CS277: Concrete Complexity **Fall 1996**

Lecture 9: September 24, 1996

Lecturer: Uri Zwick *Scribe: Felix Wu*

9.1 Khrapchenko's Theorem: Alternate Proof

Theorem 9.1 (Khrapchenko) *For every $A \subseteq f^{-1}(0)$, $B \subseteq f^{-1}(1)$,*

$$L_{U_2}(f) \geq \frac{|A \otimes B|^2}{|A||B|}.$$

9.1.1 Rychkov's Lemma

In order to give an alternate proof for Theorem 9.1, we will use the following lemma.

Lemma 9.2 (Rychkov) *If f is computed using a de Morgan (i.e., U_2) formula of size k, and if $A \subseteq f^{-1}(0)$, $B \subseteq f^{-1}(1)$, then there exist pairs of sets $(A_1, B_1), \ldots, (A_k, B_k)$, where $A_1, \ldots, A_k \subseteq A$, $B_1, \ldots, B_k \subseteq B$, such that the following conditions hold:*

1. *For $i \neq j$, $A_i \times B_i$ and $A_j \times B_j$ are disjoint.*
2. $\displaystyle\bigcup_{i=1}^{k} A_i \times B_i = A \times B$.
3. *For every $1 \leq i \leq k$, $\exists j$: $x \in A_i$, $y \in B_i \Rightarrow x_j \neq y_j$.*

(Note that $\times$ in this case indicates the Cartesian product, not the edge set defined earlier.)

This lemma states that the space $A \times B$ can be partitioned into no more than k subspaces in such a way that within each subspace, there is some position which distinguishes all of the vectors in A_i from all of the vectors in B_i. As can be seen in Figure 9.1, this means that the rectangle representing $A \times B$ can be divided into no more than k disjoint (combinatorial) rectangles, such that within each rectangle condition 3 holds.

Proof: (by induction on k)

Basis: When $k = 1$, the function f must be either x_i or $\overline{x_i}$. In this case we take $A_1 = A$, $B_1 = B$. This is clearly a (trivial) partition of $A \times B$, and i serves as the distinguishing position.

