On Formula Lower bounds

Pulkit Sinha
Omkar Bhalchandra Baraskar

Indian Institute of Science

December 2020

Krapchenko's Method

Krapchenko's Method

Krapchenko's Method

Theorem 1 (Krapchenko 1971)

Any formula computing $\mathbf{PARITY}(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Krapchenko's Method

Theorem 1 (Krapchenko 1971)

Any formula computing **PARITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

- Krapchenko used "formal complexity measures" to prove the above theorem.

Krapchenko's Method

Theorem 1 (Krapchenko 1971)

Any formula computing **PARITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

- Krapchenko used "formal complexity measures" to prove the above theorem.
- Amazingly a very similar proof strategy helps us prove that formula computing MAJORITY$(x_1, \ldots, x_n)$ requires $\Omega(n^2)$ size.

Formal Complexity Measures

Formal Complexity Measures

Definition 1

A formal complexity measure (FC) is a function from the space of boolean function taking inputs on $\{0,1\}^n$ to natural number satisfying the following properties

1. $FC(x_i) = 1$ for all input variables x_i
2. $FC(f) = FC(\neg f)$
3. $FC(f \vee g) \leq FC(f) + FC(g)$ or
4. $FC(f \wedge g) \leq FC(f) + FC(g)$

Formal Complexity Measures

Definition 1

A formal complexity measure (FC) is a function from the space of boolean function taking inputs on $\{0,1\}^n$ to natural number satisfying the following properties

1. $FC(x_i) = 1$ for all input variables x_i
2. $FC(f) = FC(\neg f)$
3. $FC(f \vee g) \leq FC(f) + FC(g)$ or
4. $FC(f \wedge g) \leq FC(f) + FC(g)$

- By application of De-Morgan's law it is easy to see that properties $2,3$ imply property 4

Formal Complexity Measures

Definition 1

A formal complexity measure (FC) is a function from the space of boolean function taking inputs on $\{0,1\}^n$ to natural number satisfying the following properties

1. $FC(x_i) = 1$ for all input variables x_i
2. $FC(f) = FC(\neg f)$
3. $FC(f \vee g) \leq FC(f) + FC(g)$ or
4. $FC(f \wedge g) \leq FC(f) + FC(g)$

- By application of De-Morgan's law it is easy to see that properties $2, 3$ imply property 4
- From the above definition it is not immediately clear how FC will help us get lower bounds for PARITY and MAJORITY. The next lemma will give us a explicit relation between FC and formula complexity of a function.

Relation between Formal Complexity Measures and Formula Complexity

Relation between Formal Complexity Measures and Formula Complexity

Lemma 1

Any complexity measure FC satisfies

$$FC(f) \leq L(f)$$

where $L(.)$ be the formula complexity of the function i.e the size of the smallest formula in the de Morgan's basis for the function.

Relation between Formal Complexity Measures and Formula Complexity

Lemma 1

Any complexity measure FC satisfies

$$FC(f) \leq L(f)$$

where $L(.)$ be the formula complexity of the function i.e the size of the smallest formula in the de Morgan's basis for the function.

Proof.

- We will prove this by induction on $L(f)$.
- The base case $l(f) = 1$ follows directly from property 1.

Relation between Formal Complexity Measures and Formula Complexity

Proof (Cont.)

- **Inductive Step:** Let f be the formula whose formula complexity is under consideration in this step. Now we have two cases to deal with,
  1. If $f = \neg f_1$.

$$
\begin{aligned}
L(f) &= L(f_1) + 1 \\
L(f) &\geq FC(f_1) + 1 \\
&= FC(\neg f_1) + 1 \\
&= FC(f) + 1 \geq FC(f)
\end{aligned}
$$

Relation between Formal Complexity Measures and Formula Complexity

Proof (Cont.)

- **Inductive Step:** Let f be the formula whose formula complexity is under consideration in this step. Now we have two cases to deal with,

2. If $f = f_1 \vee f_2$.

$$\begin{aligned} L(f) &= L(f_1) + L(f_2) + 1 \\ L(f) &\geq FC(f_1) + FC(f_2) + 1 \\ &\geq FC(f_1 \vee f_2) + 1 \\ &= FC(f) + 1 \geq FC(f) \end{aligned}$$

The case of $f = f_1 \wedge f_2$ follows in the same manner. □

Krapchenko's Measure

$$\mathcal{F}(f) = \max_{\substack{A \subseteq f^{-1}(0) \\ B \subseteq f^{-1}(1)}} \frac{e(A, B)^2}{|A||B|}$$

Krapchenko's Measure

$$\mathcal{F}(f) = \max_{\substack{A \subseteq f^{-1}(0) \\ B \subseteq f^{-1}(1)}} \frac{e(A, B)^2}{|A||B|}$$

- $\mathcal{F}$ is the function from boolean function which take input in $\{0, 1\}^n$ and outputs a real number.

Krapchenko's Measure

$$\mathcal{F}(f) = \max_{\substack{A\subseteq f^{-1}(0)\\ B\subseteq f^{-1}(1)}} \frac{e(A,B)^2}{|A||B|}$$

- $\mathcal{F}$ is the function from boolean function which take input in $\{0,1\}^n$ and outputs a real number.
- $e(A,B)$ is the number of pairs $(x,y) \in A \times B$ such that x and y differ in exactly one coordinate.
- $\mathcal{F}$ is a complexity measure.

Krapchenko's Measure

$$\mathcal{F}(f) = \max_{\substack{A \subseteq f^{-1}(0) \\ B \subseteq f^{-1}(1)}} \frac{e(A,B)^2}{|A||B|}$$

- $\mathcal{F}$ is the function from boolean function which take input in $\{0,1\}^n$ and outputs a real number.
- $e(A,B)$ is the number of pairs $(x,y) \in A \times B$ such that x and y differ in exactly one coordinate.
- $\mathcal{F}$ is a complexity measure.Before we prove that it is indeed a complexity measure we see how it helps us get lower bounds for PARITY and MAJORITY.

Lower Bound for PARITY in De-Morgan's basis

Lower Bound for PARITY in De-Morgan's basis

Lemma 2

$\mathcal{F}(PARITY) \geq n^2$ where $\mathcal{F}$ is Krapchenko's measure.

Lower Bound for PARITY in De-Morgan's basis

Lemma 2

$\mathcal{F}(PARITY) \geq n^2$ where $\mathcal{F}$ is Krapchenko's measure.

Proof.

- To prove this it is sufficient to show that there exists $A \subseteq \text{PARITY}^{-1}(0)$, $B \subseteq \text{PARITY}^{-1}(1)$ and, $\frac{e(A,B)^2}{|A||B|} \geq n^2$.

Lower Bound for PARITY in De-Morgan's basis

Proof (Cont.)

- We take $A = \text{PARITY}^{-1}(0)$ and $B = \text{PARITY}^{-1}(1)$. Now $|A| = 2^{n-1}$ and $|B| = 2^{n-1}$.

Lower Bound for PARITY in De-Morgan's basis

Proof (Cont.)

- We take $A = \text{PARITY}^{-1}(0)$ and $B = \text{PARITY}^{-1}(1)$. Now $|A| = 2^{n-1}$ and $|B| = 2^{n-1}$.
- **Observation** : Let $PARITY(x) = 0$, now let y be obtained after changing one bit x then $PARITY(y) = 1$.

Proof (Cont.)

- We take $A = \text{PARITY}^{-1}(0)$ and $B = \text{PARITY}^{-1}(1)$. Now $|A| = 2^{n-1}$ and $|B| = 2^{n-1}$.
- **Observation** : Let $PARITY(x) = 0$, now let y be obtained after changing one bit x then $PARITY(y) = 1$.
- The above observation suggests that for every $x \in A$ there are exactly n elements in B which differ from it by a bit.

Lower Bound for PARITY in De-Morgan's basis

Proof (Cont.)

- We take $A = \text{PARITY}^{-1}(0)$ and $B = \text{PARITY}^{-1}(1)$. Now $|A| = 2^{n-1}$ and $|B| = 2^{n-1}$.
- **Observation** : Let $PARITY(x) = 0$, now let y be obtained after changing one bit x then $PARITY(y) = 1$.
- The above observation suggests that for every $x \in A$ there are exactly n elements in B which differ from it by a bit. Thus $e(A, B) = n|A| = n2^{n-1}$.
- Now for this choice of A, B we have $\frac{e(A,B)^2}{|A||B|} = \frac{n^2 2^n}{(2^{n-1})(2^{n-1})} = n^2$ $\square$

Lower Bound for PARITY in De-Morgan's basis

Theorem 1 (Krapchenko 1971)

Any formula computing **PARITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Lower Bound for PARITY in De-Morgan's basis

Theorem 1 (Krapchenko 1971)

Any formula computing **PARITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Proof.

