 2^n / 2 = n \cdot 2^{n-1}$. And since half of the total number of strings evaluate to 0 and the other half evaluate to 1, we have $|S_f| = 2^{n-1} \cdot 2^{n-1} = 2^{2n-2}$. Hence by Krapchenko's theorem, we have:

$$L(\oplus_n) \geq \frac{(n \cdot 2^{n-1})^2}{2^{2n-2}} = n^2 \tag{1.5}$$

□

Chapter 2

Monotone Formula Size

2.1 Introduction

In this chapter, we look at a technique to lower bound $L^+(f)$ using rank arguments over a field \mathbb{F} , where f is a monotone boolean function. This was suggested by Razbarov in [Raz90]. The technique indeed gives a way to lower bound the monotone monochromatic partition number of a rectangle. Following the proof of the technique itself, we show a way to apply to this technique for a function over a graph.

2.2 Rank Lower Bound

Theorem 2.1. *Let f be a monotone boolean function. Let $I \subseteq f^{-1}(0)$ and $J \subseteq f^{-1}(1)$. Fix a field \mathbb{F} . Let \mathcal{M} be the set of all monotone monochromatic sub-rectangles of $I \times J$. Let A be an $|I| \times |J|$ matrix. Then,*

$$L^+(f) \geq \frac{\text{rk}(A)}{\max_{R \in \mathcal{M}} \text{rk}(A_R)} \quad (2.1)$$

Proof. Let \mathcal{R} be the smallest monotone monochromatic partition of $S_f = f^{-1}(0) \times f^{-1}(1)$. We know \mathcal{R} covers the whole of $I \times J$ since $I \times J$ is a sub-

rectangle of S_f . So $A = \sum_{R \in \mathcal{R}} A_R$. Also, by sub-additivity of rank, we have:

$$\begin{aligned} \text{rk}(A) &\leq \sum_{R \in \mathcal{R}} \text{rk}(A_R) \leq |\mathcal{R}| \cdot \max_{R \in \mathcal{R}} \text{rk}(A_R) \\ \implies |\mathcal{R}| &\geq \frac{\text{rk}(A)}{\max_{R \in \mathcal{R}} \text{rk}(A_R)} \end{aligned}$$

and we know by Rychkov's lemma:

$$|\mathcal{R}| = D^+(S_f) \leq L^+(f)$$

Hence:

$$L^+(f) \geq \frac{\text{rk}(A)}{\max_{R \in \mathcal{R}} \text{rk}(A_R)} \implies L^+(f) \geq \frac{\text{rk}(A)}{\max_{R \in \mathcal{M}} \text{rk}(A_R)}$$

□

To be able to use this theorem effectively, we should identify a high rank matrix A that has the property that the ranks of A_R , where R is monotone monochromatic should be small. In the following section we give an example for one particular function defined over a graph.

2.3 Lower bound for dense graphs of girth at least 5

Throughout this section, we fix the field $\mathbb{F} = \text{GF}(2)$. The following theorem was proved by Jukna in [Juk04].

Theorem 2.2. *Let $G = ([n], E)$ be a graph. Define f_G such that it evaluates to 0 if and only if the set of vertices that the input assigns*

to 1 is an independent set. Formally:

$$f(x_1, \dots, x_n) = \bigvee_{(i,j) \in E} x_i \wedge x_j \quad (2.2)$$

If $\text{girth}(G) \geq 5$, then

$$L^+(f) \geq \frac{|E|}{2}$$

Proof. We can view bit strings $x \in \{0, 1\}^n$ as subsets of $[n]$. We can look at vertices as singletons and edges as a set with two elements (the two endpoints). For a vertex x , define I_x to be the set of all its proper neighbours in G . For an edge $x = \{u, v\}$, define I_x as the set of all vertices other than u and v which have edges incident to either u or v . i.e,

$$I_x = \begin{cases} \{z \in V \setminus \{x\} \mid (z, x) \in E\} & \text{if } x \in V \\ \{z \in V \setminus \{u, v\} \mid (z, u) \in E \text{ or } (z, v) \in E\} & \text{if } x \in E \end{cases} \quad (2.3)$$

Note that the sets I_x are independent sets since the graph G does not have triangles or 4-cycles. So, an input that assigns a 1 to all the elements of one of the I_x s and 0s elsewhere has to be rejected by f . Among all the elements of $f^{-1}(0)$, we focus our attention on only such inputs. i.e, we consider the subset $I = \{I_x \mid x \in V \cup E\}$ of $f^{-1}(0)$. On the other hand, take any input y that assigns a 1 to the endpoints of just a single edge. It is easy to see that $f(y) = 1$. Hence, for each edge $e \in E$, the input string that assigns 1 to exactly the endpoints of e is a 1 instance. Hence $E \subseteq f^{-1}(1)$. Among all the inputs in $f^{-1}(1)$, we focus our attention on only the inputs that correspond to setting the endpoints of an edge to 1. i.e., we consider the subset $J = E$ of $f^{-1}(1)$. From now on, our focus is only on the sub-rectangle $I \times J$, where I and J are as defined above.

Let A be a $(|V| + |E|) \times |E|$ boolean matrix with a row for each vertex and edge, and a column for each edge. An input corresponding to a row

x is the input that assigns 1 to exactly I_x . And an input corresponding to a column e of A corresponds to an input that assigns 1 exactly to the endpoints of an edge $e \in E$. The entries of A are defined as follows:

$$A_{i,j} = \begin{cases} 1 & \text{if } i \cap j \neq \emptyset \\ 0 & \text{if } i \cap j = \emptyset \end{cases} \quad (2.4)$$

We claim the following:

1. A is full rank.
2. For any monotone monochromatic rectangle M of I , $\text{rk}(A_M) \leq 2$.

If we prove the above two claims, the theorem follows as a direct application of Theorem 2.1. In what follows, we prove the above two claims one by one:

Claim (1). A is full rank.

Proof. It is sufficient to show that any non-empty set of columns do not add up to all 0s over $\text{GF}(2)$. Take any subset of the columns $F \subseteq E$.

- Suppose any vertex v is contained in odd number of the selected columns. Then the columns add up to give a 1 at the row corresponding to v . And we are done.
- Suppose each vertex v is contained in even number of columns from F . Then, in the graph restricted to the edges from F , each of these vertices have an even degree. Take any edge (u, v) from this restricted graph. It intersects an odd number of edges because - It intersects itself, and intersects each of the edges incident to u and each of the edges incident on v . Since u and v are even degree, the total number of edges that (u, v) intersects is odd. Hence, the entry in the row corresponding to the edge (u, v) will add up to 1.

□

Claim (2). For any monotone monochromatic sub-rectangle $M = M^0 \times M^1$ of I , $\text{rk}(A_M) \leq 2$.

Proof. Since M is monotone monochromatic, there is a vertex $v \in V$ such that $\forall x \in M^0, v \notin I_x$ and $\forall y \in M^1, v \in y$. The rows of A_M could be of two types:

- Row x with $v \in x$. In this case, since $v \in y, \forall y \in M^1$, we have $x \cap y \neq \emptyset$. Hence $A_M = 1$ on the entire row x in the columns indexed by M^1 .
- Row x with $v \notin x$. In this case, v must be at-least a distance 2 away from x . We can prove that in this case we get the all 0s row as follows: Suppose $A_M[x, y] = 1$ for some $y \in M^1$. This means that $x \cap y \neq \emptyset$. Let $u = x \cap y$. Now since $v \notin x, u \neq v$. But $v \in y$. This gives $y = (u, v)$, which implies $v \in I_x$. This contradicts our assumption. Hence, in this case, the row x is all 0s.

□

A direct application of the rank lower bound argument (theorem 2.1) on the matrix A gives the lower bound of $|E|/2$. □

Chapter 3

Communication Complexity and Circuit Depth

3.1 Introduction

In this chapter and the next, we will look at ways to use communication complexity to give a lower bound on the circuit depth for a circuit computing a boolean function. The importance of circuit depth lies in the way circuits are related to Turing machine computation. A standard Turing machine computing a function f can be converted to a circuit whose depth is equal to the running time of the Turing machine. The only non-trivial lower bound for circuit depth in the non-monotone world is for the function \oplus_n . The first proof was by Furst, Saxe, and Sipser in [FSS84] proves that \oplus_n does not have a polynomial size circuit family with constant depth and unbounded fan-in. This was also proved at the same time, independently, using a completely different approach, by M.Ajtai in [Ajt83] This was improved to exponential by [Yao85]. There is a proof for the same by Smolensky in [Smo87]. The tightest lower bound is proved by Hastad in [Hås86]. In the monotone world, however, it is known that there exists a

function which does not have a log depth bounded fan-in circuit family, but has a \log^2 depth bounded fan-in circuit family computing it. We look at this result in the next chapter.

The two-party communication model that captures many issues of communication complexity was suggested by Yao in 1979. A connection to circuit complexity and, in particular, to circuit depth was proven by Karchmer and Wigderson in 1988. We look at both these models and give a notion of reduction, all in preparation to give lower bounds in the next chapter.

