IITM-CS6840: Advanced Complexity Theory March 26, 2012

Lecture No. 44 : Neciprouk's Lower Bound

Lecturer: Jayalal Sarma M.N. *Scribe: Sajin Koroth*

THEME: Circuit Complexity-Lower Bounds using Restrictions

LECTURE PLAN: Today we will see Neciprouk's method for lower bounding the formula size of a function. This techniques utilizes the method of restrictions. We will also introduce method of random restrictions and provide some motivation. Neciprouk's lower bound uses a counting strategy which relates the number of different functions produced by restrictions on variables to, the number of times a variable occur in the formula (which together form the number of leaves). After obtaining Neciprouk's lower bound we will show that it is tight for general functions by constructing an explicit function where Neciprouk's lower bound matches the upper bound. We will also introduce the notion of Random Restriction and introduce Sabbottovskaya's theorem which will be proved in the next lecture.

1 Neciprouk's Lower bound for bounded fan-in formulae

Firstly we will define an appropriate notion of restriction to the function which will lead to the lower bound. Let $\{v_1, \ldots, v_p\}$ be a partition of set of input indices, $[n]$. For a boolean function $f : \{0,1\}^n - \{0,1\}$. The number $r_i(f)$ is defined as the number of different functions obtained by restricting f on variable indices other than v_i. After a specific restriction $\sigma(x_i = b_i \mid i \in \bigcup_{j \in [n] \setminus v_i} \{j\})$, the function $f \mid^{\sigma}_{v_i}$ will be from $\{0,1\}^{l_i}$ to $\{0,1\}$ where l_i is $|v_i|$. Note that for two different restrictions $\sigma, \sigma^{'}$ outside v_i the resulting restricted functions $f \mid^{\sigma}_{v_i}, f \mid^{\sigma^{'}}_{v_i}$ could be the same. For example consider f to be the AND function of n variables, and let the partition be $v_i = \{\frac{n}{k}(i-1), \ldots, \frac{n}{k}\}$, that is we divide the variable indices to consecutive sets of size k. Now choose any arbitrary v_i, two restrictions $\sigma, \sigma^{'}$ which are different for some x_j (i.e. $\sigma(x_j) \neq \sigma^{'}(x_j)$) and such that they both set at least on variable to zero makes the restricted functions $f \mid^{\sigma}_{v_i}, f \mid^{\sigma^{'}}_{v_i}$ to be the constant function 0 as AND function is zero if any of the inputs are zero.

With the above definition of restrictions the following theorem can be proved.

Theorem 1. *Over the basis Ω over gates of fan-in k, given a partition of the input indices $\{v_1, \ldots, v_p\}$*

$$L(f) \geq \frac{1}{k+2} \sum_{i=1}^{p} \log(r_i(f))$$

where $L(f)$ denotes the number of leaves in a formula which correctly computes f over Ω

The proof strategy is as follows, we will upper bound the number of different restricted functions using the formula T_f for f over Ω, and the lower bound will be obtained in terms of number of leaves of of the formula. Hence we can derive from such an upper bound a lower bound on the total number of leaves in the circuit. Since the circuit is a formula it will be a tree and the total number of nodes in a tree and number of leaves in a tree are linearly related. Hence we get a lower bound for the size of the circuit.

Proof. Note that applying the restriction σ outside v_i to f is equivalent to fixing inputs of the formula T_f for input wires x_l such that l comes from $v_j, j \neq i$ to the value $\sigma(x_l)$. The tree T_f remains a tree, say T_f^σ after the restriction on lesser number of gates which computes $f\mid_{v_i}^{\sigma}$. To lower bound the number of different restricted functions we will use the following definitions: □

Definition 2 (Combinator)**.** A combinator is a gate in T_f^σ such that at most $k-2$ of its inputs are from outside T_f^σ, in other words the sub-trees rooted at those $l \leq k-2$ inputs does not contain an input wire which is not fixed by σ. The remaining inputs (at least 2) come from within T_f^σ.

By the above definition each combinator gate produce at most 2^{k-2} different functions corresponding to different restrictions of its $l \leq k-2$ inputs from outside v_i.