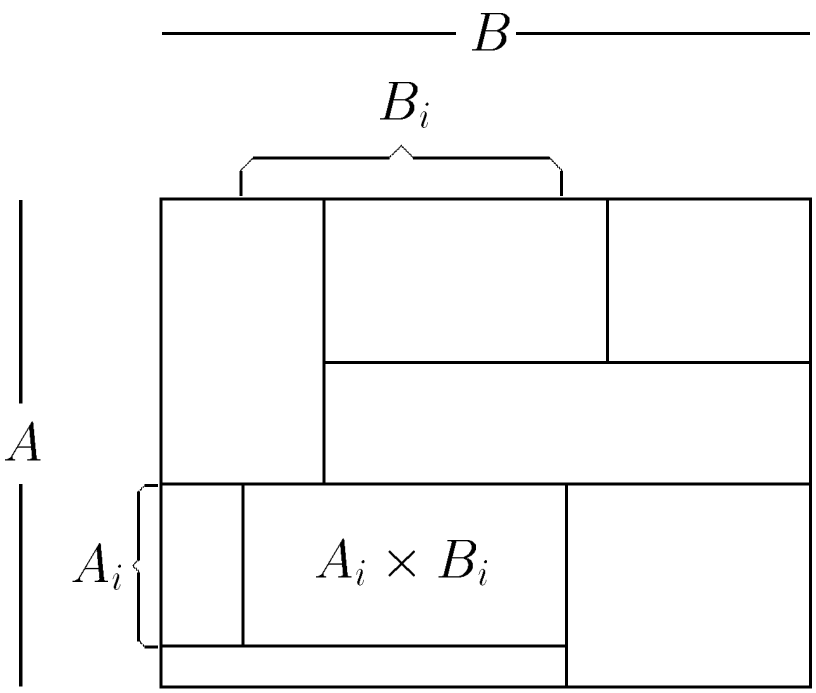

Figure 9.1: Partition of $A \times B$

Induction Step: Suppose f has a formula of size k. Without loss of generality, we can assume that the only gates are $(\vee, \neg)$. If $f = \neg f_1$, then $f^{-1}(0) = {f_1}^{-1}(1)$, $f^{-1}(1) = {f_1}^{-1}(0)$. Thus, if f_1 can be partitioned appropriately, f can also be partitioned appropriately simply by swapping the roles of A and B. Hence, if the lemma holds without negation, the lemma also holds with negation.

The interesting case to consider is when f is of the form $f_1 \vee f_2$. Let k_1 be the size of f_1, and k_2 be the size of f_2. Now set $A' = A \cap {f_1}^{-1}(0)$, $A'' = A \setminus A'$. Clearly, $A' \subseteq {f_1}^{-1}(0)$. Also, $A'' \subseteq {f_2}^{-1}(0)$, since A'' consists of all inputs which result in a value of 0 for f, but a value of 1 for f_1. Furthermore, B is a subset of both ${f_1}^{-1}(1)$ and ${f_2}^{-1}(1)$.

Hence, by the induction hypothesis, the space $A' \times B$ can be divided into no more than k_1 rectangles which satisfy condition 3 above. Also, the space $A'' \times B$ can be divided into no more than k_2 such rectangles. Since $k_1 + k_2 = k$, this implies that $A \times B$ can be divided into no more than k appropriate rectangles. ∎

9.1.2 Proof of Khrapchenko's Theorem

We are now ready to prove Khrapchenko's theorem.

Proof: Consider a partition of $A \times B$ into rectangles as in the lemma. Let $R_i = (A \otimes B) \cap (A_i \times B_i)$. Then in the rectangle $A_i \times B_i$, each row and column can contain at most one element of R_i. This is because each element of A_i differs from each element in B_i in one particular position j, whereas (x, y) is in $A \otimes B$ only if x and y differ in exactly one point. Hence, for any given $x \in A_i$, the only possible $y \in B_i$ for which $(x, y) \in R_i$ is one which differs from x exactly in position j.

As a result, we have that $|R_i| \leq |A_i|$ and $|R_i| \leq |B_i|$, which implies $|R_i|^2 \leq |A_i||B_i|$. Furthermore, $A \otimes B = \bigcup_{i=1}^{k} R_i$, with all of the R_i's disjoint. Since each R_i includes the elements of the edge set in the ith rectangle, together they include all elements of the edge set.

Therefore, we have

$$
\begin{aligned}
|A \otimes B| &= \sum_{i=1}^{k} |R_i| && \text{(since the } R_i\text{'s are disjoint)} \\
|A \otimes B|^2 &= \left(\sum_{i=1}^{k} |R_i|\right)^2 \\
&\leq k \cdot \sum |R_i|^2 && \text{(by the Cauchy-Schwartz inequality)} \\
&\leq k \cdot \sum |A_i||B_i| \\
&= k|A||B|.
\end{aligned}
$$

Since $k = L_{U_2}(f)$, this proves the theorem. ∎

9.2 The Effect of Random Restrictions on Formula Size

Definition 9.3 *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, a* partial assignment *is a function $\rho : [n] \to \{0, 1, *\}$, where we understand $*$ to mean that the corresponding variable is unassigned. (Note that $[n]$ means the set $\{1, \ldots, n\}$.) We let f_ρ indicate the resulting* restriction *of f.*

Let $R^n_m = \{\rho : [n] \to \{0, 1, *\} : |\rho^{-1}(*)| = m\}$. In other words, R^n_m is the set of all partial assignments which leave exactly m variables unassigned. What we will be interested in is the random restriction f_ρ that results from choosing a random partial assignment from R^n_m.