3.2 Games on Relations

Let $R \subseteq X \times Y \times Z$. The communication game on R is the following: There are two players - Alice and Bob, each having unbounded computational power. Alice is given $x \in X$, Bob is given $y \in Y$, and the goal is to find a $z \in Z$ such that $(x, y, z) \in R$. At the end of the game, both Alice and Bob should agree on a z such that $(x, y, z) \in R$. An input pair (x, y) to Alice and Bob is called a legal input pair if $\exists z \in Z$ such that $(x, y, z) \in R$. The communication complexity $CC(R, x, y)$ of an input pair (x, y) over a ternary relation R is the minimum number of bits required to be exchanged between Alice and Bob to agree on a $z \in Z$ such that $(x, y, z) \in R$. The communication complexity $CC(R)$ of a ternary relation R is the total number of bits required to be exchanged between Alice and Bob in order to successfully play the game on R as described above on any pair of legal inputs. More formally:

$$CC(R) = \min_{\text{protocol}} \max_{x \in X, y \in Y} CC(R, x, y) \quad (3.1)$$

It is assumed that Alice and Bob are always given a legal input pair.

We look at two special cases of this game in the following sections.

Reductions

We can define a notion of reduction between communication games played on relations. If $S \subseteq X \times Y \times Z$ and $T \subseteq X' \times Y' \times Z'$ are two relations, we say $S \leq_m T$ if the following can happen:

- Alice has $x \in X$, Bob has $y \in Y$. Goal is to find a $z \in Z$ such that $(x, y, z) \in S$.
- Alice converts x to $x' \in X'$, Bob converts y to $y' \in Y'$.
- Alice and Bob play the communication game on the relation T with Alice's input as x' and Bob's input as y' . They find a $z' \in Z'$ such that $(x', y', z') \in T$.
- Alice and Bob now communicate at most m bits between each other and arrive at a z such that $(x, y, z) \in S$.

If $m = 0$, we simply write $S \leq T$ rather than $S \leq_0 T$.

Observation 3.1. *If $S \leq_m T$, then $CC(S) \leq m + CC(T)$.*

3.3 Karchmer-Wigderson game

For any boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$, consider the relation $KW_f \subseteq f^{-1}(0) \times f^{-1}(1) \times [n]$ defined as follows: $(x, y, i) \in KW_f \iff x \in f^{-1}(0), y \in f^{-1}(1), x_i \neq y_i$. The Karchmer-Wigderson game is the communication game played on this relation. So, in this model, Alice is given an input $x \in f^{-1}(0)$, and Bob is given a $y \in f^{-1}(1)$. The goal is to find a position i where their inputs differ. In this section, by $CC(f)$, we always mean $CC(KW_f)$. Given a rectangle S , let $CC(S)$ denote the communication complexity of the Karchmer-Wigderson game when the inputs given to Alice and Bob are guaranteed to come from S .

Theorem 3.2. *For any boolean function f , $CC(f) = d(f)$.*

Proof. We need to show two directions:

1. Circuit to protocol: $d(f) \geq CC(f)$.
2. Protocol to circuit: $CC(f) \geq d(f)$.

We will show these one by one:

Lemma 3.3 (Circuit to protocol). *For any boolean function f , $d(f) \geq CC(f)$.*

Proof. Alice is given an $x \in f^{-1}(0)$ and Bob is given a $y \in f^{-1}(1)$. Both of them have the same minimal depth circuit C for f , whose depth is $d = d(f)$. Assume that C has all negations pushed to the leaves. This is a reasonable assumption since pushing negations to leaves does not affect the depth of the circuit. We need to come up with a communication protocol such that Alice and Bob find an i where $x_i \neq y_i$ within d bits of communication. Consider the following protocol:

- Both Alice and Bob look at the topmost gate in C . If it is an \wedge gate, then it is Alice's turn to speak. If the topmost gate is an \vee gate, then, it is Bob's turn.
- If it is Alice's turn, Alice sends the bit 0 if the left branch of the \wedge gate is computing a 0. Else she sends 1. Alice and Bob proceed downward along the branch that Alice indicated to look at a gate which has a depth of $d - 1$.
- If it is Bob's turn, Bob looks at which branch of the \vee gate is giving a 1 output, and sends 0 if it is the left branch. Else Bob sends 1. Alice and Bob proceed along the branch that Bob indicated to look at a gate that has depth $d - 1$.

- Alice and Bob continue doing the above steps till they hit a leaf.

The above protocol works by moving down the circuit using the branches which evaluate differently for Alice and Bob. And hence, at the end Alice and Bob end up in a leaf which is labelled by the same literal, but evaluates differently for Alice and Bob. We can also observe that for each bit sent, the depth of the remaining circuit reduces by 1. Hence the total number of bits exchanged is $= d$. \square

Lemma 3.4. [*Protocol to Circuit*] For any boolean function f , $CC(f) \geq d(f)$

Proof. A boolean function g is said to separate a rectangle S if the following two properties hold:

1. $\forall x \in S^0, g(x) = 0$.
2. $\forall y \in S^1, g(y) = 1$.

That is, g separates a rectangle S if $S \subseteq S_g$.

We show that a circuit of depth $CC(f)$ computing f exists by proving a more general claim

Claim. For any rectangle $S = S^0 \times S^1$, there exists a boolean function g such that g separates S and $CC(S) \geq d(g)$.

Assuming the above claim, we prove the lemma as follows: Look at the rectangle $S_f = f^{-1}(0) \times f^{-1}(1)$. By the above claim, we know that there is a function g separating S_f and $CC(S_f) \geq d(g)$. But since $f^{-1}(0) \cup f^{-1}(1) = \{0, 1\}^n$, it has to be the case that g is indeed the function f itself. Hence we get $d(f) \leq CC(S_f)$

Now we prove the above claim:

Proof. Let $c = CC(S)$.

Base case: $c = 0$, then the whole of S is monochromatic. i.e., there exists an i such that $\forall x \in S^0, \forall y \in S^1, x_i \neq y_i$. This also means $\exists b \in \{0, 1\}$, $\forall p \in S^0, p_i = b$ and $\forall p \in S^1, p_i = \bar{b}$. In other words, the color of S is i . So f evaluates to 1 on input z if $z_i = \bar{b}$ and evaluates to 0 if $z_i = b$. Hence the circuit for f would be either be z_i or \bar{z}_i depending on b , and hence has a depth of 0.

Induction step: Let the protocol use c bits. Suppose Alice sent the first bit, then there is a partition of the rows S^0 into S_1^0 and S_2^0 such that Alice sends bit 0 if her input $x \in S_1^0$, and sends 1 if her input $x \in S_2^0$. Also, $S_1^0 \times S^1$ and $S_2^0 \times S^1$ are disjoint and have a communication complexity of $c-1$. From induction hypothesis, we have functions f_1 and f_2 that separate $S_1^0 \times S^1$ and $S_2^0 \times S^1$ respectively and both f_1 and f_2 have the property that $d(f_1) \leq CC(S_1^0 \times S^1)$ and $d(f_2) \leq CC(S_2^0 \times S^1)$. Let C_1 be the minimum depth circuit for f_1 and C_2 be the minimum depth circuit for f_2 . It is easy to see that $CC(S_1^0 \times S^1) \leq c-1$ and $CC(S_2^0 \times S^1) \leq c-1$. We claim that the desired circuit for f is simply $C_1 \wedge C_2$. The correctness of this claim can be seen as follows: f_1 and f_2 output 1 when given an input from S^1 because they separate $S_1^0 \times S^1$ and $S_2^0 \times S^1$ respectively. So $\forall z \in S^1, f_1(z) = 1$ and $f_2(z) = 1$. On the other hand, $\forall z \in S_1^0, f_2(z)$ could be anything but $f_1(z)$ is surely 0 (because f_1 separates $S_1^0 \times S^1$). And $\forall z \in S_2^0$, although $f_1(z)$ might evaluate to a 1, we know for sure that $f_2(z) = 0$ because $f_2(z)$ separates $S_2^0 \times S^1$. So, $\forall z \in S^1$, we have $f_1(z) = f_2(z) = 1$, and $\forall z \in S_1^0 \cup S_2^0$, at least $f_1(z) = 0$ or $f_2(z) = 0$ or both. Hence the claim that $C_1 \wedge C_2$ is a correct circuit for f holds. In the other case, where Bob transmits the first bit, a similar argument would give us a circuit $C_1 \vee C_1$. □

This ends the proof of Lemma 3.4. □

This ends the proof of Theorem 3.2. □

We can define a slight variation of this game for the monotone setting: Let f be a monotone boolean function on n variables. We define the relation $KW_f^+ \subseteq KW_f$, defined as: $(x, y, i) \in KW_f^+ \iff x \in f^{(-1)}(0), y \in f^{(-1)}(1), (x_i = 0) \wedge (y_i = 1)$. Alice is given $x \in X$ and Bob is given $y \in Y$ and the goal here is to find an i where $x_i = 0$ and $y_i = 1$. There has to exist at least one such i since f is monotone.

Theorem 3.5. $CC(KW_f^+) = d^+(f)$

Proof. Exactly the same proof as for theorem 3.2. Here, we do not need to assume that all negations appear only in leaves because there are no negations. \square

Now, we look at a technique to lower bound the communication complexity of the Karchmer-Wigderson game on a boolean function.