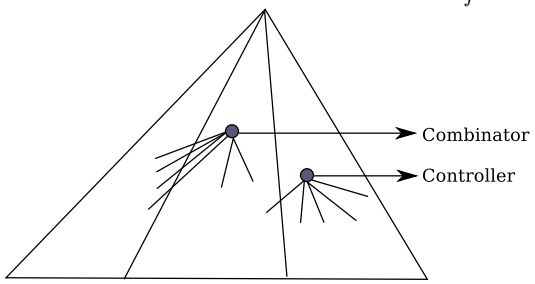
Definition 3 (Controller)**.** A controller is a gate in T_f^σ such that $k-1$ of its inputs are from outside T_f^σ, in other words the sub-trees rooted at those $k-1$ inputs does not contain an input wire which is not fixed by σ. The remaning input (exactly one) comes from within T_f^σ.

Note that a controller gate is different from combinator because only one input is not fixed and hence the function computed by such a function can be written as $ag \oplus b$ where $a, b \in \{0,1\}$ and g is the input to the gate from within T_f^σ and a, b depends on the type of the gate and the restriction σ. For example if the gate is an $\vee$ gate and the $k-1$ input gatesare set to 0 by σ, then $a = 1, b = 0$, because the output of the gate be just be g. Hence the number of different functions a controller gate can produce is at most 4 corresponding to various choices of a, b.

Also note that any gate in T_f^σ is by definition is either a combinator or a controller. Figure 1 illustrates combinators and controllers.

Since T_f is a formula a variable can appear several times. Let n_i denote the number of times variables indexed by v_i appears in T_f. Clearly $n_i \geq l_i$. Note that number of internal vertices's in tree of degree k is at most $n_i - 1$ (this case happens when all internal nodes are of degree 2, for $k > 2$ the number of internal nodes will be much lesser as expansion

Figure 1: Restricted Formula T_f^σ



will happen with a faster exponent). Hence the total number of vertices's in T_f^σ is upper bounded by $2n_i - 1$ where σ is a restriction outside v_i. The number of controllers in T_f^σ is upper bounded by the number of edges in T_f^σ as each controller requires one edge from within T_f^σ. Number of combinators is upper bounded by half the number of nodes in T_f^σ because each combinator requires two vertices's from T_f^σ, and since T_f^σ is a formula each vertex has out-degree 1 and hence they cannot be part of another combinator. Hence the total number combinator is at most $n_i - 1$. So the total number of functions which can be produced by different $\sigma' s$ are upper bounded by number of T_f^σ which have at least one gate computing a different function from each other. Hence the total number of restricted functions is

$$\begin{aligned} r_i(f) &\leq \#'_f s \text{ combinators produce } \times \#'_f s \text{ controllers produce} \\ &\leq \left(2^{k-2}\right)^{\#\text{combinators}} \times (4)^{\#\text{controllers}} \\ &\leq \left(2^{k-2}\right)^{n_i-1} \times (4)^{2n_i-2} \\ &= 2^{(n_i-1)(k-2)+(n_i-1)4} \\ &= 2^{k+2(n_i-1)} \end{aligned}$$

Now let us analyze the following summation $\sum_{i=1}^{p} \log(r_i(f))$

$$\begin{aligned}
\sum_{i=1}^{p} \log(r_i(f)) &\leq \sum_{i=1}^{p} \log\left(2^{(k+2)(n_i-1)}\right) \\
&= \sum_{i=1}^{p} (k+2)(n_i-1) \\
&= (k+2)\sum_{i=1}^{p} (n_i-1) \\
&\leq (k+2)\sum_{i=1}^{p} (n_i) \\
&= (k+2)\,L(f) \;\because \text{the leaves in tree } T_f \text{ are the input variables}
\end{aligned}$$

Hence we have proved the theorem.