- Lemma 1 says for every "appropriate" boolean function,formal complexity measure $L(f) \geq FC(f)$ holds, where $L(.)$ is the formula complexity.
- Lemma 2 tells us $\mathcal{F}(PARITY) \geq n^2$, where $\mathcal{F}$ is Krapchenko's measure.
- Combining lemmma 1 and 2 we get $L(PARITY) = \Omega(n^2)$.

$\square$

Lower Bound for MAJORITY in De-Morgan's basis

Lower Bound for MAJORITY in De-Morgan's basis

Theorem 2

Any formula computing **MAJORITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Lower Bound for MAJORITY in De-Morgan's basis

Theorem 2

Any formula computing $\textbf{MAJORITY}(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

- Similar to the case of PARITY we will lower bound the Krapchenko's measure for MAJORITY and then use the lemma 1 to give the lower bound for formula complexity.

Lower Bound for MAJORITY in De-Morgan's basis

Lemma 3

$\mathcal{F}(MAJORITY) \geq \Omega(n^2)$ where $\mathcal{F}$ is Krapchenko's measure.

Lower Bound for MAJORITY in De-Morgan's basis

Lemma 3

$\mathcal{F}(MAJORITY) \geq \Omega(n^2)$ where $\mathcal{F}$ is Krapchenko's measure.

Proof.

- Now similar to prove of PARITY we will give $A \subseteq \text{MAJORITY}^{-1}(0)$ and $B \subseteq \text{MAJORITY}^{-1}(1)$ s.t $\frac{e(A,B^2}{|A||B|} = \Omega(n^2)$.

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1.

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.
- **Observation:**
  1. $A \subseteq MAJORITY^{-1}(0)$ and $B \subseteq MAJORITY^{-1}(1)$.

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.
- **Observation:**
  1. $A \subseteq MAJORITY^{-1}(0)$ and $B \subseteq MAJORITY^{-1}(1)$.
  2. Let $x \in A$ then $MAJORITY(x) = 0$, now let y be obtained after changing one bit x. If the bit being changed was set to 0 then $MAJORITY(y) = 1$.

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.
- **Observation:**
  1. $A \subseteq MAJORITY^{-1}(0)$ and $B \subseteq MAJORITY^{-1}(1)$.
  2. Let $x \in A$ then $MAJORITY(x) = 0$, now let y be obtained after changing one bit x. If the bit being changed was set to 0 then $MAJORITY(y) = 1$. If the bit being changed was set to 1 then $MAJORITY(y) = 0$ (i.e it remains unchanged).

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.
- **Observation:**
  1. $A \subseteq MAJORITY^{-1}(0)$ and $B \subseteq MAJORITY^{-1}(1)$.
  2. Let $x \in A$ then $MAJORITY(x) = 0$, now let y be obtained after changing one bit x. If the bit being changed was set to 0 then $MAJORITY(y) = 1$. If the bit being changed was set to 1 then $MAJORITY(y) = 0$ (i.e it remains unchanged).
- The above observation suggests that for every $x \in A$ there are exactly $\lfloor \frac{n}{2} \rfloor + 1$ elements in B which differ from it by a bit.

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- Let A be the set of number where $\lfloor \frac{n}{2} \rfloor$ number of bits are set to 1 and B be the set of numbers where $\lfloor \frac{n}{2} \rfloor + 1$ bits are set to 1. Now $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.
- **Observation:**
  1. $A \subseteq MAJORITY^{-1}(0)$ and $B \subseteq MAJORITY^{-1}(1)$.
  2. Let $x \in A$ then $MAJORITY(x) = 0$, now let y be obtained after changing one bit x. If the bit being changed was set to 0 then $MAJORITY(y) = 1$. If the bit being changed was set to 1 then $MAJORITY(y) = 0$ (i.e it remains unchanged).
- The above observation suggests that for every $x \in A$ there are exactly $\lfloor \frac{n}{2} \rfloor + 1$ elements in B which differ from it by a bit. Thus $e(A, B) = (\lfloor \frac{n}{2} \rfloor + 1)|A| = (\lfloor \frac{n}{2} \rfloor + 1)\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$.

Proof (Cont.)

- We now have $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$ and $e(A, B) = (\lfloor \frac{n}{2} \rfloor + 1)\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- We now have $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$ and $e(A, B) = (\lfloor \frac{n}{2} \rfloor + 1)\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$
- Thus

$$\frac{e(A,B)^2}{|A||B|} = \frac{(\lfloor \frac{n}{2} \rfloor + 1)^2 \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}^2}{\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}} = (\lfloor \frac{n}{2} \rfloor + 1)^2$$

Lower Bound for MAJORITY in De-Morgan's basis

Proof (Cont.)

- We now have $|A| = |B| = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$ and $e(A, B) = (\lfloor \frac{n}{2} \rfloor + 1)\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}$
- Thus

$$\frac{e(A, B)^2}{|A||B|} = \frac{(\lfloor \frac{n}{2} \rfloor + 1)^2 \binom{n}{\lfloor \frac{n}{2} \rfloor + 1}^2}{\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}} = (\lfloor \frac{n}{2} \rfloor + 1)^2 = \theta(n^2)$$

$\square$

Lower Bound for MAJORITY in De-Morgan's basis

Theorem 2

Any formula computing **MAJORITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Lower Bound for MAJORITY in De-Morgan's basis

Theorem 2

Any formula computing **MAJORITY**$(x_1, x_2, \ldots, x_n)$ has size $\Omega(n^2)$ in de-Morgan basis.

Proof.

- Lemma 1 says for every "appropriate" boolean function,formal complexity measure $L(f) \geq FC(f)$ holds, where $L(.)$ is the formula complexity.
- Lemma 3 tells us $\mathcal{F}(MAJORITY) \geq \Omega(n^2)$, where $\mathcal{F}$ is Krapchenko's measure.
- Combining lemmma 1 and 3 we get $L(MAJORITY) = \Omega(n^2)$.

□

Krapchenko's Measure is Formal Complexity Measure

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof.

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof.

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - **Observation:** For every element x in $f^{-1}(0)$ there is exactly one element y in $f^{-1}(1)$ which differs from x exactly at one bit position, which is the ith bit.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof.

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - **Observation:** For every element x in $f^{-1}(0)$ there is exactly one element y in $f^{-1}(1)$ which differs from x exactly at one bit position, which is the ith bit.
   - From the above observation $\frac{e(A,B)}{|A|} \leq 1$ where $A \subseteq f^{-1}(0)$ $B \subseteq f^{-1}(1)$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof.

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - **Observation:** For every element x in $f^{-1}(0)$ there is exactly one element y in $f^{-1}(1)$ which differs from x exactly at one bit position, which is the ith bit.
   - From the above observation $\frac{e(A,B)}{|A|} \leq 1$ where $A \subseteq f^{-1}(0)$ $B \subseteq f^{-1}(1)$. Similarly we can prove $\frac{e(A,B)}{|B|} \leq 1$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof.

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - **Observation:** For every element x in $f^{-1}(0)$ there is exactly one element y in $f^{-1}(1)$ which differs from x exactly at one bit position, which is the ith bit.
   - From the above observation $\frac{e(A,B)}{|A|} \leq 1$ where $A \subseteq f^{-1}(0)$ $B \subseteq f^{-1}(1)$. Similarly we can prove $\frac{e(A,B)}{|B|} \leq 1$.Multiplying these two we have $\frac{e(A,B)^2}{|A||B|} \leq 1 \implies \mathcal{F}(x_i) \leq 1$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof (Cont.)

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - We now have $\mathcal{F}(x_i) \leq 1$

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof (Cont.)

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - We now have $\mathcal{F}(x_i) \leq 1$
   - Take $A = \{0^n\}$ and $B = \{0^{i-1}10^{n-i}\}$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof (Cont.)

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - We now have $\mathcal{F}(x_i) \leq 1$
   - Take $A = \{0^n\}$ and $B = \{0^{i-1}10^{n-i}\}$.Now we have $A \subseteq f^{-1}(0)$ $B \subseteq f^{-1}(1)$ and $e(A, B) = |A| = |B| = 1$. Thus $\mathcal{F}(x_i) \geq 1$.