Monochromatic Partition Number

Lemma 3.6. *For any boolean function f , $\log(D(S_f)) \leq CC(f)$*

Proof. By Theorem 3.2, we know that $CC(f) = d(f)$. And since the number of leaves of a minimal leaf-size circuit for f cannot exceed $2^{d(f)}$, we have $d(f) \geq \log(L(f))$. Also, from Rychkov's lemma, we have $L(f) \geq D(S_f)$, which means $\log(L(f)) \geq \log(D(S_f))$. Putting these all together in one line, we get:

$$CC(f) = d(f) \geq \log(L(f)) \geq \log(D(S_f)) \quad (3.2)$$

Hence the proof. \square

We now look at an alternate proof for the above lemma which tells us how to partition S_f into at most $2^{CC(f)}$ parts:

Proof. Let $c = CC(f)$. Both Alice and Bob start with S_f as the active rectangle. With each bit that Alice (Bob) sends, the rows (columns) of the

active rectangle get split into two disjoint parts, and the active rectangle gets reduced to a smaller one. So the communication itself can be viewed as a tree with the root labelled with the whole rectangle S_f . Its children are labelled by rectangles obtained after splitting the rows (columns) based on the bit sent by Alice (Bob). This communication tree's depth would be exactly c . Since after c bits, Alice and Bob are able to decide on an i such that $x_i \neq y_i$, the leaves of this tree must correspond to monochromatic sub-rectangles. Hence, given a protocol that uses c bits on the rectangle S_f , we can give a monochromatic partition of S_f into at most 2^c parts. \square

3.4 Yao's game

This is another special case of the communication game defined earlier on relations. Here the relation R is indeed a boolean function $f : \{0, 1\}^l \times \{0, 1\}^m \rightarrow \{0, 1\}$. Alice is given an $x \in \{0, 1\}^l$, Bob is given a $y \in \{0, 1\}^m$. The goal of the game is to compute $f(x, y)$. Another interpretation: We can look at a matrix $M_f = X \times Y$, such that $M_f[x, y] = f(x, y)$. Alice is given a row x and Bob is given a column y . The goal is to compute $M_f[x, y]$. In general, denote the ternary relation corresponding to a function f by R_f and denote by M_f , the matrix whose rows correspond to $x \in \{0, 1\}^l$ and columns $y \in \{0, 1\}^m$ with each entry being $f(x, y)$. This model was first introduced by Yao in [Yao79].

Observation 3.7. *If $R_f \leq KW_g$, then, $CC(R_f) \leq CC(KW_g)$. In other words, if the Yao game on a function f reduces to the Karchmer-Wigderson game on a function g , then $CC(R_f) \leq CC(KW_g)$. Hence, a lower bound to $CC(KW_g)$ can be obtained by giving a lower bound on $CC(R_f)$.*

Now, we introduce the notion of "value-monochromatic". Consider a

boolean function f , and its associated matrix M_f as defined earlier. A rectangle S is called a value-monochromatic rectangle if $\forall (x_1, y_1), (x_2, y_2) \in S$, we have $M_f[x_1, y_1] = M_f[x_2, y_2]$. Informally, this is saying that the matrix M_f has the same value at all coordinates within that rectangle. So intuitively, we would expect that after Alice and Bob play the Yao game on f , they end up in a value-monochromatic sub-rectangle. We formalize this in the next sub-section along with some techniques to lower bound the communication complexity of the Yao Game played on a function f .

Note that the notion of “value-monochromatic” is also called simply “monochromatic” in many sources. But we give the above definition of monochromaticity a different name so as to prevent any confusion that might arise when the two different notions are being used under a same proof.

Lower bound techniques

Fooling set

Suppose a communication protocol produces the same bits to be exchanged on two different instances (x, y) and (x', y') , then we can prove that the bits exchanged for the instances (x, y') and (x', y) are also the same as for (x, y) . So if the function, that the communication protocol being played on, behaves differently on (x, y') or (x', y) compared to (x, y) , then this would show that the protocol is wrong. With this intuition, we can define a fooling set more formally:

Definition 3.8. *Let $f : X \times Y \rightarrow \{0, 1\}$. A set $S \subseteq X \times Y$ is a fooling set for f if there is a $z \in \{0, 1\}$ such that*

- *For every $(x, y) \in S$, $f(x, y) = z$.*

- For every distinct pair $(x_1, y_1) \in S$ and $(x_2, y_2) \in S$, either $f(x_1, y_2) \neq z$ or $f(x_2, y_1) \neq z$.

Lemma 3.9. *If f has a fooling set of size m , then $CC(R_f) \geq \log(m)$.*

Proof. Suppose there exists a protocol \mathcal{P} for R_f which exchanges $c < \log(m)$ bits. Then, clearly, there are at most 2^c possible bit patterns that can be exchanged. Let S be the fooling set with $|S| = m$. Then, by the pigeon hole principle, there are at least two distinct pairs $(x_1, y_1) \in S$ and $(x_2, y_2) \in S$ such that the bit pattern exchanged by the protocol is the same. Consider the rectangle formed with (x_1, y_1) and (x_2, y_2) as the corner points. It is easy to see that the bit pattern exchanged will be the same for the four pairs that form this rectangle. i.e., the bit pattern exchanged by the protocol for (x_1, y_2) , (x_2, y_1) , (x_1, y_1) and (x_2, y_2) are exactly the same. Hence, the protocol \mathcal{P} gives the same answer for the inputs (x_1, y_1) , (x_2, y_2) , (x_1, y_2) , and (x_2, y_1) . But by the definition of fooling set: $\exists z \in \{0, 1\}$ such that $\forall (x, y) \in S$, $f(x, y) = z$, but either $f(x_1, y_2) \neq z$ or $f(x_2, y_1) \neq z$. Hence the protocol is concluding a wrong output in at least one of these pairs. \square

Example 3.10. *We can give a lower bound for the function $\text{DISJ} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ which is defined as $\text{DISJ}(x, y) = 1 \iff x \cap y = \emptyset$. We claim that following set is a fooling set: $S = \{(A, \bar{A}) \mid A \subseteq \{1, \dots, n\}\}$ and hence $CC(R_{\text{DISJ}}) \geq n$. The correctness of the claim is as follows: The first required condition for fooling set is clearly satisfied since $\forall (A, \bar{A}) \in S$, $\text{DISJ}(A, \bar{A}) = 1$. For the second condition, take $(A, \bar{A}) \in S$ and $(B, \bar{B}) \in S$ where $A \neq B$, then, either $\exists x \in A \setminus B$, or $\exists x \in B \setminus A$. In the first case, $\text{DISJ}(A, \bar{B}) = 0$ and in the second case, $\text{DISJ}(\bar{A}, B) = 0$. Hence S is indeed a fooling set. By definition of S , $|S| = 2^n$. Hence, $CC(\text{DISJ}) \geq n$.*

Value-monochromatic Partition Number

This method gives a lower bound on $CC(\mathcal{R}_f)$ for a boolean function f using the value-monochromatic partition number defined earlier. This can be seen as a generalization of the fooling set method.

Lemma 3.11. $CC(\mathcal{R}_f) \geq \log(D(M_f))$

Proof. Identical to the alternative proof of Lemma 3.6, using sub-rectangles of M_f instead of S_f . \square

Rank Lower Bound

Let f be a boolean function on $X \times Y$, the rank of M_f can be used to lower bound the value-monochromatic partition number $D(M_f)$.

Lemma 3.12. $D(M_f) \geq \text{rk}(M_f)$.

Proof. Let $A = M_f$. Recollect that A_R , where R is a rectangle, denotes the matrix A with all entries outside R set to 0. Let \mathcal{R} be a value-monochromatic partition of M_f .

Observation 3.13. $\forall R \in \mathcal{R}, \text{rk}(A_R) \leq 1$.

This is because R is value-monochromatic. So it is either all 0s or all 1s. Now, we can write $M_f = \sum_{R \in \mathcal{R}} A_R$ since \mathcal{R} covers the whole of A . By sub-additivity of rank, we have $\text{rk}(M_f) \leq \sum_{R \in \mathcal{R}} \text{rk}(A_R)$. From the above observation, we get $\sum_{R \in \mathcal{R}} \text{rk}(A_R) \leq |\mathcal{R}|$. And hence, $\text{rk}(M_f) \leq \sum_{R \in \mathcal{R}} \text{rk}(A_R) \leq |\mathcal{R}|$. Since this holds for any value-monochromatic partition, it follows that $\text{rk}(M_f) \leq D(M_f)$. \square

Corollary 3.14. *From Lemma 3.11 and Lemma 3.12, we get*

$$CC(\mathcal{R}_f) \geq \log(\text{rk}(M_f))$$

Discrepancy bound

This is a technique to lower bound the value-monochromatic partition number of a rectangle. Let $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$. Let R be the matrix obtained by replacing 0s with 1s and 1s with -1 s in the matrix M_f . The discrepancy of a sub rectangle $A \times B$ of $\{0, 1\}^n \times \{0, 1\}^n$ in R is defined as :

$$\frac{1}{2^{2n}} \left| \sum_{x \in A, y \in B} R_{x,y} \right| \quad (3.3)$$

Discrepancy $\text{Disc}(f)$ of a function f is the maximum discrepancy over all sub rectangles of $\{0, 1\}^n \times \{0, 1\}^n$ in M_f .