In particular, suppose we choose $\rho \in R^n_{\epsilon n}$ uniformly at random. Over any basis, we know that $E[L(f_\rho)] \leq \epsilon L(f)$. This is because each leaf has a probability ϵ of remaining unassigned, and hence, we expect an ϵ-fraction of the original formula complexity (measured in leaves) to remain.

However, over U_2, we should do even better in terms of reducing the formula complexity, because $\wedge$ and $\vee$ gates have the potential to be determined by one input alone. Thus, certain assignments to leaves might allow us to eliminate gates further into the tree, thereby eliminating leaves that were not directly assigned by ρ. In particular, the *expected* formula complexity ought then to be smaller.

We will show that over U_2, $E_{\rho \in R^n_{\epsilon n}}[L(f_\rho)] \leq \epsilon^{1.5} \cdot L(f) + \frac{1}{3}$.

Consider the implication of this result for the function $f = XOR_n$. Take $\epsilon = \frac{1}{n}$. Then we have:

$$E[L(f_\rho)] \leq (\tfrac{1}{n})^{1.5} L(XOR_n) + \tfrac{1}{3}.$$

But the LHS is 1, since we will have assigned values to all but one input, and this input will determine the value of the XOR. Hence, we get $L_{U_2}(XOR_n) = \Omega(n^{1.5})$.

9.3 Shrinkage of Restricted Formulae over U_2

The following lemma characterizes the extra shrinkage we get for restricted formulae over U_2.

Lemma 9.4 (Subbotovskaya) *If $L(f) > 1$,*

$$E_{\rho \in R^n_{n-1}}[L(f_\rho)] \leq (1 - \tfrac{1.5}{n}) L(f).$$

This lemma states that in assigning a random value to a randomly chosen variable, the formula size shrinks by a factor of at least $(1 - \frac{1.5}{n})$. Note that for a general basis, we only get a factor of $(1 - \frac{1}{n})$, since in general, we can do no better than to eliminate exactly the leaves to which we assigned a value. Also note that if $L(f) = 1$, i.e. $f = x_i$, $E[L(f_\rho)] = 1 - \frac{1}{n}$, since with probability $\frac{1}{n}$ we reduce the formula complexity to 0 by assigning a value to x_i, and with the remaining probability, the formula complexity remains at 1.

Proof: To begin with, we write out the definition of $E[L(f_\rho)]$ and rewrite it slightly to focus our attention on the quantity representing the amount saved in any one restriction.

$$\begin{aligned} E[L(f_\rho)] &= \tfrac{1}{2n} \sum_{i=1}^{n} \sum_{b=0,1} L(f_{x_i \leftarrow b}) \\ &= L(f) - \tfrac{1}{2n} \sum_{i=1}^{n} \sum_{b=0,1} (L(f) - L(f_{x_i \leftarrow b})). \end{aligned}$$

What we will focus on is computing the terms in this last sum.