Krapchenko's Measure is Formal Complexity Measure

Lemma 4

Krapchenko's measure is a formal complexity measure.

Proof (Cont.)

1. $\mathcal{F}(x_i) = 1$, for all input variables x_i :
   - For the given function $f(x) = x_i$, $f^{-1}(0) = \{x \mid x_i = 0\}$ and $f^{-1}(1) = \{x \mid x_i = 1\}$.
   - We now have $\mathcal{F}(x_i) \leq 1$
   - Take $A = \{0^n\}$ and $B = \{0^{i-1}10^{n-i}\}$.Now we have $A \subseteq f^{-1}(0)$ $B \subseteq f^{-1}(1)$ and $e(A, B) = |A| = |B| = 1$. Thus $\mathcal{F}(x_i) \geq 1$.
   - Combining these two we have $\mathcal{F}(x_i) = 1$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

2. $\mathcal{F}(f) = \mathcal{F}(\neg f)$ for all boolean functions f:
   The proof follows from the fact that the definition of $\mathcal{F}$ is symmteric with respect to $A \subseteq f^{-1}(0)$ and $B \subseteq f^{-1}(1)$

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
   - To prove this, it is enough to prove that for every pair of sets $A \subseteq h^{-1}(0)$ and $B \subseteq h^{-1}(1)$, there exists sets $A_f \subseteq f^{-1}(0), B_f \subseteq f^{-1}(1), A_g \subseteq g^{-1}(0), B_g \subseteq g^{-1}(1)$ which satisfy

$$\frac{e(A,B)^2}{|A||B|} \leq \frac{e(A_f,B_f)^2}{|A_f||B_f|} + \frac{e(A_g,B_g)^2}{|A_g||B_g|}$$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
   - To prove this, it is enough to prove that for every pair of sets $A \subseteq h^{-1}(0)$ and $B \subseteq h^{-1}(1)$, there exists sets $A_f \subseteq f^{-1}(0), B_f \subseteq f^{-1}(1), A_g \subseteq g^{-1}(0), B_g \subseteq g^{-1}(1)$ which satisfy

$$\frac{e(A,B)^2}{|A||B|} \leq \frac{e(A_f,B_f)^2}{|A_f||B_f|} + \frac{e(A_g,B_g)^2}{|A_g||B_g|}$$

   - Now we fix A, B and we will show the existence of sets A_f, A_g, B_f, B_g as described above.

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
   - Now observe that $h^{-1}(0) = f^{-1}(0) \wedge g^{-1}(0)$ and $h^{-1}(1) = f^{-1}(1) \vee g^{-1}(1)$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
   - Now observe that $h^{-1}(0) = f^{-1}(0) \wedge g^{-1}(0)$ and $h^{-1}(1) = f^{-1}(1) \vee g^{-1}(1)$
   - We take $A_f = A_g = A$, $B_f = B \cap f^{-1}(1)$, $B_g = B \setminus B_f$.

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
   - Now observe that $h^{-1}(0) = f^{-1}(0) \wedge g^{-1}(0)$ and $h^{-1}(1) = f^{-1}(1) \vee g^{-1}(1)$
   - We take $A_f = A_g = A$, $B_f = B \cap f^{-1}(1)$, $B_g = B \setminus B_f$.Observe that each of the above sets is appropriately defined.

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
    - Now observe that $h^{-1}(0) = f^{-1}(0) \wedge g^{-1}(0)$ and $h^{-1}(1) = f^{-1}(1) \vee g^{-1}(1)$
    - We take $A_f = A_g = A$, $B_f = B \cap f^{-1}(1)$, $B_g = B \setminus B_f$.Observe that each of the above sets is appropriately defined.
    - Now as B_f and B_g partition B, thus $e(A, B) = e(A, B_f) + e(A, B_g) = e(A_f, B_f) + e(A_g, B_g)$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g:
Now we have,

$$\begin{aligned}\frac{e(A,B)^2}{|A||B|} &= \frac{1}{|A|} \cdot \frac{(e(A_f, B_f) + e(A_g, B_g))^2}{|B_f| + |B_g|} \\ &\leq \frac{1}{|A|} \cdot \frac{e(A_f, B_f)^2}{|B_f|} + \frac{e(A_g, B_g)^2}{|B_g|} \\ &\left[\frac{(a+b)^2}{c+d} \leq \frac{a^2}{c} + \frac{b^2}{d} \right] [2]\end{aligned}$$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g: Now we have,

$$\begin{aligned}
\frac{e(A,B)^2}{|A||B|} &= \frac{1}{|A|} \cdot \frac{(e(A_f, B_f) + e(A_g, B_g))^2}{|B_f| + |B_g|} \\
&\leq \frac{1}{|A|} \cdot \frac{e(A_f, B_f)^2}{|B_f|} + \frac{e(A_g, B_g)^2}{|B_g|} \\
&\leq \frac{e(A_f, B_f)^2}{|A||B_f|} + \frac{e(A_g, B_g)^2}{|A||B_g|}
\end{aligned}$$

Krapchenko's Measure is Formal Complexity Measure

Proof (Cont.)

3. $\mathcal{F}(h) \leq \mathcal{F}(f) + \mathcal{F}(g)$ where $h = f \vee g$, for all boolean functions f, g: Now we have,

$$\begin{aligned}
\frac{e(A,B)^2}{|A||B|} &= \frac{1}{|A|} \cdot \frac{(e(A_f,B_f) + e(A_g,B_g))^2}{|B_f| + |B_g|} \\
&\leq \frac{1}{|A|} \cdot \frac{e(A_f,B_f)^2}{|B_f|} + \frac{e(A_g,B_g)^2}{|B_g|} \\
&\leq \frac{e(A_f,B_f)^2}{|A_f||B_f|} + \frac{e(A_g,B_g)^2}{|A_g||B_g|}
\end{aligned}$$

□

Can we do better?

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the krapchenko's method ?

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the krapchenko's method ?
- At first glance the question seems a bit incomplete and it looks like the answer should depend on the boolean function we are computing.

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the krapchenko's method ?
- At first glance the question seems a bit incomplete and it looks like the answer should depend on the boolean function we are computing.
- But it turns out that Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$.

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the krapchenko's method ?
- At first glance the question seems a bit incomplete and it looks like the answer should depend on the boolean function we are computing.
- But it turns out that Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$.
- Before we prove the above statement let's state by what we mean by "lower bound given by Krapchenko's Method". In above proves for parity and majority the pivotal tool used was Krapchenko's measure. So when we say "lower bound given by Krapchenko's method" we actually mean lower bound given by using the Krapchenko's measure. Thus the question boils down to asking what is the maximum value of Krapchenko's measure ?

Can we do better?

Lemma 5

Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$

Can we do better?

Lemma 5

Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$

Proof.

- As discussed before it suffices to proving that Krapchenko's measure cannot attain values greater tha n^2 i.e
- Let f be a boolean function on n variables.Now we have

$$\mathcal{F}(f) = \max_{\substack{A \subseteq f^{-1}(0) \\ B \subseteq f^{-1}(1)}} \frac{e(A, B)^2}{|A||B|}$$

$$\leq \max_{\substack{A \subseteq \{0,1\}^n \\ B \subseteq \{0,1\}^n}} \frac{e(A, B)^2}{|A||B|}$$

Can we do better?

Lemma 5

Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$

Proof (Cont.)

- We now observe that $e(A, B) \leq n \min(|A|, |B|)$.

Can we do better?

Lemma 5

Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$

Proof (Cont.)

- We now observe that $e(A, B) \leq n \min(|A|, |B|)$.
- Now we get

$$\begin{aligned}
\mathcal{F}(f) &\leq \max_{\substack{A\subseteq\{0,1\}^n \\ B\subseteq\{0,1\}^n}} \frac{e(A, B)^2}{|A||B|} \\
&\leq \max_{\substack{A\subseteq\{0,1\}^n \\ B\subseteq\{0,1\}^n}} \frac{n^2 min(|A|, |B|)^2}{|A||B|}
\end{aligned}$$

Can we do better?