Lemma 3.15. $D(M_f) \geq \frac{1}{\text{Disc}(f)}$

Proof. Let $\chi = D(M_f)$. We can show that there exists a sub rectangle of $\{0, 1\}^n \times \{0, 1\}^n$ with a discrepancy at least χ as follows: Since M_f has 2^{2n} entries, and it can be partitioned into χ value-monochromatic parts, by pigeon hole principle, one of the parts of this partition has at least $2^{2n}/\chi$ entries. Let this part be P . Since each of these parts is value-monochromatic, the entries in the matrix M_f inside rectangle P are all 0s or all 1s. So entries in rectangle P in the matrix R (obtained by changing 0s to 1s and 1s to -1 s of M_f) will be all 1s or all -1 s. Hence it will have a discrepancy of $1/\chi$. $\text{Disc}(f)$ takes max over all sub rectangles, hence, $\text{Disc}(f) \geq \frac{1}{\chi}$. \square

3.5 Non-determinism

The notion of non-deterministic protocols is a natural one. Here, the protocol is allowed to make a guess, and then Alice and Bob exchange

information to verify the guess. If, in any one of the non-deterministic branches, it is found that the answer is 1, the protocol returns 1. The non-deterministic complexity of a function f is the total number of bits exchanged plus the size of the guesses made by the best protocol which is allowed to make guesses for the Yao game on f . From now on, we denote non-deterministic complexity of a function f by $N^1(R_f)$. For eg: Consider the non-equality function $NEQ : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, which is defined as $NEQ(x, y) = 1 \iff x \neq y$. The communication game corresponding to R_{NEQ} would be: Alice and Bob are both given an n bit number. The goal is to determine if $x \neq y$. A non deterministic protocol for this is: A guess $0 < i \leq n$ is made. Alice sends bob the bit at the i 'th position of her input. If the bit is different from bit at the i 'th position of Bob's input, then, Bob declares that $NEQ(x, y) = 1$. The guess made was of size $\log(n)$ and one more bit was exchanged. Hence $N^1(R_{NEQ}) = O(\log(n))$. Note that if the input pair (x, y) was a 0 instance of a function f , then at the end of a non-deterministic protocol for f , it is not necessary for Alice and Bob to know exactly which value-monochromatic rectangle of M_f the input pair belonged to. Just the fact that (x, y) does not belong to a rectangle colored 1 suffices.

Non-deterministic communication can be seen as a communication tree whose leaves are value-monochromatic sub-rectangles, which together, form a cover of the 1s of the matrix M_f . By this, we mean that the value-monochromatic sub-rectangles are allowed to overlap, but together they have to cover all the 1s. Let $C^1(R_f)$ denote a best possible 1 cover of R_f . This is a relaxation of the condition that the value-monochromatic sub-rectangles need to be disjoint. So intuitively, non-deterministic communication complexity for a relation R_f , where f is a boolean function, should be lesser than deterministic communication complexity.

On similar lines we can define the notion of co-nondeterministic communication complexity of a function f as the number of bits exchanged plus the number of bits guessed by a best communication protocol which has the leaves of its communication tree as value-monochromatic sub-rectangles that cover all the 0s of the rectangle R_f . We denote the co-nondeterministic communication complexity by $N^0(R_f)$.

Further, we can look at protocols in which Alice and Bob determine exactly which monochromatic rectangle the pair of inputs is in. Such a protocol will have leaves that correspond to monochromatic sub-rectangles that cover the whole of R_f . The number of bits used by the best possible such protocol for a function f will be denoted by $N(R_f)$.

Observation 3.16. *For $b \in \{0, 1\}$, $N^b(R_f) = O(\log(C^b(f)))$.*

The above observation can be seen as follows: Let, without loss of generality, $b = 0$, and let the rectangles of the 0-cover be labelled such that each rectangle gets a unique number. Alice has x , Bob has y , Consider a protocol that guesses the label of the rectangle where (x, y) lies. Alice sends 1 if and only if her row x indeed lies in the rectangle that was guessed. Bob sends 1 if and only if his column y is contained in the rectangle that was guessed. If both of them sent a 1, then the branch corresponding to this guess returns a 1. Similarly for $b = 1$. Hence a total of $\log(C^b(f)) + 2$ bits were used by the protocol in guessing and exchanges. On similar lines, it can be seen that $N(R_f) = O(\log(C(f)))$.

Say that a rectangle $R = A \times B$ intersects another rectangle $S = A' \times B'$ in rows if $A \cap A' \neq \emptyset$. Similarly, say that R intersects S in columns if $B \cap B' \neq \emptyset$. We have the following observation:

Observation 3.17. *In any value-monochromatic cover, a 1-rectangle intersects at most half of the 0-rectangles in rows, or intersects at most half of the 0-rectangles in columns.*

The above observation follows because if a 1-rectangle R intersects more than half of the 0-rectangles in rows and also intersects more than half the 0-rectangles in columns, then it means that there is at least one 0-rectangle that intersects R in both rows and columns, which means that R overlaps with a 0-rectangle. This is a contradiction to fact that R is value-monochromatic.

The following theorem connecting $CC(R_f), N^0(R_f)$ and $N^1(R_f)$ is proved in [KN97]. The proof is as follows:

Theorem 3.18. $CC(R_f) = O(N^0(R_f) \times N^1(R_f))$.

Proof. The general idea of the proof is: Alice and Bob initially start off with all 0-rectangles alive. With each phase of the protocol, they halve the number of 0-rectangles alive. The overall goal is to search for a 0-rectangle which contains (x, y) . If they cannot find such a rectangle, they conclude that (x, y) must be contained in a 1-rectangle. The protocol is such that number of bits exchanged in each phase is at most $\log(C^1(f))$, and there are at most $\log(C^0(f))$ phases. The protocol in more detail:

1. Alice looks at the 0-rectangles that are alive. If there are none, she announces that $f(x, y) = 1$. Otherwise, she searches for a 1-rectangle Q that contains the row x and intersects in rows with at most half of the 0-rectangles that are alive, and sends the unique label of Q (this ends the phase and Alice and Bob update the set of alive 0-rectangles to all those 0-rectangles that intersect with Q). If she cannot find such a rectangle, then she tells Bob that such a rectangle does not exist.
2. Bob searches for a 1-rectangle R which contains the column y and intersects in columns with at most half of the alive 0-rectangles. If such a rectangle exists, then Bob sends its unique label to Alice (this

ends the phase and Alice and Bob update the set of alive rectangles as before). Else, Bob announces that $f(x, y) = 0$.

Each phase uses $\log(C^1(f)) + O(1)$ bits. And there are at most $\log(C^0(f))$ number of phases. To prove that the protocol works, we note that if (x, y) is in a 0-rectangle, then that 0-rectangle remains alive throughout. Hence, if there is no 0-rectangle alive, then it is correct to conclude that $f(x, y) = 1$. On the other hand, if (x, y) belongs to a 1-rectangle, then by a previous observation, either this 1-rectangle intersects in rows with half of the 0-rectangles, or intersects in columns with half of the 0-rectangles. Hence, if both Alice and Bob cannot find such a rectangle, then it is correct to conclude that $f(x, y) = 0$. Hence the result follows. \square

Observation 3.19. *The above theorem implies that it cannot happen that both $N^0(R_f)$ and $N^1(R_f)$ are exponentially smaller than $CC(R_f)$. Also, we get $CC(R_f) = O((N(f))^2)$ since $N^1(f) = O(N(f))$ and $N^0(f) = O(N(f))$.*

Chapter 4

Lower Bounds using Communication Complexity

4.1 Introduction

The lower bounds for circuit depth achieved so far for circuits in the non-monotone world are not that significant. For various such results, see [Weg87], [Dun88], [Hås86], [BS90]. In this chapter we will study three non-trivial lower bounds for circuit depth in the monotone world. The first result we look at uses direct summing to construct a function in stages while maintaining a lower bound to circuit depth at each stage. The second result is an even stronger one, where we look at s-t connectivity. It uses a reduction from a relation that appears to be tailor made for the purpose. And the last result is a more general approach to use a function with a special property to construct another function which has a non-trivial lower bound on circuit depth.

4.2 Direct Sum

In this section we look at an $\Omega(\log(n) \cdot \log \log(n))$ depth lower bound for monotone circuits proved in [KRW91] using direct summing as a way to boost the required communication complexity. Some definitions required for what follows: Let $\mathcal{P}(n)$ denote the set of all subsets of the set $[n]$. And let $\mathcal{P}_k(n)$ denote the set of all subsets S of $[n]$ such that $|S| = k$.

Definition 4.1. $\text{DISJ}_{n,k}$ denotes the disjointness function. It is defined as $\text{DISJ}_{n,k} : \mathcal{P}_k(n) \times \mathcal{P}_k(n) \rightarrow \{0, 1\}$ where $\text{DISJ}_{n,k}(S, T) = 1 \iff S \cap T = \emptyset$.

We will look at $\text{DISJ}_{n,k}$ as a ternary relation.

Composition of two boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and $g : \{0, 1\}^m \rightarrow \{0, 1\}$ is defined as follows: $f \circ g : \{0, 1\}^{nm} \rightarrow \{0, 1\}$

$$f \circ g(x) = f(g(x_1, x_2, \dots, x_m), g(x_{m+1}, \dots, x_{2m}), \dots, g(x_{m(n+1)}, \dots, x_{nm}))$$

So function composition can be seen as a way to combine two boolean functions in a tree-like fashion. In the above case, the g 's form the leaves of the tree, and f is the root.