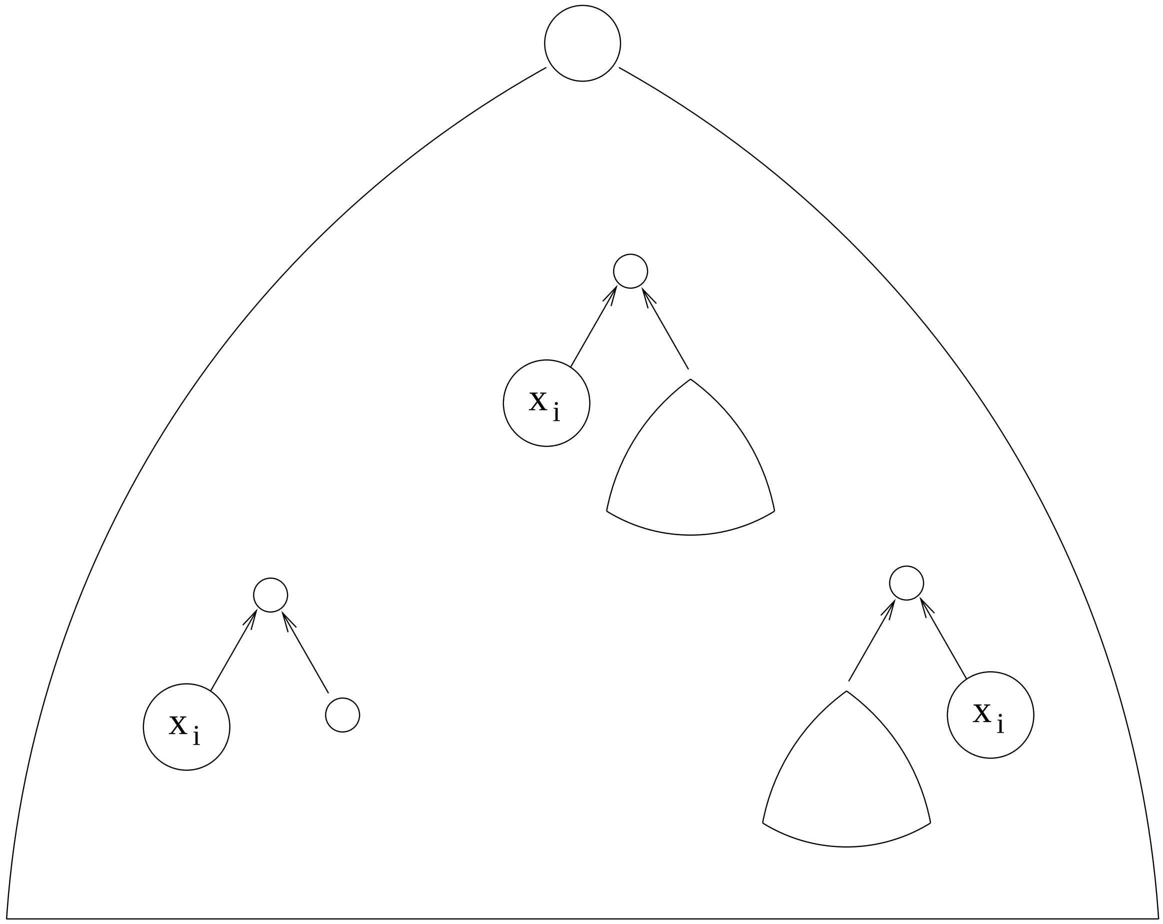

Figure 9.2: x_i leaves in circuit for f

Consider a minimal formula for f. Let l_i be the number of occurrences of the input x_i in the formula. (Hence, $\sum_{i=1}^{n} l_i = L(f)$.)

Consider all of the leaves labelled with x_i, as well as their parents and siblings. (See Figure 9.2.)

Claim 9.5 *Any x_i leaf cannot have a sibling whose subtree contains an x_i leaf.*

Proof: Suppose we do have a situation in which the sibling subtree of an x_i leaf also contains an x_i leaf. Without loss of generality, suppose the parent of these nodes is an $\wedge$-gate. (See Figure 9.3.) Note then that if $x_i = 0$, the value of the right subtree doesn't matter. Hence, we might as well replace the x_i leaf embedded in that subtree with a 1, since we never use its value when it is 0. However, since we can eliminate a leaf and compute the