Lemma 5

Krapchenko's method cannot provide us with a lower bound better than $\Omega(n^2)$

Proof (Cont.)

$$
\begin{aligned}
\mathcal{F}(f) &\leq \max_{\substack{A \subseteq \{0,1\}^n \\ B \subseteq \{0,1\}^n}} \frac{n^2 min(|A|,|B|)^2}{|A||B|} \\
&\leq \max_{\substack{A \subseteq \{0,1\}^n \\ B \subseteq \{0,1\}^n}} n^2 \cdot \frac{min(|A|,|B|)^2}{|A||B|} \\
&\leq n^2
\end{aligned}
$$

□

Subbotovskaya's Method

Subbotovskaya's Method

- Gives a slightly weaker formula lower bound for $PARITY$: $\Omega(n^{1.5})$ using a new method involving random restrictions

Subbotovskaya's Method

- Gives a slightly weaker formula lower bound for *PARITY*: $\Omega(n^{1.5})$ using a new method involving random restrictions
- Later used by Andreev to obtain a formula lower bound for a different function.

Subbotovskaya's Method

- Gives a slightly weaker formula lower bound for *PARITY*: $\Omega(n^{1.5})$ using a new method involving random restrictions
- Later used by Andreev to obtain a formula lower bound for a different function.
- Main idea:

  For any function f, if we are given formula for f, we can obtain an upper bound(in terms of the original formula size) on the optimal formula size for any restriction f'. When accompanied with a lower bound on formula for f', we get a lower bound for the original formula.

- Goal:
  We want to maximize the "gap" between the formula for f and f', i.e, we want the reduction in size of the formula to be as large as possible, as this will, in the reverse direction, give a larger lower bound for the formula for f.

Subbotovskaya's Method

- Goal:
  We want to maximize the "gap" between the formula for f and f', i.e, we want the reduction in size of the formula to be as large as possible, as this will, in the reverse direction, give a larger lower bound for the formula for f.
- Note that it is okay to obtain this gap for only some restrictions, as long as we can guarantee the existence of the corresponding restriction.

- Goal:
  We want to maximize the "gap" between the formula for f and f', i.e, we want the reduction in size of the formula to be as large as possible, as this will, in the reverse direction, give a larger lower bound for the formula for f.
- Note that it is okay to obtain this gap for only some restrictions, as long as we can guarantee the existence of the corresponding restriction.
- We will use the probabilistic method for this.

Subbotovskaya's Method

Definition 1

A boolean formula F is said to *Nice* if the following holds:

Subbotovskaya's Method

Definition 1

A boolean formula F is said to *Nice* if the following holds:

1. There are no $\neg$ gates in F, except possibly just after the inputs/leaf nodes.

Subbotovskaya's Method

Definition 1

A boolean formula F is said to *Nice* if the following holds:

1. There are no $\neg$ gates in F, except possibly just after the inputs/leaf nodes.
2. For any $\wedge$ or $\vee$ gate in F such that one of its input is some literal x_i(either some input directly or the $\neg$ of it), then the other input is a formula whose output does not depend on x_i.

Subbotovskaya's Method

Definition 1

A boolean formula F is said to *Nice* if the following holds:

1. There are no $\neg$ gates in F, except possibly just after the inputs/leaf nodes.
2. For any $\wedge$ or $\vee$ gate in F such that one of its input is some literal x_i(either some input directly or the $\neg$ of it), then the other input is a formula whose output does not depend on x_i.
3. All leaf nodes are some input variables OR F outputs a constant function, in which case F must have no gates.

Subbotovskaya's Method

Definition 1

A boolean formula F is said to *Nice* if the following holds:

1. There are no $\neg$ gates in F, except possibly just after the inputs/leaf nodes.
2. For any $\wedge$ or $\vee$ gate in F such that one of its input is some literal x_i(either some input directly or the $\neg$ of it), then the other input is a formula whose output does not depend on x_i.
3. All leaf nodes are some input variables OR F outputs a constant function, in which case F must have no gates.

Observe that every boolean function f has a corresponding nice formula F which has size atmost constant times the the size of the optimal formula for f.

Subbotovskaya's Method

Observe that every function f has a corresponding nice formula F which has size atmost constant times the the size of the optimal formula for f.

- Condition 1. can be made to be satisfied trivially.

Subbotovskaya's Method

Observe that every function f has a corresponding nice formula F which has size atmost constant times the the size of the optimal formula for f.

- Condition 1. can be made to be satisfied trivially.
- For Condition 2.

$$h(x_1, x_2, x_3 \ldots) \wedge x_1 = h(1, x_2, x_3 \ldots) \wedge x_1$$

Observe that every function f has a corresponding nice formula F which has size atmost constant times the the size of the optimal formula for f.

- Condition 1. can be made to be satisfied trivially.
- For Condition 2.

$$h(x_1, x_2, x_3 \dots) \wedge x_1 = h(1, x_2, x_3 \dots) \wedge x_1$$

Similarly, for $\vee$

$$h(x_1, x_2, x_3 \dots) \vee x_1 = h(0, x_2, x_3 \dots) \vee x_1$$

Subbotovskaya's Method

$|F| :=$ Number of literals in F

Subbotovskaya's Method

$|F| :=$ Number of literals in F
Note that this can go to 0 for the case of F not taking any inputs.

Subbotovskaya's Method

$|F| :=$ Number of literals in F
Note that this can go to 0 for the case of F not taking any inputs.

Definition 2

We denote by f_π the function obtained by restricting the input of f by the restriction π

Subbotovskaya's Method

$|F| :=$ Number of literals in F
Note that this can go to 0 for the case of F not taking any inputs.

Definition 2

We denote by f_π the function obtained by restricting the input of f by the restriction π

Lemma 1

Suppose given a boolean function f taking $n \geq 2$ inputs $x_1 \ldots x_n$. Let π be a restriction chosen uniformly randomly from the set of restrictions setting the value of exactly one x_i. Let F denote some optimal nice formula for f, and F_π some optimal nice formula for f_π. Then,

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

Proof of Lemma 1

Lemma 1

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

Proof of Lemma 1

Lemma 1

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

For every restriction π of the given kind, we construct a formula F'_π for f_π which is nice, such that over the uniform distribution considered,

$$\mathbb{E}_\pi[|F'_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

Proof of Lemma 1

Lemma 1

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

For every restriction π of the given kind, we construct a formula F'_π for f_π which is nice, such that over the uniform distribution considered,

$$\mathbb{E}_\pi[|F'_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

Since $|F'_\pi| \geq |F_\pi|$ for each π due to the optimality of F_π, it suffices to find such an F'_π for each π, as $\mathbb{E}[|F'_\pi|]_\pi \geq E[|F_\pi|]$

Proof of Lemma 1

Let π be the restriction which sets x_i to j, for $i \in [n]$ and $j \in \{0, 1\}$

Proof of Lemma 1

Let π be the restriction which sets x_i to j, for $i \in [n]$ and $j \in \{0, 1\}$

Then, to obtain F'_π, we simply assign the values of the inputs in F, and lazily evaluate gates as much as we can.

Proof of Lemma 1

Let π be the restriction which sets x_i to j, for $i \in [n]$ and $j \in \{0,1\}$

Then, to obtain F'_π, we simply assign the values of the inputs in F, and lazily evaluate gates as much as we can.

$$(h \wedge 1) \to h \qquad (h \wedge 0) \to 0$$

$$(h \vee 1) \to 1 \qquad (h \vee 0) \to h$$

$$\neg 0 \to 1 \qquad \neg 1 \to 0$$

Proof of Lemma 1

Let π be the restriction which sets x_i to j, for $i \in [n]$ and $j \in \{0, 1\}$

Then, to obtain F'_π, we simply assign the values of the inputs in F, and lazily evaluate gates as much as we can.

$$(h \wedge 1) \to h \qquad (h \wedge 0) \to 0$$

$$(h \vee 1) \to 1 \qquad (h \vee 0) \to h$$

$$\neg 0 \to 1 \qquad \neg 1 \to 0$$

Note that throughout the lazy evaluation, the conditions 1 and 2 in Definition 1 are satisfied, and the lazy evaluation terminates only when condition 3 is satisfied.

Proof of Lemma 1

Let π be the restriction which sets x_i to j, for $i \in [n]$ and $j \in \{0, 1\}$

Then, to obtain F'_π, we simply assign the values of the inputs in F, and lazily evaluate gates as much as we can.

$$(h \wedge 1) \to h \qquad (h \wedge 0) \to 0$$

$$(h \vee 1) \to 1 \qquad (h \vee 0) \to h$$

$$\neg 0 \to 1 \qquad \neg 1 \to 0$$

Note that throughout the lazy evaluation, the conditions 1 and 2 in Definition 1 are satisfied, and the lazy evaluation terminates only when condition 3 is satisfied.