The Direct Sum of two relations $R \subseteq X \times Y \times Z$ and $S \subseteq X' \times Y' \times Z'$ is defined as follows: $R \otimes S \subseteq (X \times X') \times (Y \times Y') \times (Z \times Z')$. $((x, x'), (y, y'), (z, z')) \in R \otimes S \iff (x, y, z) \in R$ and $(x', y', z') \in S$. Intuitively, solving the communication game on the direct sum of R and S is like solving each of them simultaneously. The main theorem is the following:

Theorem 4.2. *For infinitely many n , there exists a monotone boolean function f on n variables such that*

$$\Omega(\log(n) \log \log(n)) \leq d^+(f) = \text{CC}(\text{KW}_f^+)$$

To prove this, we prove a more general theorem:

Theorem 4.3. $\forall k, m$, such that $k < m$, let $l = \lfloor 2k4^k \ln(m) \rfloor$. \exists a monotone boolean function g on $m+l$ variables such that for any $y \in \mathbb{Z}$, the function $f = g^{(y)}$ satisfies the following equation:

$$\Omega(y \cdot k \cdot \log(m)) \leq \text{CC}(\text{DISJ}_{m,k}^{(y)}) \leq \text{CC}(\text{KW}_g^{+(y)}) = \text{CC}(\text{KW}_f^+) \quad (4.1)$$

Assuming the above theorem, we prove Theorem 4.2 as follows:

Proof. Choose $m = \log(n)$ and $k = \frac{1}{4} \log(m)$. Since g is on $m + \lfloor 2k4^k \ln(m) \rfloor$ variables, this setting of k and m gives a g that works on $O(m)$ variables. We want f as a function on $O(n)$ variables. i.e., we want $(m+l)^y = O(n)$. Since $(m+l) = O(m)$, choose $y = \frac{\log(n)}{\log \log(n)}$. By plugging in these values, equation 4.1 now gives:

$$\Omega\left(\frac{\log(n)}{\log \log(n)} \cdot \frac{1}{4} \log(\log(n)) \cdot \log \log(n)\right) \leq \text{CC}(\text{KW}_f^+)$$

□

Now we set about proving Theorem 4.3.

Sketch of Proof:

1. Find a monotone function g on $(m+l)$ variables that has the property that for k such that $l = \lfloor 2k4^k \ln(m) \rfloor$, $\text{DISJ}_{m,k} \leq \text{KW}_g^+$. Hence $\forall y$, $\text{DISJ}_{m,k}^{(y)} \leq \text{KW}_g^{+(y)}$
2. Compose g with itself y times to get a monotone function f on $(m+l)^y$ variables.
3. Prove that $\text{CC}(\text{KW}_g^{+(y)}) \leq \text{CC}(\text{KW}_f^+)$.

4. From 3 and the fact that $\text{DISJ}_{m,k}^{(y)} \leq \text{KW}_g^{+(y)}$, we get:

$$\text{CC}(\text{DISJ}_{m,k}^{(y)}) \leq \text{CC}(\text{KW}_g^{+(y)}) \leq \text{CC}(\text{KW}_{g(y)}^+) = \text{CC}(\text{KW}_f^+)$$

5. Prove that $\text{CC}(\text{DISJ}_{m,k}^{(y)}) = \Omega(y \cdot k \cdot (\log(m) - \log(k)))$.

□

In what follows, we prove the claims 3, 5, 1 in that order.

Claim (3). For any two monotone boolean functions $f : \{0, 1\}^m \rightarrow \{0, 1\}$ and $g : \{0, 1\}^n \rightarrow \{0, 1\}$,

$$\text{KW}_f^+ \otimes \text{KW}_g^+ \leq \text{KW}_{f \circ g}^+.$$

Proof. The game on the left side is: Alice's input: $x \in f^{-1}(0)$, and an $x' \in g^{-1}(0)$. Bob's input: $y \in f^{-1}(1)$, and an $y' \in g^{-1}(1)$. Goal is to find i, j such that $x_i = 0, y_i = 1$ and $x'_j = 0, y'_j = 1$. Note that $|x| = |y| = m$ and $|x'| = |y'| = n$.

Alice constructs $u = u^1, u^2, \dots, u^n$; $|u| = mn$ where

$$u^i = \begin{cases} x' & \text{if } x_i = 0. \\ 1^m & \text{if } x_i = 1. \end{cases}$$

Then we have: $\forall i, g(u^i) = x_i$. And so $f \circ g(u) = f(x) = 0$.

Bob constructs $v = v^1, v^2, \dots, v^n$; $|v| = mn$, where

$$v^i = \begin{cases} y' & \text{if } y_i = 1. \\ 0^m & \text{if } y_i = 0. \end{cases}$$

Then, we have: $\forall i, g(v^i) = y_i$ and so, $f \circ g(v) = f(y) = 1$.

Alice and Bob play the communication game on the relation $\text{KW}_{f \circ g}^+$ with

inputs u, v . At the end of the game, they both agree on an index k such that $u_k = 0$ and $v_k = 1$. They compute $i = \lceil \frac{k}{m} \rceil$ and $j = (k-1) \bmod(m) + 1$. We claim that i and j as computed are a correct answer to the communication game played on the relation KW_f^+ and KW_g^+ respectively. The correctness can be seen as follows: We know $u_k = 0, v_k = 1$. Since $u_k = 0$, u_k cannot be in a block u^i where $x_i = 1$. And since $v_k = 1$, v_k cannot be in a block v^i where $y_i = 0$. So $x_i = 0$, and $y_i = 1$. Hence i is a correct answer to the game KW_f^+ . On the other hand, since $x_i = 0$, $u^i = x'$. So $u_k = j^{\text{th}}$ bit of $u^i = x'_j$. Similarly, since $y_i = 1$, $v^i = y'$. So $v_k = j^{\text{th}}$ bit of $v^i = y'_j$. And $x'_j = 0$ and $y'_j = 1$. And hence j is a valid answer to the game KW_g^+ . This ends the proof for Claim(3). \square

Claim (5). $CC(\text{DISJ}_{m,k}^{(y)}) = \Omega(k \cdot y \cdot (\log(m) - \log(k)))$

Proof. The proof is in two stages:

(a) Let R and R' be the ternary relations associated with a pair of boolean functions, then,

$$CC(R \otimes R') \geq \log(\text{rk}(M_R)) + \log(\text{rk}(M_{R'})).$$

(b) For any $m, k < m$, $\text{rk}(\text{DISJ}_{m,k}) = \Omega(k(\log(m) - \log(k)))$.

Proving the first part gives $CC(\text{DISJ}_{m,k}^{(y)}) \geq y \cdot \log(\text{rk}(\text{DISJ}_{m,k}))$. And with the second part, we get $CC(\text{DISJ}_{m,k}^{(y)}) = \Omega(y \cdot k(\log(m) - \log(k)))$. We prove the two parts one by one:

Proof of (a). Let $R : X \times Y \rightarrow \{0, 1\}$ and $R' : X' \times Y' \rightarrow \{0, 1\}$. Define the function $R \cdot R' : X \times X' \times Y \times Y' \rightarrow \{0, 1\}$ such that $R \cdot R'(x, y, x', y') = R(x, y) \wedge R'(x', y')$.

Observation 4.4. $M_{R \cdot R'} = M_R \otimes M_{R'}$ where \otimes denotes tensor product of the matrices.

Now, $R \cdot R' \leq R \otimes R'$ in a natural way. Hence $CC(R \otimes R') \geq CC(R \cdot R')$. Using corollary 3.14, we get $CC(R \otimes R') \geq \log(\text{rk}(M_{R,R'}))$. Using the above observation, we have: $CC(R \otimes R') \geq \log(\text{rk}(M_{R,R'})) \geq \log(\text{rk}(M_R \otimes M_{R'}))$. Using the well known fact that rank of a matrix is multiplicative with respect to tensoring, we get: $\text{rk}(M_R)\text{rk}(M_{R'}) \leq \text{rk}(M_R \otimes M_{R'})$. Using this fact, we now get: $\log(\text{rk}(M_R \otimes M_{R'})) \geq \log(\text{rk}(M_R)) + \log(\text{rk}(M_{R'}))$. Hence $CC(R \otimes R') \geq \log(\text{rk}(M_R)) + \log(\text{rk}(M_{R'}))$. \square

Proof of (b).

Fact 4.5. *It is known that the disjointness matrix has full rank over \mathbb{R} . For a full proof, see [Got66]*

From the above fact: $\text{rk}(\text{DISJ}_{m,k}) = \binom{m}{k} \geq \left(\frac{m}{k}\right)^k$. \square

This ends the proof of Claim(5). \square

Now we need to prove that a function with the properties in step 1 exists. This was proved in [Raz90].

Theorem 4.6. *For every m, k , and $l = \lfloor 2k4^k \ln(m) \rfloor$, \exists monotone function $g : \{0, 1\}^m \times \{0, 1\}^l \rightarrow \{0, 1\}$ such that $\text{DISJ}_{m,k} \leq \text{KW}_g^+$.*

Proof. Consider a function MINCOVER, whose input is a bipartite graph $G = (\mathbb{P}, \mathbb{Q}, E)$ and an integer k . MINCOVER evaluates to 1 if and only if there exists a $\mathbb{P}_0 \subseteq \mathbb{P}, |\mathbb{P}_0| = k$ such that $\forall q \in \mathbb{Q}, \exists P \in \mathbb{P}_0, (P, q) \in E$. This function can also be seen as a set-cover. i.e., \mathbb{Q} can be viewed as the universe and \mathbb{P} can be viewed as a collection of subsets of \mathbb{Q} . The function returns 1 if and only if there is a k sized sub-collection of \mathbb{P} that can cover the whole of \mathbb{Q} .