2 Tightness of Neciprouk's Lower bound

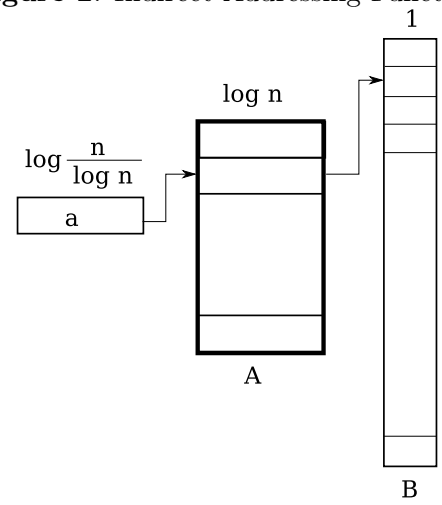
To prove the tightness we will give an explicit function f which requires $\Omega\left(\frac{n^2}{\log n}\right)$ size using Neciprouk's lower bound which also has a matching upper bound, i.e. a formula which computes f of size $O\left(\frac{n^2}{\log n}\right)$. The function is the indirect address function $f : \{0,1\}^{2n+\log n-\log\log n} \to \{0,1\}$ which is defined as follows : the input can be thought of as a three tuple (a, A, B) where a of $\log\left(\frac{n}{\log n}\right)$ bits, B is of n bits and A if of n bits also. The input is interpreted as encoding of an index and two tables, A is a table of n $1-$bit entries, B is a table of $\frac{n}{\log n}$ $\log n$ bit entries and a is an index into table B, and $B[a]$ intern indexes a bit in the table A. Output of the function is thus $A[B[a]]$. Figure 2 illustrates this interpretation of the input.

We will first show the upper bound, i.e. a formula computing the indirect addressing function. The key ideas is the following, we can construct a $n:1$ MUX from a $2:1$ MUX recursively. To implement a $n:1$ MUX we need to implement two $\frac{n}{2}:1$ MUX and a $2:1$ MUX. Hence we get the following recurrence for the size of an $n:1$ MUX,

$$T(n) = 2T\left(\frac{n}{2}\right) + 3$$

Hence we get that an $n:1$ MUX can be built using $O(n)$ gates (obtained by solving above recurrence). For the first table addressing we need a $\frac{n}{\log n}:1$ MUX which requires size $O(\frac{n}{\log n})$ and for i^{th} bit of selector bit we need 2^i duplicate wires. Hence the total number of duplications of selector bits is $\sum_{i=1}^{\log n} 2^i = O(n)$, and for each of these we will need the $\frac{n}{\log n}:1$ MUX, hence the total size is $O(n \times \frac{n}{\log n})$. Hence the upper bound.

Figure 2: Indirect Addressing Function



Now will show a matching lower bound using Neciprouk's method. To get v_i we will divide the n bits which represent the contents of the table B into $\log n$ size equal chunks. Hence there $\frac{n}{\log n}$ such partitions. For each setting of different A's the functions which are restricted every where other than one table entry of B (of $\log n$ bits) will have different output for some input x. Hence the number of possible functions for the restriction is 2^n and there are at least $\frac{n}{\log n}$ of such partitions. Hence by Neciprouk's method,

$$L_\Omega(f) \geq \frac{1}{k+2} \sum_{i=1}^{\frac{n}{\log n}} \log 2^n \geq \frac{n^2}{(k+2)\log n}$$

Hence we have proved that indirect addressing function is tight for Neciprouk's method.

3 Random Restrictions

We can generalize our method of restrictions by introducing random restrictions. A random restriction is a randomly picked function from the following family of functions $\delta : [n] \to \{0, 1, *\}$ where δ is interpreted as a restriction as follows :

$$\delta(i) \quad = \quad \begin{cases} 0 & \text{assign 0 to } x_i \\ 1 & \text{assign 1 to } x_i \\ * & x_i \text{ is unassigned} \end{cases}$$

Randomly choosing such a function is equivalent to for each index $i \in [n]$ first tossing a coin

to decide to leave it unassigned or to fix it, and then if it is decided to fix it, toss another coin and fix the index according to the outcome of the toss.