$\implies$ F'_π obtained is nice.

Proof of Lemma 1

For Condition 2:
For any $\wedge$ gate at some stage of lazy evaluation, with one of the inputs x, which is a literal of F, it might either get removed, or

$$h \wedge x \quad \rightarrow \quad h' \wedge x$$

Proof of Lemma 1

Now, we show that, if s_i denotes the number of x_i and $\neg x_i$ literals in F with input x_i or $\neg x_i$, then

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

Proof of Lemma 1

Now, we show that, if s_i denotes the number of x_i and $\neg x_i$ literals in F with input x_i or $\neg x_i$, then

$$\mathbb{E}_{j \in \{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2} s_i$$

Essentially, $|F| - |F'_\pi|$ denotes the reduction in the number of literals when doing the lazy evaluation, so its enough to lower bound this.

Proof of Lemma 1

Now, we show that, if s_i denotes the number of x_i and $\neg x_i$ literals in F with input x_i or $\neg x_i$, then

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

Essentially, $|F| - |F'_\pi|$ denotes the reduction in the number of literals when doing the lazy evaluation, so its enough to lower bound this.
For a literal y of x_i, i.e, either x_i or $\neg x_i$, consider an $\wedge$ gate in F: $h \wedge y$

$$(h \wedge y) \to \begin{cases} h & \text{when } y \to 1 \\ 0 & \text{when } y \to 0 \end{cases}$$

Proof of Lemma 1

Now, we show that, if s_i denotes the number of x_i and $\neg x_i$ literals in F with input x_i or $\neg x_i$, then

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

Essentially, $|F| - |F'_\pi|$ denotes the reduction in the number of literals when doing the lazy evaluation, so its enough to lower bound this.
For a literal y of x_i, i.e, either x_i or $\neg x_i$, consider an $\wedge$ gate in F: $h \wedge y$

$$(h \wedge y) \rightarrow \begin{cases} h & \text{when } y \rightarrow 1 \\ 0 & \text{when } y \rightarrow 0 \end{cases}$$

$\implies$ Average/expected loss of literals $\geq \frac{3}{2}$ $\implies$ Overall expected loss of literals$\geq \frac{3}{2}s_i$

Proof of Lemma 1

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

Proof of Lemma 1

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

$$\implies \mathbb{E}_{i\in[n]}\left[E_{j\in\{0,1\}}[|F| - |F'_\pi|]\right] \geq E_{i\in[n]}\left[\frac{3}{2}s_i\right]$$

Proof of Lemma 1

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

$$\implies \mathbb{E}_{i\in[n]}\left[E_{j\in\{0,1\}}[|F| - |F'_\pi|]\right] \geq E_{i\in[n]}\left[\frac{3}{2}s_i\right]$$

$$\implies \mathbb{E}_\pi[|F| - |F'_\pi|]] \geq \frac{3}{2}\cdot\frac{\sum_{i=1}^n s_i}{n} = \frac{3|F|}{2n}$$

Proof of Lemma 1

$$\mathbb{E}_{j\in\{0,1\}}[|F| - |F'_\pi|] \geq \frac{3}{2}s_i$$

$$\implies \mathbb{E}_{i\in[n]}\left[E_{j\in\{0,1\}}[|F| - |F'_\pi|]\right] \geq E_{i\in[n]}\left[\frac{3}{2}s_i\right]$$

$$\implies \mathbb{E}_\pi[|F| - |F'_\pi|]] \geq \frac{3}{2}\cdot\frac{\sum_{i=1}^n s_i}{n} = \frac{3|F|}{2n}$$

$$\implies \mathbb{E}_\pi[|F_\pi|] \leq \left(1 - \frac{1.5}{n}\right)|F|$$

□

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables, then for optimal nice circuits F and F_π for f and f_π,

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables, then for optimal nice circuits F and F_π for f and f_π,we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(\frac{n-k}{n}\right)^{1.5} |F|$$

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables, then for optimal nice circuits F and F_π for f and f_π,we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(\frac{n-k}{n}\right)^{1.5} |F|$$

We will use Lemma 1 in succession to prove this.

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables, then for optimal nice circuits F and F_π for f and f_π,we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(\frac{n-k}{n}\right)^{1.5} |F|$$

We will use Lemma 1 in succession to prove this. When we randomly pick π, suppose we "mark" π with a uniformly random order $x_{j_1}, x_{j_2} \ldots x_{j_k}$ on the variables fixed by π.

Subbotovskaya's Method

Lemma 2

If π is chosen uniformly from the set of restrictions on $x_1, \ldots x_n$, $n \geq 2$ restricting exactly $k \leq n-1$ variables, then for optimal nice circuits F and F_π for f and f_π,we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \left(\frac{n-k}{n}\right)^{1.5} |F|$$

We will use Lemma 1 in succession to prove this. When we randomly pick π, suppose we "mark" π with a uniformly random order $x_{j_1}, x_{j_2} \ldots x_{j_k}$ on the variables fixed by π. This does not change the expectation as each restriction can be marked with $k!$ orderings.

Proof of Lemma 2

For each π(along with the ordering), define π_i to be the subrestriction of π which sets the value of only first i x_{j_i}'s. So, $f_{\pi_0} = f$, and $f_{\pi_k} = f_\pi$.

Proof of Lemma 2

For each π(along with the ordering), define π_i to be the subrestriction of π which sets the value of only first i x_{j_i}'s. So, $f_{\pi_0} = f$, and $f_{\pi_k} = f_\pi$.
Now, from Lemma 1, we know that

$$\mathbb{E}_{\pi_{i+1}|\pi_i}[|F_{\pi_{i+1}}|] \leq \left(1 - \frac{1.5}{n-i}\right) |F_{\pi_i}|$$

Proof of Lemma 2

For each π(along with the ordering), define π_i to be the subrestriction of π which sets the value of only first i x_{j_i}'s. So, $f_{\pi_0} = f$, and $f_{\pi_k} = f_\pi$. Now, from Lemma 1, we know that

$$\mathbb{E}_{\pi_{i+1}|\pi_i}[|F_{\pi_{i+1}}|] \leq \left(1 - \frac{1.5}{n-i}\right) |F_{\pi_i}|$$

$$\implies \mathbb{E}_{\pi_{i+1}}[|F_{\pi_{i+1}}|] \leq \left(1 - \frac{1.5}{n-i}\right) \mathbb{E}_{\pi_i}[|F_{\pi_i}|]$$

$$\implies \mathbb{E}_{\pi}[|F_{\pi}|] \leq |F| \prod_{i=0}^{k-1} \left(1 - \frac{1.5}{n-i}\right)$$

Proof of Lemma 2

$$\mathbb{E}_\pi[|F_\pi|] \quad \leq |F| \prod_{i=0}^{k-1} \left(1 - \frac{1.5}{n-i}\right)$$

Proof of Lemma 2

$$\begin{aligned}
\mathbb{E}_{\pi}[|F_{\pi}|] &\leq |F| \prod_{i=0}^{k-1} \left(1 - \frac{1.5}{n-i}\right) \\
&\leq |F| \prod_{i=0}^{k-1} \left(1 - \frac{1}{n-i}\right)^{1.5} \quad [(1-x)^c \geq (1-cx)]
\end{aligned}$$

Proof of Lemma 2

$$
\begin{aligned}
\mathbb{E}_\pi[|F_\pi|] &\le |F| \prod_{i=0}^{k-1} \left(1 - \frac{1.5}{n-i}\right) \\
&\le |F| \prod_{i=0}^{k-1} \left(1 - \frac{1}{n-i}\right)^{1.5} \quad [(1-x)^c \ge (1-cx)] \\
&= |F| \left(\frac{n-k}{n}\right)^{1.5}
\end{aligned}
$$

$\square$

Theorem

Every formula computing *PARITY* has size $\Omega(n^{1.5})$

Proof :
Let f be the function computing parity on n variables $x_1 \dots x_n$, and F be an optimal nice formula for it.

Subbotovskaya's Method

Theorem

Every formula computing *PARITY* has size $\Omega(n^{1.5})$

Proof :

Let f be the function computing parity on n variables $x_1 \dots x_n$, and F be an optimal nice formula for it. Consider π uniformly chosen from the set of restrictions on $x_1, x_2 \dots x_n$, setting the values of exactly $n-1$ of these.