Fix $\mathbb{P}, \mathbb{Q}, k$, we can define a monotone function called SETCOVER that works on $|\mathbb{P}|+|\mathbb{Q}|$ variables based on MINCOVER as follows: $\text{SETCOVER}(\mathbb{P}' \subseteq \mathbb{P}, \mathbb{Q}' \subseteq \mathbb{Q})$, evaluates to $\text{MINCOVER}(\mathbb{P}', \mathbb{Q} \setminus \mathbb{Q}', E \cap (\mathbb{P}' \times \mathbb{Q} \setminus \mathbb{Q}'))$. That

is, the function SETCOVER is a restriction of MINCOVER to the sub-collection \mathbb{P}' and $Q \setminus Q'$. Note that we restrict SETCOVER to $Q \setminus Q'$ rather than Q' , because otherwise, the function would not be monotone. SETCOVER corresponds to the g in the theorem.

To prove the theorem, we will first need to prove a lemma about covering the matrix $\text{DISJ}_{m,k}$. Let the rows of the matrix $\text{DISJ}_{m,k}$ be U and columns V . Define rectangle $R_i^0 = \{u \in U \mid i \in u\} \times \{v \in V \mid i \in v\}$ for each $1 \leq i \leq m$. Intuitively, R_i^0 is the rectangle contains all pairs that intersect at i , and hence the matrix $\text{DISJ}_{m,k}$ would have an entry of 0 at each position that belongs to R_i^0 .

Observation 4.7. *The R_i^0 s cover all the 0 entries of the matrix $\text{DISJ}_{m,k}$*

Given an $\varepsilon \in \{0, 1\}^n$, define the rectangle $R_\varepsilon^1 = \{u \in U \mid \forall i \in u, \varepsilon_i = 1\} \times \{v \in V \mid \forall i \in v, \varepsilon_i = 0\}$. Intuitively R_ε^1 covers all the pairs (u, v) such that u is a subset of ε and v does not intersect with ε . Clearly all entries in $\text{DISJ}_{m,k}$ with coordinates in R_ε^1 are 1. Now we prove a lemma about the number of ε 's required to cover all the 1's using R_ε^1 rectangles.

Lemma 4.8. *For $l = \lfloor 2k4^k \ln(m) \rfloor$, $\exists \varepsilon_1, \varepsilon_2, \dots, \varepsilon_l$ such that $\bigcup_{i=1}^l R_{\varepsilon_i}^1$ covers all the 1s in $\text{DISJ}_{m,k}$.*

Proof. We prove this by a probabilistic argument. Note that for a fixed u, v such that $u \cap v = \emptyset$ and a random ε , $\Pr[(u, v) \in R_\varepsilon^1] = \frac{1}{2^{|u|+|v|}}$. The probabilistic argument is as follows: Pick independently at random

$\varepsilon_1, \varepsilon_2, \dots, \varepsilon_l \in \{0, 1\}^m$. Then, we have the following:

$$\begin{aligned}
\Pr[\exists(u, v) \in \mathcal{U} \times \mathcal{V}, u \cap v &= \emptyset \text{ and } (u, v) \text{ not covered by any } R_{\varepsilon_i}^1] \\
&\leq |\mathcal{U}| \cdot |\mathcal{V}| \cdot \max_{u \cap v = \emptyset} \Pr \left[(u, v) \notin \bigcup_{i=1}^l R_{\varepsilon_i}^1 \right] \\
&< m^{2k} \left(1 - \frac{1}{2^{|u|+|v|}} \right)^l \\
&\leq 1 \quad \text{for our choice of } l.
\end{aligned}$$

□

Now, we give a reduction from $\text{DISJ}_{m,k}$ to $\text{KW}_{\text{SETCOVER}}^+$.

Lemma 4.9. $\text{DISJ}_{m,k} \leq \text{KW}_{\text{SETCOVER}}^+$. Here SETCOVER is on $m + l$ variables, where $l = \lfloor 2k4^k \ln(m) \rfloor$.

Proof. The game on the left side is: Alice and Bob are each given as input k sized subsets. The goal is to determine if their subsets intersect. Assume that both Alice and Bob have agreed upon same set of $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_l$ from lemma 4.8 before the game begins (this can be done since the ε_i s do not depend on the inputs). Fix $Q = \{q_1, q_2, \dots, q_l\}$, where l is from lemma 4.8, and $\mathbb{P} = \{P_1, P_2, \dots, P_m\}$, where for each $1 \leq i \leq m$, $P_i = \{q_j \in Q \mid i \in \varepsilon_j\}$. Elements of Q correspond to ε_i s. Alice has to convert her subset to a 0 instance (\mathbb{P}_0, Q_0) of SETCOVER, Bob has to convert his subset to a 1 instance (\mathbb{P}_1, Q_1) of SETCOVER such that the answer to the communication game on $\text{KW}_{\text{SETCOVER}}^+$ gives a way for them to determine whether or not their input subsets intersect.

We first describe Bob's reduction: Let input to Bob be the set Y . Bob converts this to (\mathbb{P}_1, Q_1) where $\mathbb{P}_1 = \{P_i \mid i \in Y\}$ and $Q_1 = \{q_i \mid \varepsilon_i \cap Y = \emptyset\}$. Intuitively, Bob restricts \mathbb{P} to the elements corresponding to Y and chooses those elements to remove from the universe Q that cannot be covered by

Y. Hence, by definition, (\mathbb{P}_1, Q_1) is a 1 instance.

Let input to Alice be X. Alice restricts \mathbb{P} to elements corresponding to \bar{X} , and Q to elements which can be covered by every set P_i , $i \in X$. Formally: $\mathbb{P}_0 = \{P_i | i \notin X\}$ and $Q_0 = \{q_i \in Q | \exists j \in X, \varepsilon_i \cap P_j = \emptyset\}$.

Claim. (\mathbb{P}_0, Q_0) is a 0 instance of SETCOVER

Proof.

$$\begin{aligned}
X \cap \bar{X} = \emptyset &\implies \text{DISJ}[X, \bar{X}] = 1 \\
&\implies \exists \varepsilon \in \{\varepsilon_1, \dots, \varepsilon_l\}, R_\varepsilon^1 \text{ covers } \text{DISJ}[X, \bar{X}] \\
&\implies \exists \varepsilon, X \subseteq \varepsilon \text{ and } \varepsilon \cap \bar{X} = \emptyset \\
&\implies \exists \varepsilon, \forall i \in X, P_i \cap \varepsilon \neq \emptyset \text{ and } \forall i \in \bar{X}, P_i \cap \varepsilon = \emptyset
\end{aligned}$$

This means that there is at least one ε for which, in Alice's instance, none of the P_i s chosen by Alice can cover it. Hence it is a 0 instance. \square

After playing the communication game on KW_{SETCOVER}^+ with the modified inputs, with no extra communication, Alice and Bob can determine if $X \cap Y = \emptyset$. Suppose the answer to KW_{SETCOVER}^+ was p (position where Alice's instance had 0 and Bob's instance had a 1).

Claim. If $p \leq m$, then $X \cap Y \neq \emptyset$.

Proof.

$$\begin{aligned}
p \leq m &\implies p \in Y \text{ and } p \notin \bar{X} \\
&\implies p \in Y \text{ and } p \in X \\
&\implies p \in X \cap Y \\
&\implies X \cap Y \neq \emptyset
\end{aligned}$$

\square

Claim. If $p > m$, then, $X \cap Y = \emptyset$

Proof. Let $j = p - m$. Then, we have:

$$\begin{aligned}
 p > m &\implies \forall i \in X, P_i \cap Q_j \neq \emptyset && \text{since } p^{\text{th}} \text{ bit is 0 for Alice} \\
 &\text{and } \forall i \in Y, Q_j \cap P_i = \emptyset && \text{since } p^{\text{th}} \text{ bit is a 1 for Bob} \\
 &\implies X \cap Y = \emptyset
 \end{aligned}$$

□

This ends the proof of Lemma 4.7.

□

This ends the proof of the Theorem 4.3.

□

4.3 FORK Game

In this section we look at an $\log^2(n)$ depth lower bound for directed $s-t$ connectivity which was proved in [KW88]. The $s-t$ connectivity function STCON_n is defined as follows: Given a directed graph on n nodes, a source vertex s and a target vertex t ,

$\text{STCON}(G, s, t) = 1 \iff$ there is a directed path starting at s and ending in t

We assume without loss of generality that the vertices are numbered from 1 to n and $s = 1$ and $t = n$. KW game on this function consists of Alice being given a graph G_0 that does not have an $s-t$ path while Bob has a graph G_1 that does have an $s-t$ path. The goal is to find an edge (u, v) which is not present in G_0 , but is present in G_1 .

Define a relation $\text{FORK}_{w,l} : \Sigma^l \times \Sigma^l \times [l]$ where $(x, y, i) \in \text{FORK} \iff (x_i = y_i \text{ and } x_{i+1} \neq y_{i+1})$. Here Σ is a w sized alphabet. Assume the symbols in Σ are $\Sigma[1], \Sigma[2], \dots, \Sigma[w]$. Assume x and y have a 0 and $l+1$

position such that $x_0 = y_0 = \Sigma[1]$ and $x_{l+1} = \Sigma[w - 1]$ and $y_{l+1} = \Sigma[w]$. With these assumptions, for every $x, y \in \Sigma^l$, $\exists i \in \{0\} \cup [l]$ such that $(x, y, i) \in \text{FORK}$.