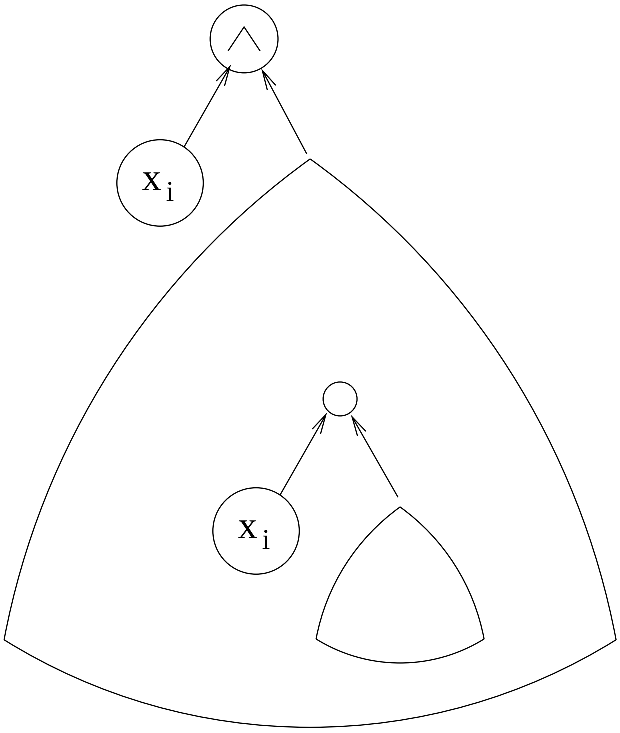

Figure 9.3: Impossible positioning of x_i leaves

same function, we have contradicted our assumption that this is a minimal formula. Hence, the situation described above is impossible. ■

For any parent of an x_i leaf, there exists some assignment to x_i which makes the sibling irrelevant. In such a situation, an assignment to x_i allows us to remove both the x_i node and its sibling, which contains at least one leaf. Hence, we have:

$$\sum_{b=0,1} L(f) - L(f_{x_i \leftarrow b}) \geq 3l_i.$$

That is, for each of l_i leaves labelled x_i, for one assignment to x_i we will be able to remove at least one leaf (the x_i leaf itself), and for the other assignment, we will be able to remove at least two leaves (the leaf itself and the leaves in its sibling). Note that the fact that an x_i leaf is never embedded in the sibling of another x_i leaf is crucial here. This allows us to sum over all l_i leaves without counting a removed leaf more than once.

Substituting this bound into the equation above yields:

$$E[L(f_\rho)] \leq L(f) - \tfrac{1}{2n}\sum_{i=1}^{n} 3l_i = (1 - \tfrac{3}{2n})L(f).$$

■

Note that using these methods, this bound is tight. In the worst case, every leaf has another leaf for a sibling, resulting in exactly 3 leaves removed per assigned leaf.

9.4 Exercises

Exercise 7: (Weighted Formula Complexity)

Suppose that inputs $x_1, \ldots, x_n$ have costs associated with them, $c_1, \ldots, c_n$. We can define formula cost in a natural way to be the sum of the weights of all inputs.

What is the *weighted* formula complexity of the function XOR_n over U_2?

(Note that if all the weights are 1, we just have the usual n^2 solution. Furthermore, over B_2, we can achieve the optimal formula cost of $\sum c_i$.)

CS277: Concrete Complexity **Fall 1996**

Lecture 10: September 26, 1996

Lecturer: Uri Zwick *Scribe: Felix Wu*

10.5 Lower Bound of Subbotovskaya

In the previous section, we proved the following two lemmas:

Lemma 10.6 *If $f : \{0,1\}^n \to \{0,1\}$ and $L(f) > 1$, then*

$$E_{\rho \in R^n_{n-1}}[L(f_\rho)] \leq (1 - \tfrac{1.5}{n})L(f).$$

Lemma 10.7 *If $f : \{0,1\}^n \to \{0,1\}$ and $L(f) = 1$, then*

$$E_{\rho \in R^n_{n-1}}[L(f_\rho)] = (1 - \tfrac{1}{n})L(f).$$

(Note, by the way, that these functions f depend formally on n variables, but need not actually depend on all n variables.)

Using these lemmas (or variants of them, as we shall see), we will now prove the following theorem.