Subbotovskaya's Method

Theorem

Every formula computing *PARITY* has size $\Omega(n^{1.5})$

Proof :
Let f be the function computing parity on n variables $x_1 \dots x_n$, and F be an optimal nice formula for it. Consider π uniformly chosen from the set of restrictions on $x_1, x_2 \dots x_n$, setting the values of exactly $n-1$ of these. From Lemma 2, we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \frac{|F|}{n^{1.5}}$$

Subbotovskaya's Method

Theorem

Every formula computing *PARITY* has size $\Omega(n^{1.5})$

Proof :
Let f be the function computing parity on n variables $x_1 \dots x_n$, and F be an optimal nice formula for it. Consider π uniformly chosen from the set of restrictions on $x_1, x_2 \dots x_n$, setting the values of exactly $n - 1$ of these. From Lemma 2, we have

$$\mathbb{E}_\pi[|F_\pi|] \leq \frac{|F|}{n^{1.5}}$$

So, there must exist some π for which

$$|F_\pi| \leq \frac{|F|}{n^{1.5}}$$

Subbotovskaya's Method

Since π restricts only $n-1$ variables, $|F_\pi| \geq 1$

$$1 \leq \frac{|F|}{n^{1.5}} \implies |F| \geq n^{1.5}$$

$\square$

Andreev's Method

Andreev's Method

- Used Subbotovskaya's Method of Random restrictions to obtain a lower bound for a different function.

Andreev's Method

- Used Subbotovskaya's Method of Random restrictions to obtain a lower bound for a different function.

Definition 1

Consider the function $f : \{0,1\}^{2n} \to \{0,1\}$ such that $f(x,y)$, for $x, y \in \{0,1\}^n$, computes the following:

- Take x to be the the truth table of a boolean function ϕ_x on $\lg n$ bits.
- Let the string y be taken as the entries of matrix Y with $\lg n$ rows and $\frac{n}{\lg n}$ colums.

Then

$$f(x,y) = \phi_x\left(\oplus_{j=1}^{\frac{n}{\lg n}} Y_{1j}, \oplus_{j=1}^{\frac{n}{\lg n}} Y_{2j} \cdots \oplus_{j=1}^{\frac{n}{\lg n}} Y_{(\lg n)j}\right)$$

Note that this function can be easily computed.

Andreev's Method

Theorem

Any formula that computes f as defined in the previous definition has size $\Omega(n^{2.5-\epsilon})$ for every $\epsilon > 0$

Proof : Take an optimal nice formula (as defined in previous section) F for computing f.
Note that ϕ_x can vary over all the boolean functions on $\lg n$ bits.

Claim 1

There is a ϕ_x which requires a boolean formula of size $c\frac{n}{\lg n}$, where $c > 0$ is a constant not dependent on n.

Andreev's Method

Theorem

Any formula that computes f as defined in the previous definition has size $\Omega(n^{2.5-\epsilon})$ for every $\epsilon > 0$

Proof : Take an optimal nice formula (as defined in previous section) F for computing f.

Note that ϕ_x can vary over all the boolean functions on $\lg n$ bits.

Claim 1

There is a ϕ_x which requires a boolean formula of size $c\frac{n}{\lg n}$, where $c > 0$ is a constant not dependent on n.

This is seen by the non uniform size heirarchy theorem mentioned in Lecture 9.

Andreev's Method

Theorem

Any formula that computes f as defined in the previous definition has size $\Omega(n^{2.5-\epsilon})$ for every $\epsilon > 0$

Proof : Take an optimal nice formula (as defined in previous section) F for computing f.
Note that ϕ_x can vary over all the boolean functions on $\lg n$ bits.

Claim 1

There is a ϕ_x which requires a boolean formula of size $c\frac{n}{\lg n}$, where $c > 0$ is a constant not dependent on n.

This is seen by the non uniform size heirarchy theorem mentioned in Lecture 9. We fix such a ϕ, and let x_0 represent the truth table for this ϕ.

Andreev's Method

Let f_ϕ denote the function obtained by restricting x to x_0 in f. Let F_ϕ be the optimal nice formula computing f_ϕ.

Andreev's Method

Let f_ϕ denote the function obtained by restricting x to x_0 in f. Let F_ϕ be the optimal nice formula computing f_ϕ. Clearly, $|F_\phi| \leq |F|$, as one can lazily evaluate F.

Andreev's Method

Let f_ϕ denote the function obtained by restricting x to x_0 in f. Let F_ϕ be the optimal nice formula computing f_ϕ. Clearly, $|F_\phi| \leq |F|$, as one can lazily evaluate F.

Now, the main idea of the proof is to obtain a small formula for ϕ from F_ϕ, which will put a lower bound on $|F_\phi|$, and thus $|F|$.

Andreev's Method

Let f_ϕ denote the function obtained by restricting x to x_0 in f. Let F_ϕ be the optimal nice formula computing f_ϕ. Clearly, $|F_\phi| \leq |F|$, as one can lazily evaluate F.

Now, the main idea of the proof is to obtain a small formula for ϕ from F_ϕ, which will put a lower bound on $|F_\phi|$, and thus $|F|$.

We again utilize random restrictions, and the Lemma 2 proved in the previous section.

Andreev's Method

Let f_ϕ denote the function obtained by restricting x to x_0 in f. Let F_ϕ be the optimal nice formula computing f_ϕ. Clearly, $|F_\phi| \leq |F|$, as one can lazily evaluate F.
Now, the main idea of the proof is to obtain a small formula for ϕ from F_ϕ, which will put a lower bound on $|F_\phi|$, and thus $|F|$.
We again utilize random restrictions, and the Lemma 2 proved in the previous section.
Let π be chosen randomly from the set of restrictions on the input variables of f_ϕ (there are n of those remaining) restricting exactly $n - k$ variables for $k = 10 \lg n \lg \lg n$

Andreev's Method

Claim 2

With probability ≥ 0.9, the following happens:

For each $i \in [\lg n]$, atleast 1 Y_{ij} has not been set by π.

Claim 2

With probability ≥ 0.9, the following happens:

For each $i \in [\lg n]$, atleast 1 Y_{ij} has not been set by π.

Proof :

Andreev's Method

Claim 2

With probability ≥ 0.9, the following happens:

For each $i \in [\lg n]$, atleast 1 Y_{ij} has not been set by π.

Proof : For an individual i, the probability that no Y_{ij} survived is clearly

$$p = \frac{\binom{n\left(1-\frac{1}{\lg n}\right)}{k}}{\binom{n}{k}} = \prod_{i=0}^{k-1} \frac{n\left(1-\frac{1}{\lg n}\right)-i}{n-i} \leq \prod_{i=0}^{k-1}\left(1-\frac{1}{\lg n}\right)$$

$$\implies p \leq \left(1-\frac{1}{\lg n}\right)^{k} = \left(1-\frac{1}{\lg n}\right)^{\lg n \cdot 10 \lg \lg n} \leq \frac{1}{(\lg n)^{10}}$$

Andreev's Method

Claim 2

With probability ≥ 0.9, the following happens:
For each $i \in [\lg n]$, atleast 1 Y_{ij} has not been set by π.

Proof : For an individual i, the probability that no Y_{ij} survived is clearly

$$p = \frac{\binom{n\left(1-\frac{1}{\lg n}\right)}{k}}{\binom{n}{k}} = \prod_{i=0}^{k-1} \frac{n\left(1-\frac{1}{\lg n}\right)-i}{n-i} \leq \prod_{i=0}^{k-1}\left(1-\frac{1}{\lg n}\right)$$

$$\implies p \leq \left(1-\frac{1}{\lg n}\right)^{k} = \left(1-\frac{1}{\lg n}\right)^{\lg n \cdot 10 \lg\lg n} \leq \frac{1}{(\lg n)^{10}}$$

Now, by union bound, the probability of no Y_{ij} surviving for atleast 1 is i $\leq \frac{1}{(\lg n)^9} \leq 0.1$ for large enough n.