Theorem 4.10. $d^+(\text{STCON}_n) = \text{CC}(\text{KW}_{\text{STCON}_n}^+) \geq \Omega(\log^2(n))$

Sketch of Proof: The proof sketch is as follows:

1. Show that $\text{FORK}_{\sqrt{n}, \sqrt{n}-2} \leq \text{KW}_{\text{STCON}_n}^+$.
2. Show that $\text{CC}(\text{FORK}_{w,l}) = \Omega(\log(l) \cdot \log(w))$.
3. Hence conclude that $\text{CC}(\text{KW}_{\text{STCON}_n}^+) \geq \Omega(\log^2(n))$.

□

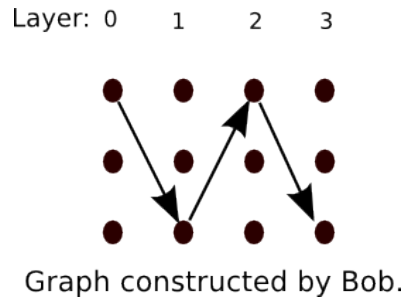
We now prove (1) and (2) below:

Proof of 1. We will restrict the domain of STCON_n to layered graphs with $l+2$ layers with each layer having w vertices such that $l+2 = w = \sqrt{n}$. Now we can show that $\text{FORK}_{n,n}$ reduces to the KW relation of this restricted version, and hence reduces to the KW game of the more general STCON_n . Recall that $\text{KW}_{\text{STCON}_n}^n$ is the communication game where Alice gets a graph with no $s - t$ path and Bob gets a graph which has a directed $s - t$ path.

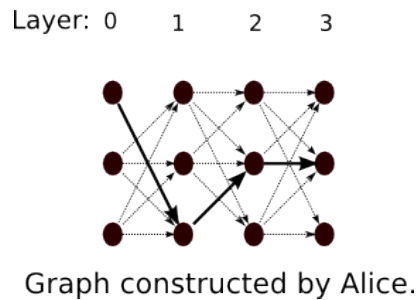
Alice is given an $x \in \{0, 1\}^l$. Alice constructs the graph G_0 with $l+2$ layers - corresponding to the positions x_0, x_1, \dots, x_{l+1} . Each layer has w vertices - corresponding to the w letters of the alphabet. Let $v_{i,j}$ denote vertex j of layer i . Alice constructs the path P_x corresponding to x_1, x_2, \dots, x_l . i.e., For $0 \leq i \leq l$, $(v_{i,x_i}, v_{i+1,x_{i+1}}) \in E$. Alice also adds an edge between a vertex of layer i that is not in the path P to every other vertex of layer $i+1$. G_1 is clearly a 0 instance of STCON since $t = v_{l+1,w}$ and $x_{l+1} = \Sigma[w - 1]$ by assumption.

Bob is given a $y \in \{0, 1\}^l$. Bob constructs the graph G_1 with the same number of layers and vertices as above. But G_1 consists of only the path P_y corresponding to y in G . i.e., For $0 \leq i \leq l$, $(v_{i,y_i}, v_{i+1,y_{i+1}}) \in E$. This is clearly a 1 instance.

For example, let $\Sigma = \{1, 2, 3\}$. Let the input to Alice and Bob be “32” and “31” respectively. The strings after extending with a 0th and 3rd coordinate are “1322” and “1313” respectively. Bob constructs the graph shown in the following figure:



And Alice constructs the following graph:



Now, by playing the communication game on the two constructed graphs, Alice and Bob can determine the answer to $\text{FORK}(x, y)$. This is as follows: STCON on G_0 and G_1 gives an edge (u, v) that is not in G_0 but is in G_1 . u belongs to some layer i . u belongs to path P_x because otherwise, (u, v)

would have been an edge in G_0 by construction. u is in the path P_y also because otherwise, (u, v) would not be an edge in G_1 . But now, $u \in P_x$, but $(u, v) \notin G_0$. This can only be possible if $v \notin P_x$. Hence i is a correct answer to $\text{FORK}(x, y)$. This ends the proof of (1). \square

Proof of 2. We first define the notion of an (α, l) protocol. An (α, l) protocol \mathcal{P} for $\text{FORK}_{w,l}$ is a protocol with a good set S , $|S| \geq \alpha \cdot \{0, 1\}^l$ such that $\forall x \in S, y \in S, \mathcal{P}(x, y) = \text{FORK}(x, y)$. Let $C(\alpha, l)$ denote the minimum number of bits required for an (α, l) communication protocol for FORK . We use the following two lemmas that we will prove later:

Lemma 4.11 (a). *If there is a c -bit (α, l) protocol for FORK , there is also a $c - 1$ bit $(\alpha/2, l)$ protocol for FORK . i.e.,*

$$C\left(\frac{\alpha}{2}, l\right) \leq C(\alpha, l) - 1 \leq C(\alpha, l)$$

Lemma 4.12 ((b) Amplification). *If there exists a c bit (α, l) protocol for FORK , then there is also a c -bit $(\sqrt{\alpha}/2, l/2)$ protocol for it. i.e.,*

$$C\left(\frac{\sqrt{\alpha}}{2}, \frac{l}{2}\right) \leq C(\alpha, l)$$

Assuming the above stated lemmas, we show how to prove the theorem: Since $C(1, l) \geq C(1/w^{1/3}, l)$ its sufficient to prove that $C(1/w^{1/3}, l) = \Omega(\log(l) \cdot \log(w))$. Apply the two lemmas to prove the theorem as follows:

- Start with a $(1/w^{1/3}, l)$ protocol.
- Apply lemma (a) $\Theta(\log(w))$ times to get $C(1/w^{1/3}, l) \geq \Omega(\log(w)) + C(4/w^{2/3}, l)$.
- Apply lemma (b) once to get $C(4/w^{2/3}, l) \geq C(1/w^{1/3}, l/2)$.
- Now we have $C(1/w^{1/3}, l) \geq \Omega(\log(w)) + C(1/w^{1/3}, l/2)$.

- Repeating the above procedure inductively $\Theta(\log(l))$ times, we get $C(1/w^{1/3}, l) \geq \Omega(\log(l) \cdot \log(w))$.

This ends proof of (2). □

We now prove the above two lemmas:

Proof of (a) Lemma 4.11. Assume without loss of generality that Alice had sent the first bit in the (α, l) protocol \mathcal{P} . Let $S \subseteq \{0, 1\}^l$ be the good set of \mathcal{P} . Let $S_0, S_1 \subseteq S$ be the sets of strings for which Alice sends 0 and 1 as the first bit respectively. Let S_b be the larger among S_1 and S_0 ; then, clearly $|S_b| \geq |S|/2$. Define a new protocol \mathcal{P}' which is exactly the same as \mathcal{P} without sending the first bit. So Alice and Bob assume the first bit to be b and continue working just like \mathcal{P} . Then, \mathcal{P}' is a $c - 1$ bit protocol with the good set S_b . Hence \mathcal{P}' is a $c - 1$ bit $(\alpha/2, l)$ protocol for FORK. □

Proof of (b) Lemma 4.12. The following proof is taken from [KN97]:

We will need the following combinatorial lemma that we state without proof:

Lemma 4.13. *Consider $n \times n$ boolean matrix. Let m be the number of 1s in it, and m_i be the number of 1s in the i^{th} row. Denote by $\alpha = m/n^2$, the density of 1s in the matrix, and by $\alpha_i = m_i/n$, the density of 1s in row i . Then at least one of the following holds:*

- (a) *There is some row i for which $\alpha_i \geq \sqrt{\alpha/2}$*
- (b) *The number of rows for which row density is greater than $\alpha/2$ is at least $\sqrt{\alpha/2} \cdot n$.*

Now we proceed to prove the amplification lemma. Let S be the good set for the (α, l) protocol being considered. Consider a matrix A whose rows and columns correspond to strings from $\Sigma^{l/2}$. An entry corresponding

to row u and column v of A is a 1 if and only if $u \circ v \in S$. Since $|S| \geq \alpha \Sigma^l$, the density of this matrix is at least α . Applying the above lemma to this matrix, we get that it satisfies either (a) or (b).

Suppose the matrix satisfied (a), then, there exists a row corresponding to some string $u \in \Sigma^{l/2}$ that has density at least $\sqrt{\alpha/2}$. On input $x, y \in \Sigma^{l/2}$, Alice and Bob use the original (α, l) protocol on the strings $u \circ x$ and $u \circ y$. Since we are prefixing both x and y with the same string u , the answer to FORK on these strings surely lies in the second half of the strings. The protocol succeeds whenever $u \circ x$ and $u \circ y$ are in S . So define $S' = \{x \mid ux \in S\}$. Then, S' is good for this protocol and $|S'| = \sqrt{\alpha/2} \geq \sqrt{\alpha}/2$. Hence the lemma goes through if (a) happens.

Suppose the matrix satisfies (b). Let $S' \subseteq S$ be the set of rows for which density of 1s is at least $\alpha/2$. We will show that there exists functions $f, g : \Sigma^{l/2} \rightarrow \Sigma^{l/2}$ and a set $S'' \subseteq S'$ such that the following holds:

1. $\forall x \in S'', x \circ f(x) \in S$,
2. $\forall y \in S'', y \circ g(y) \in S$,
3. $\forall x, y \in S''$, the strings $f(x)$ and $g(y)$ are different in all coordinates,
4. $|S''| \geq \sqrt{\alpha}/2$.