Definition 4 (Random Restriction on all but k variables)**.** It is the set of all functions δ described as above with the additional requirement that exactly k indices are unassigned.

$$R_k = \{\delta \mid | \{i \mid \delta(i) = *\} \mid= k\}$$

It is interesting to know on a random restriction what is the expected size of the formula obtained after applying the restriction. Let $L(f)$ denote the minimum size of a formula computing f. Then we are interested in the following quantity

$$\mathbb{E}_\delta \left[L \left(f \mid_\delta \right) \right]$$

Sabbotavskaya who was a student of Lupanov studied the problem of expected decrease in formula size for a function f when hit with a random restriction. She with her seminal paper in 1961 on formula lower bound based on random restrictions started off the sub area of circuit lower bounds using random restrictions.

Even though we will not presenting Sabbotavskaya's method in this lecture we will prove a trivial lower bound on $\mathbb{E}_\delta \left[L \left(f \mid_\delta \right) \right]$.

Consider the family R_k of random restrictions which leaves exactly k variables unassigned. When we apply such a random restriction to the formula which achieves size $L(f)$, each leaf of it being an input variable x_i (which might occur more than once) gets fixed to a value with probability $1 - \frac{k}{n}$. But since there could be multiple occurrence of the same variable, given that a particular leaf is fixed the conditional probability that another leaf which also represent the same input gets fixed is 1. Hence at most $\left(\frac{k}{n}\right) L(f)$ leaves remains unassigned on average (it could be much lesser also if there are multiple occurrences of a variable as pointed earlier). Hence we get that

$$\mathbb{E}_\delta \left[L \left(f \mid_\delta \right) \right] \leq \left(\frac{k}{n} \right) L(f)$$

Sabbotavskaya improved the above lower bound to lower bound of the form below. People have also proved the limits to such lower bounds by proving upper bounds on Γ.

$$\mathbb{E}_\delta \left[L \left(f \mid_\delta \right) \right] \leq \left(\frac{k}{n} \right)^\Gamma L(f)$$

Theorem 5 (Sabbotavskaya (1961))**.**

$$\mathbb{E}_\delta \left[L \left(f \mid_\delta \right) \right] \leq \left(\frac{k}{n} \right)^{\frac{3}{2}} L(f)$$

The following theorem can be proved from Sabbotavskaya's theorem.

Theorem 6.

$$\Pr_{\delta \in R_k}\left[L(f\mid_\delta) \geq 4\left(\frac{k}{n}\right)^{\frac{3}{4}} L(f)\right] \geq \frac{3}{4}$$

Proof. Applying Markov's Inequality to Sabbotavskaya's theorem we get that

$$\Pr_{\delta \in R_k}\left[L(f\mid_\delta) \geq 4\left[\mathbb{E}_\delta\left[L\left(f\mid_\delta\right)\right]\right]\right] \leq \frac{\mathbb{E}_\delta\left[L\left(f\mid_\delta\right)\right]}{4\mathbb{E}_\delta\left[L\left(f\mid_\delta\right)\right]} \leq \frac{1}{4}$$

Hence it implies that

$$\Pr_{\delta \in R_k}\left[L(f\mid_\delta) \geq 4\left(\frac{k}{n}\right)^{\frac{3}{4}} L(f)\right] \geq \left[1 - \frac{1}{4} = \frac{3}{4}\right]$$

□

Corollary 7. *There exists a δ such that the formula size goes down by $\left(\frac{k}{n}\right)^{\frac{3}{4}}$ factor.*

The above corollary can be applied to obtain a lower bound for the parity function on n inputs. Note that parity function on even **any** single input with other inputs fixed to **any** binary string is not a constant function. Hence setting $k = 1$ for the above corollary we get that there exists a $\delta \in R_1$ such that

$$L\left(f\mid_\delta\right) \leq 4\left(\frac{1}{n}\right)^{\frac{3}{4}} L(f)$$

But since parity is a non-constant on any input and any $\delta \in R_1$, $L(f\mid_\delta) \geq 1$. Combining with the above inequality we get that

$$L(f) \geq \frac{n^{\frac{3}{4}}}{4}$$

Coming back to the value of Γ , Karpchenkov in 1969 proved that $\Gamma \leq 2$. Hastad in 1998 proved that $\Gamma = 2$. We will discuss proof of Sabbotavskaya's theorem in tomorrows lecture.