Andreev's Method

Now, let the function obtained by the restriction π applied on function f_ϕ be denoted f_π, and the corresponding optimal nice formula be F_π

Andreev's Method

Now, let the function obtained by the restriction π applied on function f_ϕ be denoted f_π, and the corresponding optimal nice formula be F_π Now, by the Lemma 2 of the previous section, we have that

$$\mathbb{E}_\pi[|F_\pi|] \leq |F_\phi| \left(\frac{k}{n}\right)^{1.5}$$

Andreev's Method

Now, let the function obtained by the restriction π applied on function f_ϕ be denoted f_π, and the corresponding optimal nice formula be F_π Now, by the Lemma 2 of the previous section, we have that

$$\mathbb{E}_\pi[|F_\pi|] \leq |F_\phi| \left(\frac{k}{n}\right)^{1.5}$$

So, Markov bound gives that there is a constant $c' > 0$ such that

$$Pr_\pi \left[|F_\pi| > 10|F_\phi| \left(\frac{k}{n}\right)^{1.5} \right] \leq 0.1 \frac{\mathbb{E}_\pi[|F_\pi|]}{|F_\phi| \left(\frac{k}{n}\right)^{1.5}} \leq 0.1$$

Andreev's Method

Now, let the function obtained by the restriction π applied on function f_ϕ be denoted f_π, and the corresponding optimal nice formula be F_π Now, by the Lemma 2 of the previous section, we have that

$$\mathbb{E}_\pi[|F_\pi|] \leq |F_\phi| \left(\frac{k}{n}\right)^{1.5}$$

So, Markov bound gives that there is a constant $c' > 0$ such that

$$Pr_\pi \left[|F_\pi| > 10|F_\phi| \left(\frac{k}{n}\right)^{1.5} \right] \leq 0.1 \frac{\mathbb{E}_\pi[|F_\pi|]}{|F_\phi| \left(\frac{k}{n}\right)^{1.5}} \leq 0.1$$

Andreev's Method

So, again by union bound, we have that with atmost 0.2 probability, π restricts the value of all Y_{ij} for atleast 1 i OR $|F_\pi| \geq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.

Andreev's Method

So, again by union bound, we have that with atmost 0.2 probability, π restricts the value of all Y_{ij} for atleast 1 i OR $|F_\pi| \geq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.

So, there must be a π such that atleast 1 Y_{ij} survives for each i and $|F_\pi| \leq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.

Andreev's Method

So, again by union bound, we have that with atmost 0.2 probability, π restricts the value of all Y_{ij} for atleast 1 i OR $|F_\pi| \geq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.

So, there must be a π such that atleast 1 Y_{ij} survives for each i and $|F_\pi| \leq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.
Now, with this F_π, we can construct a formula for ϕ which atmost as large as as F_π.

Andreev's Method

So, again by union bound, we have that with atmost 0.2 probability, π restricts the value of all Y_{ij} for atleast 1 i OR $|F_\pi| \geq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.

So, there must be a π such that atleast 1 Y_{ij} survives for each i and $|F_\pi| \leq 10|F_\phi| \left(\frac{k}{n}\right)^{1.5}$.
Now, with this F_π, we can construct a formula for ϕ which atmost as large as as F_π.
This can be done by simply restricting all but one Y_{ij} for each i in F_π, and then to compute $\phi(x_1, x_2 \ldots x_{\lg n})$, we simply set the remaining Y_{ij} in F_π to x_i or $\neg x_i$ as suitable, for each $i \in [\lg n]$.

Andreev's Method

So, again by union bound, we have that with atmost 0.2 probability, π restricts the value of all Y_{ij} for atleast 1 i OR $|F_\pi| \geq 10|F_\phi|\left(\frac{k}{n}\right)^{1.5}$.

So, there must be a π such that atleast 1 Y_{ij} survives for each i and $|F_\pi| \leq 10|F_\phi|\left(\frac{k}{n}\right)^{1.5}$.
Now, with this F_π, we can construct a formula for ϕ which atmost as large as as F_π.
This can be done by simply restricting all but one Y_{ij} for each i in F_π, and then to compute $\phi(x_1, x_2 \ldots x_{\lg n})$, we simply set the remaining Y_{ij} in F_π to x_i or $\neg x_i$ as suitable, for each $i \in [\lg n]$.
So, by our choice of ϕ, and this specific π, we get that

$$\frac{cn}{\lg n} \leq |F_\pi| \leq 10|F_\phi|\left(\frac{k}{n}\right)^{1.5}$$

Andreev's Method

Setting the value of k, and using that $|F| \geq |F_\phi|$

$$|F| \geq \frac{c \cdot n^{2.5}}{10^{2.5}(\lg n)^{2.5}(\lg \lg n)^{1.5}}$$

Setting the value of k, and using that $|F| \geq |F_\phi|$

$$|F| \geq \frac{c \cdot n^{2.5}}{10^{2.5}(\lg n)^{2.5}(\lg \lg n)^{1.5}}$$

So, the size of any nice formula computing f is lower bounded by $\Omega(n^{2.5-\epsilon})$ for every $\epsilon > 0$.

Andreev's Method

Setting the value of k, and using that $|F| \geq |F_\phi|$

$$|F| \geq \frac{c \cdot n^{2.5}}{10^{2.5}(\lg n)^{2.5}(\lg \lg n)^{1.5}}$$

So, the size of any nice formula computing f is lower bounded by $\Omega(n^{2.5-\epsilon})$ for every $\epsilon > 0$.
As shown earlier, since the order of size the overall optimal formula and the optimal nice formula is same, the above bound also holds in general.

$\square$

Andreev's Method

We also briefly discuss the exercise mentioned at the end of this section in the lecture notes.

Exercise

Show that f can be computed by a formula of size $O(n^3)$

Andreev's Method

We also briefly discuss the exercise mentioned at the end of this section in the lecture notes.

Exercise

Show that f can be computed by a formula of size $O(n^3)$

We do this as follows:

1. Find a formula for $g(x, y_1, y_2 \ldots y_{\lg n}) = \phi_x(y_1, y_2 \ldots y_{\lg n})$ with size atmost $O(n)$. Naive approach leads to $O(n \lg n)$ (which would also work). Here, x is again the truth table for some function ϕ on $\lg n$ variables

Andreev's Method

We also briefly discuss the exercise mentioned at the end of this section in the lecture notes.

Exercise

Show that f can be computed by a formula of size $O(n^3)$

We do this as follows:

1. Find a formula for $g(x, y_1, y_2 \ldots y_{\lg n}) = \phi_x(y_1, y_2 \ldots y_{\lg n})$ with size atmost $O(n)$. Naive approach leads to $O(n \lg n)$ (which would also work). Here, x is again the truth table for some function ϕ on $\lg n$ variables
2. Compute each $\bigoplus_{j=1}^{\frac{n}{\lg n}} Y_{ij}$ and with $O(\frac{n^2}{(\lg n)^2})$ formula.

Andreev's Method

We also briefly discuss the exercise mentioned at the end of this section in the lecture notes.

Exercise

Show that f can be computed by a formula of size $O(n^3)$

We do this as follows:

1. Find a formula for $g(x, y_1, y_2 \ldots y_{\lg n}) = \phi_x(y_1, y_2 \ldots y_{\lg n})$ with size atmost $O(n)$. Naive approach leads to $O(n \lg n)$ (which would also work). Here, x is again the truth table for some function ϕ on $\lg n$ variables
2. Compute each $\bigoplus_{j=1}^{\frac{n}{\lg n}} Y_{ij}$ and with $O(\frac{n^2}{(\lg n)^2})$ formula.

Replacing each y_i in 1) with the corresponding formula from 2 will lead to a $O(n^3(\lg n)^{-2})$ size formula.

Andreev's Method

Discussing point 1. Naive method($O(n \lg n)$):

$$\bigvee_{a_i \in \{y_i, \neg y_i\}} \left(\phi(a_1, a_2 \ldots a_{\lg n}) \wedge \left(\bigwedge_{i \in [\lg n]} a_i \right) \right)$$

Discussing point 1. Naive method($O(n \lg n)$):

$$\bigvee_{a_i \in \{y_i, \neg y_i\}} \left(\phi(a_1, a_2 \ldots a_{\lg n}) \wedge \left(\bigwedge_{i \in [\lg n]} a_i \right) \right)$$

Combining $y_{\lg n}$ terms and $\neg y_{\lg n}$ terms

$$\left(y_{\lg n} \wedge \bigvee_{a_i \in \{y_i, \neg y_i\}} \left(x_k^1 \wedge \left(\bigwedge_{i \in [\lg n - 1]} a_i \right) \right) \right) \vee$$

$$\left(\neg y_{\lg n} \wedge \bigvee_{a_i \in \{y_i, \neg y_i\}} \left(x_k^0 \wedge \left(\bigwedge_{i \in [\lg n - 1]} a_i \right) \right) \right)$$

Andreev's Method

Discussing point 1. Naive method($O(n \lg n)$):

$$\bigvee_{a_i \in \{y_i, \neg y_i\}} \left(\phi(a_1, a_2 \ldots a_{\lg n}) \wedge \left(\bigwedge_{i \in [\lg n]} a_i \right) \right)$$

Combining $y_{\lg n}$ terms and $\neg y_{\lg n}$ terms

$$\left(y_{\lg n} \wedge \bigvee_{a_i \in \{y_i, \neg y_i\}} \left(x_k^1 \wedge \left(\bigwedge_{i \in [\lg n - 1]} a_i \right) \right) \right) \vee$$

$$\left(\neg y_{\lg n} \wedge \bigvee_{a_i \in \{y_i, \neg y_i\}} \left(x_k^0 \wedge \left(\bigwedge_{i \in [\lg n - 1]} a_i \right) \right) \right)$$

Repeating this, we get an $O(n)$ formula.