Theorem 10.8 *If $f : \{0,1\}^n \to \{0,1\}$, then*

$$E_{\rho \in R^n_{\epsilon n}}[L(f_\rho)] \leq \epsilon^{1.5} \cdot L(f) + \tfrac{1}{3}.$$

Proof: Construct a random sequence of functions $f_n = f, f_{n-1}, \ldots, f_m$ $(m = \epsilon n)$, where f_i is obtained by assigning a random value to a randomly chosen variable of f_{i+1}. (Once again, this is a *formal* variable of f_{i+1}. f_{i+1} may not actually depend on this variable.)

We want to show that

$$E[L(f_m)] \leq (\tfrac{m}{n})^{1.5} \cdot L(f) + \tfrac{1}{3}.$$

If we could show that

$$E[L(f_k)] \leq (1 - \tfrac{1.5}{k+1})E[L(f_{k+1})],$$

then we would have

$$E[L(f_m)] \leq \left[\prod_{k=m+1}^{n} (1 - \tfrac{1.5}{k})\right] \cdot L(f).$$

From there, we could use the fact that

$$\prod_{k=m+1}^{n} (1 - \tfrac{1.5}{k}) \leq (\tfrac{m}{n})^{1.5}$$

to give the desired result.

(This last fact can be shown quite readily by induction. For a quick plausibility argument, note that $1 - \frac{1.5}{k} \approx e^{-1.5/k}$, and hence $\prod(1 - \frac{1.5}{k}) \approx e^{-1.5\sum 1/k} \approx e^{-1.5\log(n/m)} = (\frac{m}{n})^{1.5}$.)

However, the problem is that the equation above is derived from Lemma 10.6, and the inequality no longer holds if any of the $L(f_{k+1}) = 1$, because then we would need to use Lemma 10.7. Instead, we use the following lemma (thereby introducing the odd $\frac{1}{3}$ constant).

Lemma 10.9 *If $f : \{0,1\}^n \to \{0,1\}$, then*

$$E_{\rho \in R^n_{n-1}}[L(f_\rho) - \tfrac{1}{3}] \leq (1 - \tfrac{1.5}{n})(L(f) - \tfrac{1}{3}).$$

Proof: If $L(f) > 1$, this follows directly from Lemma 10.6.

If $L(f) = 1$, both sides evaluate to $\frac{2}{3} - \frac{1}{n}$, and hence, the lemma holds.

If $L(f) = 0$, the LHS is $-\frac{1}{3}$, while the RHS is $-\frac{1}{3}$ times something smaller than 1. Hence, the lemma holds. ■

As a consequence of this new lemma, we get:

$$E[L(f_k) - \tfrac{1}{3}] \leq (1 - \tfrac{1.5}{k+1})E[L(f_{k+1}) - \tfrac{1}{3}],$$

which implies

$$E[L(f_m) - \tfrac{1}{3}] \leq \left[\prod_{k=m+1}^{n} (1 - \tfrac{1.5}{k})\right] \cdot (L(f) - \tfrac{1}{3}).$$

Using the same product bound as before now yields the statement of the theorem. ■

(Instead of using Lemma 10.9, we could have written the appropriate expectation as:

$$E[L(f_k)] \leq (1 - \tfrac{1.5}{k+1})E[L(f_{k+1})] + (\tfrac{0.5}{k+1})Pr[L(f_{k+1}) = 1],$$

where the second term gives the exact correction needed to compensate for intermediate formulae of size 1. Such an approach also leads to the same constant $\frac{1}{3}$ at the end.)

Corollary 10.10 $L_{U_2}(XOR_n) = \Omega(n^{1.5})$.

10.6 Lower Bound of Andreev

The following two theorems (one of which we have seen before) are due to Shannon.

Theorem 10.11 *For most* $f \in B_n$, $C_{B_2}(f) \geq \frac{2^n}{n}$.

Theorem 10.12 *For most* $f \in B_n$, $L_{B_2}(f) \geq \frac{2^n}{\log n}$.