Assuming that we can show the existence of f and g with the above properties, the new protocol is as follows: On input $x, y \in \Sigma^{l/2}$, Alice and Bob use the original c -bit protocol (α, l) protocol on $x \circ f(x)$ and $y \circ g(y)$. By property 3, the answer to FORK on these strings cannot lie in the second half. Because of property 4, this is an $(\sqrt{\alpha}/2, l/2)$ protocol.

The existence of f and g with the above mentioned properties is proved by a probabilistic argument in [KN97].

□

Observation 4.14. *We have just seen that any monotone circuit family computing STCON has a depth asymptotically greater than $\log^2(n)$. It's easy to see that STCON can be computed by a circuit family with depth $O(\log^2(n))$. This can be done by using Savitch's trick of divide and conquer for reachability. Hence, the above result is asymptotically tight for STCON.*

4.4 A general technique for a $\log^2 n$ lower bound to monotone circuit depth

In this section we look at a general technique to prove a $\log^2 n$ lower bound on the monotone circuit depth of a function f which we will construct using a function g that has a deterministic communication complexity significantly larger than its non-deterministic communication complexity.

Let $g : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$. Let C_1, C_2, \dots, C_t be a value-monochromatic cover (covering both 0s and 1s) of the best non-deterministic protocol for R_g . Define the relation $\Psi \subseteq \{0, 1\}^n \times \{0, 1\}^n \times \{1, 2, \dots, t\}$ such that $(x, y, i) \in \Psi \iff (x, y) \in C_i$. Note that for each (x, y) there is at least one i for which $(x, y, i) \in \Psi$, and also, there could be more than one such i .

Observation 4.15. $R_g \leq \Psi$

Now, we construct a function $f : \{0, 1\}^t \rightarrow \{0, 1\}$. The input to f can be thought of as a subset of C_1, \dots, C_t . f is defined as: $f(z_1, z_2, \dots, z_t) = 1$ if there exists a column y in M_g such that $\forall i, y \in C_i \implies z_i = 1$. f is monotone by definition. We want to prove a non-trivial lower bound on the monotone KW game of f .

Lemma 4.16. $\Psi \leq KW_f^+$

Proof. Given an $x \in \{0, 1\}^n$, Alice constructs $x' \in \{0, 1\}^t$ by assigning $x'_i = 0$ if the row x belongs to C_i and 1 otherwise. $f(x') = 0$ because for each column y of M_g , there is an i such that C_i covers (x, y) and so $y \in C_i$ but $z_i = 0$. Hence y is not a witness column for $f(x')$.

Bob is given an $y \in \{0, 1\}^n$. He constructs $y' \in \{0, 1\}^t$ by assigning $y'_i = 1$ if column y belongs to C_i , 0 otherwise. $f(y') = 1$ by definition of f , because the column y itself serves as the witnessing column for $f(y')$.

Alice and Bob play the communication game on f with inputs as x' and y' respectively and obtain an answer i such that $x'_i = 0$ and $y'_i = 1$. By the way we defined x' and y' , it can be seen that row x belongs to C_i and column y belongs to C_i . Hence Alice and Bob now know that the value-monochromatic rectangle to which both x and y belong to is i , and hence i is a legal answer to the communication game on Ψ . \square

Corollary 4.17. *From the above lemma and observation*

$$R_g \leq \Psi \leq KW_f^+, \text{ and so } CC(R_g) \leq CC(\Psi) \leq CC(KW_f^+).$$

Lemma 4.18. *If $N(R_g) = O(\log(n))$ and $CC(R_g) = \Omega(\log^2(n))$, then, $d^+(f) = CC(KW_f^+) \geq \log^2 n$*

Proof. Note that f works on $t = 2^{N(R_g)}$ variables. So if $N(R_g) = O(\log(n))$, f works on n variables. And since $CC(R_g) \leq CC(KW_f^+)$, $\Omega(\log^2 n) \leq CC(KW_f^+)$. \square

An example for a function that has non-deterministic communication complexity $O(\log(n))$ and deterministic communication complexity $\log^2 n$ is $\text{DISJ}_{n, \log(n)}$. $CC(\text{DISJ}_{n, \log(n)}) = \theta(\log^2 n)$ - follows from the fact 4.5 that R_{DISJ} is full rank.

Claim. $N(\text{DISJ}_{n, \log n}) = O(\log(n))$

Proof. It is easy to see that $N^0(\text{DISJ}_{n, \log(n)}) \leq \log(n)$ - Alice is given X , Bob is given Y , $|X| = |Y| = \log n$, A guess $i \in [n]$ is made. Alice sends a 1 if

$i \in X$. Bob announces that the sets are not disjoint if $i \in Y$, and announces that the sets are disjoint otherwise.

Suppose we can prove that indeed $N^1(\text{DISJ}_{n, \log n}) = O(\log(n) + \log \log(n))$, then, $N(\text{DISJ}_{n, \log n}) = O(\log n)$ as follows: Let f be short for $\text{DISJ}_{n, \log n}$. From Observation 3.16, we get $C^0(f) = 2^{N^0(f)}$ and $C^1(f) = 2^{N^1(f)}$. Using the fact that $C(f) = C^0(f) + C^1(f)$, and substituting $O(\log n)$ for both $N^1(f)$ and $N^0(f)$, we have: $C(f) = O(n)$. So $N(f) = O(\log n)$. It can be shown using a probabilistic argument that $N^1(f) = O(\log n + \log \log n)$. This is given in detail in [KN97]. \square

Bibliography

- [Ajt83] M. Ajtai, Σ_1^1 -formulae on finite structures, *Annals of Pure and Applied Logic* **24** (1983), 1–48.
- [BS90] Ravi B. Boppana and Michael Sipser, *The complexity of finite functions*, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, 1990, pp. 757–804.
- [Dun88] P. E. Dunne, *The complexity of Boolean networks*, Academic Press Professional, Inc., San Diego, CA, USA, 1988.
- [FSS84] Furst, Saxe, and Sipser, *Parity, circuits, and the polynomial-time hierarchy*, *MST: Mathematical Systems Theory* (1984).
- [Got66] D. H. Gottlieb, *A certain class of incidence matrices*, *Proceedings of the American Mathematical Society* **17** (1966), no. 6, 1233–1237.
- [Hås86] Johan Håstad, *Almost optimal lower bounds for small depth circuits*, *STOC: ACM Symposium on Theory of Computing (STOC)*, 1986, pp. 6–20.
- [Hås98] ———, *The shrinkage exponent of De Morgan formulas is 2*, *SIAM J. Comput* **27** (1998), no. 1, 48–64.

- [Juk04] Stasys P Jukna, *On graph complexity*, ECCCTR: Electronic Colloquium on Computational Complexity, technical reports, 2004.
- [Juk09] ———, *Boolean function complexity: Advances and frontiers*, 10 October 2009.
- [KKN95] Karchmer, Kushilevitz, and Nisan, *Fractional covers and communication complexity*, SIJDM: SIAM Journal on Discrete Mathematics 8 (1995).
- [KN97] Eyal Kushilevitz and Noam Nisan, *Communication complexity*, Cambridge University Press, New York, 1997.
- [KRW91] Mauricio Karchmer, Ran Raz, and Avi Wigderson, *Super-logarithmic depth lower bounds via the direct sum in communication complexity*, Proceedings of 6 th Structures in Complexity Theory, 1991, pp. 299–304.
- [KW88] Mauricio Karchmer and Avi Wigderson, *Monotone circuits for connectivity require super-logarithmic depth*, STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing (New York, NY, USA), ACM, 1988, pp. 539–550.
- [MSPZ92] N. Pippenger M. S. Paterson and U. Zwick, *Optimal carry save networks*, Cambridge University Press, 1992, pp. 174–201.
- [Raz90] Alexander A. Razborov, *Applications of matrix methods to the theory of lower bounds in computational complexity*, Combinatorica 10 (1990), no. 1, 81–93.
- [Sha49] C. Shannon, *The synthesis of two-terminal switching circuits*, Bell System Technical Journal 28 (1949), 59–98.

- [Smo87] Roman Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, STOC: ACM Symposium on Theory of Computing (STOC) (New York City), ACM, 25–27 May 1987, pp. 77–82.
- [Vol99] Heribert Vollmer, *Introduction to circuit complexity: A uniform approach*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.
- [Weg87] Ingo Wegener, *The complexity of Boolean functions*, B. G. Teubner, and John Wiley & Sons, 1987.
- [Yao79] Yao, *Some complexity questions related to distributive computing*, STOC: ACM Symposium on Theory of Computing (STOC), 1979.
- [Yao85] ———, *Separating the polynomial-time hierarchy by oracles*, FOCS: IEEE Symposium on Foundations of Computer Science (FOCS), 1985.

Appendix

Markov's Inequality

Markov's inequality states that if X is any random variable, and $a > 0$, then, the following holds:

$$\Pr [|X| \geq a] \leq \frac{|E|}{a}$$

Cauchy-Schwarz Inequality

The Cauchy-Schwarz inequality is the following:

$$\left| \sum_{i=0}^n x_i y_i \right|^2 \leq \sum_{j=1}^n |x_j|^2 \sum_{j=1}^n |y_j|^2$$

Shannon's argument

The following theorem that shows that most functions require exponential size circuits was proved by Shannon:

Theorem 1 (Shannon). Let $\varepsilon > 0$. The ratio of all n -ary Boolean functions that can be computed by circuits over the basis $\{\wedge, \vee, \sim\}$ with $(1 - \varepsilon) \frac{2^n}{n}$ gates approaches 0 as $n \rightarrow \infty$.