The latter of these theorems will be used to construct the Andreev function.

10.6.1 The Andreev Function

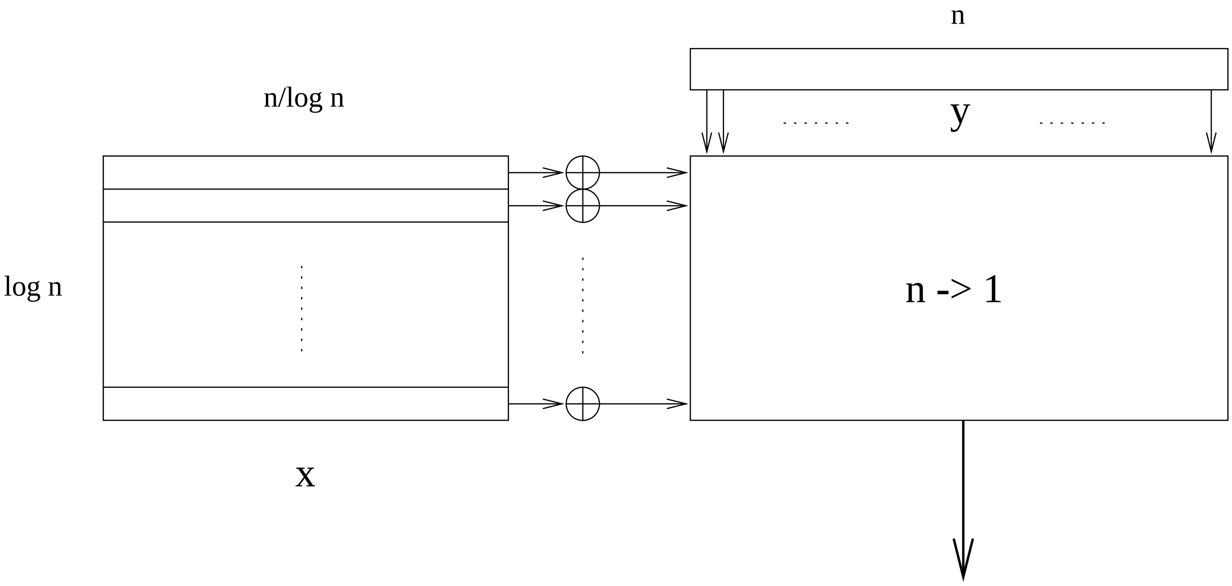

Figure 10.4: The Andreev Function

The function $An(x, y)$ takes as input two n-bit numbers, that is, $|x| = |y| = n$. The operation of the function is diagrammed in Figure 10.4. The bits of x are organized into $\log n$ rows of $n/\log n$ bits each. The function begins by computing the *XOR* of the bits in each row. It then uses these bits to index into the array y, returning the appropriate bit of y. Note that any given value of y can be thought of as a function of $\log n$ bits, and the selector can be thought of as performing a table lookup on this function.

From Theorem 10.12, we know that there exists a function f on $\log n$ variables whose formula complexity is at least $n/\log\log n$. Let y^* be the truth table for this function, and consider the function $An^*(x) = An(x, y^*)$. The formula complexity of An is at least that of An^*, since the latter is just a restriction of the former. Hence, by proving a bound for $L(An^*)$, we will have proven a bound for $L(An)$.

Note that $L(An) = O(n^3)$. We can compute the parities using $O(n^2)$ leaves. The selector only has $O(n)$ gates, but since this is a formula, we need to recompute the parities every time we need them. Hence, we get a total of $O(n^3)$.

10.6.2 Using the Andreev Function

Theorem 10.13 $L(An) = \Omega(n^{2.5-o(1)})$

Proof: As stated before, we will prove a bound for $L(An^*)$ in order to prove a bound for $L(An)$.

Restrict An^* with a random assignment that leaves only $m = 2\log n \log\log n$ variables free. When n is large enough, with probability $> 1/2$, there will be at least one free variable in each row.

Consider the probability that none of the variables in the first row remains unrestricted after the assignment. The probability that one of the free variables is not in the first row is $(1 - \frac{1}{\log n})$. Hence, the probability that none of them are in the first row is at most

$$(1 - \tfrac{1}{\log n})^m < 2^{-m/\log n} = \tfrac{1}{\log^2 n}.$$

Thus, the probability that any row has no free variables is

$$\leq \log n(\tfrac{1}{\log^2 n}) = \tfrac{1}{\log n} < \tfrac{1}{2} \qquad (n \geq 5).$$

Hence, with probability at least $1/2$, every row contains an unrestricted variable. This means that the new restricted function can still compute any index into y^*, and hence, the formula complexity of the restricted function is at least that of the "hard" function f that we began with. That is, we have:

$$\tfrac{1}{2}(n/\log\log n) \leq E_{\rho\in R^n_m}[L(An^*)].$$

But from Theorem 10.8, we have that:

$$E_{\rho\in R^n_m}[L(An^*)] \leq (\tfrac{m}{n})^{1.5} L(An^*) + \tfrac{1}{3}.$$

Hence,

$$L(An) \geq L(An^*) = \Omega(\frac{n^{2.5}}{\log^{1.5} n(\log\log n)^{2.5}}).$$

■

Notice the general proof methodology here. The idea is that we find a random assignment such that the resulting restriction is still fairly large, and we can use this to show that the original formula must have been even larger.

10.6.3 Further Improvements

Definition 10.14 *The shrinkage exponent Γ_Ω of a basis Ω is the supremum over all values γ such that $\forall f$,*

$$E_{\rho \in R^n_{\epsilon n}}[L(f_\rho)] = O(\epsilon^\gamma \cdot L(f) + 1).$$

From Theorem 10.8 above, we have shown that $\Gamma_{U_2} \geq 1.5$.

Theorem 10.13 can also be rewritten as:

Theorem 10.15 $L_{U_2}(An) = \Omega(n^{\Gamma_{U_2}+1-o(1)})$.

Furthermore, it has since been shown that:

Theorem 10.16 (Håstad) $\Gamma_{U_2} = 2$.

thereby implying the following:

Corollary 10.17 $L_{U_2}(An) = \Omega(n^{3-o(1)})$,

which as we saw above is basically tight.

10.7 Summary of Results

Nečiporuk	$\Omega(\frac{n^2}{\log n})$	any basis		1966
Khrapchenko	$\Omega(n^2)$	U_2	XOR_n, MAJ_n	1970
Subbotovskaya	$\Omega(n^{1.5})$	U_2	XOR_n	1961
Andreev	$\Omega(n^{2.5-o(1)})$	U_2	An	
Håstad	$\Omega(n^{3-o(1)})$	U_2	An	

10.8 Monotone Formulae for Majority: Preview

Over any complete basis Ω, we have the following:

$D_\Omega(MAJ_n) = O(\log n)$

$L_\Omega(MAJ_n) = n^{O(1)}$.

However, since MAJ_n is monotone, it is computable from the incomplete monotone basis $\{\wedge, \vee\}$. What then is $L_{\{\wedge,\vee\}}(MAJ_n)$?

10.9 Exercises

Exercise 8: Show that $\Gamma_{U_2} \leq 2$.

Exercise 9: Let $\gamma < \Gamma_\Omega$. Show that:

1. $L_\Omega(XOR_n) = \Omega(n^\gamma)$
2. $L_\Omega(An) = \Omega(n^{\gamma+1})$
3. $L_\Omega(MAJ_n) = \Omega(n^\gamma)$

Exercise 10: Obtain a lower bound on $L_{B_2}(An)$.

Exercise 11: Obtain a monotone formula of size $n^{O(\log n)}$ for the majority function.