Nechiporuk's Method

Nechiporuk's Method

- Nechiporuk's method gives lower bound on formula size of some explicit functions in full binary basis.

Nechiporuk's Method

- Nechiporuk's method gives lower bound on formula size of some explicit functions in full binary basis.
- Nechiporuk bound gives a lower bound of $\Omega(n^2)$ which is the best known lower bound on the size of formulae over full binary basis

Nechiporuk's Method

- Nechiporuk's method gives lower bound on formula size of some explicit functions in full binary basis.
- Nechiporuk bound gives a lower bound of $\Omega(n^2)$ which is the best known lower bound on the size of formulae over full binary basis
- The key ingredient in Nechiporuk's Method is the idea of restrictions, to understand what we exactly mean by that we have to start with definition of restrictions.

Restrictions

Restrictions

Definition 1

Let f be a function on $\{0,1\}^n$ and Y be a subset of input variables. $N_f(Y)$ is the number of distinct functions $g : \{0,1\}^Y \to \{0,1\}$ we can get when we fix all the input bits except Y. Each of this subfunction g is said to be a restriction of f on Y.

Restrictions

Definition 1

Let f be a function on $\{0,1\}^n$ and Y be a subset of input variables. $N_f(Y)$ is the number of distinct functions $g : \{0,1\}^Y \to \{0,1\}$ we can get when we fix all the input bits except Y. Each of this subfunction g is said to be a restriction of f on Y.

- Intuitively larger value of $N_f(Y)$ means that the function is harder to compute hence the corresponding formula would be of larger size. Nechiporuk's Method uses this notion of "hard" functions to give a lower bound.

Block Distinction Functions

Block Distinction Functions

Definition 2

Let f be a boolean function on n variables. We can group variables to form a block i.e

$$f(x_1,\ldots,x_n) = f(\underline{x_1,\ldots,x_c},\underline{x_{c+1},\ldots,x_{2c}},\ldots,\underline{x_{(b-1)c+1},\ldots,x_{bc}})$$

Here we have b blocks each having c variables. Then the function f is given as

$$f = \begin{cases} 1 \text{ if all the } Y_i's \text{ are different} \\ 0 \text{ otherwise} \end{cases}$$

We will call function described in the above as "Block Distinction Functions".

Block Distinction Functions

Definition 2

Let f be a boolean function on n variables. We can group variables to form a block i.e

$$f(x_1,\ldots,x_n) = f(\underline{x_1,\ldots,x_c},\underline{x_{c+1},\ldots,x_{2c}},\ldots,\underline{x_{(b-1)c+1},\ldots,x_{bc}})$$

Here we have b blocks each having c variables. Then the function f is given as

$$f = \begin{cases} 1 \text{ if all the } Y_i's \text{ are different} \\ 0 \text{ otherwise} \end{cases}$$

We will call function described in the above as "Block Distinction Functions".

- These functions are also called "Elements Distinction function" where instead of grouping terms we give block as inputs hence the name.

Block Distinction Functions

Definition 2

Let f be a boolean function on n variables. We can group variables to form a block i.e

$$f(x_1, \ldots, x_n) = f(\underline{x_1, \ldots, x_c}, \underline{x_{c+1}, \ldots, x_{2c}}, \ldots, \underline{x_{(b-1)c+1}, \ldots, x_{bc}})$$

Here we have b blocks each having c variables. Then the function f is given as

$$f = \begin{cases} 1 \text{ if all the } Y_i^{\prime}s \text{ are different} \\ 0 \text{ otherwise} \end{cases}$$

We will call function described in the above as "Block Distinction Functions".

- Nechiporuk's method provides a lower bound on the size of formula computing Block Distiction functions of specific block sizes over the full binary basis.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Theorem 2 (Nechiporuk's Bound)

Let $f:\{0,1\}^n \to \{0,1\}$ be a boolean function on the the set of variables $X=\{x_1,\dots,x_n\}$, now let $Y_1,\dots,Y_m$ be the disjoint subsets of X. Then

$$L_{B_2}(f)+1 \geq \frac{1}{4}\sum_{i=1}^{m} log(N_f(Y_i))$$

where $L_{B_2}(.)$ gives the formula complexity in the full binary basis.

Nechiporuk's Method

- Theorem 1 gives us the class of functions whose formula complexity is lower bounded by $\Omega(n^2)$ (ignoring the log factor).

Nechiporuk's Method

- Theorem 1 gives us the class of functions whose formula complexity is lower bounded by $\Omega(n^2)$ (ignoring the log factor).
- Theorem 2 tells us about the pivotal idea involved in this method i.e how counting the number of restrictions will help us lower bound the formula complexity.

Nechiporuk's Method

- Theorem 1 gives us the class of functions whose formula complexity is lower bounded by $\Omega(n^2)$ (ignoring the log factor).
- Theorem 2 tells us about the pivotal idea involved in this method i.e how counting the number of restrictions will help us lower bound the formula complexity.
- We will prove theorem 1 as doing so will implicitly prove 2, I will remark on this at the end.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Proof.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Proof.

- Going by our intuition we will first try to bound $N_f(Y_i)$.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)} \log(\log(n)))$.

Proof.

- Going by our intuition we will first try to bound $N_f(Y_i)$.
- Consider fixing $\{Y_1, \ldots, Y_b\} \setminus Y_i$. Now we have two cases
  1. If two blocks have same assignment. Then the restriction of function to Y_i is a zero function.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)}\log(\log(n)))$.

Proof.

- Going by our intuition we will first try to bound $N_f(Y_i)$.
- Consider fixing $\{Y_1, \ldots, Y_b\} \setminus Y_i$. Now we have two cases
  1. If two blocks have same assignment. Then the restriction of function to Y_i is a zero function.
  2. If all the blocks have pairwise distinct assignment then corresponding each assignment we have unique restriction on Y.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)} \log(\log(n)))$.

Proof.

- Going by our intuition we will first try to bound $N_f(Y_i)$.
- Consider fixing $\{Y_1, \ldots, Y_b\} \setminus Y_i$. Now we have two cases
  1. If two blocks have same assignment. Then the restriction of function to Y_i is a zero function.
  2. If all the blocks have pairwise distinct assignment then corresponding each assignment we have unique restriction on Y.

Nechiporuk's Method

Theorem 1

Size of formula computing the Block Distinction function with size of each block equal to $\log(n)$ is $\Omega(\frac{n^2}{\log^2(n)} \log(\log(n)))$.

Proof.

- Going by our intuition we will first try to bound $N_f(Y_i)$.
- Consider fixing $\{Y_1, \ldots, Y_b\} \setminus Y_i$. Now we have two cases
  1. If two blocks have same assignment. Then the restriction of function to Y_i is a zero function.
  2. If all the blocks have pairwise distinct assignment then corresponding each assignment we have unique restriction on Y.
- From the above discussion we have $N_f(Y_i) = \binom{2^c}{b-1} + 1$.

Nechiporuk's Method

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.

Nechiporuk's Method

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.
- Consider fixing variables in $\{Y_1, \ldots, Y_b\} \setminus Y_i$.

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.
- Consider fixing variables in $\{Y_1, \ldots, Y_b\} \setminus Y_i$.
- **Observations :**
  1. The resulting formula would compute restriction function on Y_i.Also the induced formula would not have any leaves labelled by 0 and 1.

Nechiporuk's Method

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.
- Consider fixing variables in $\{Y_1, \ldots, Y_b\} \setminus Y_i$.
- **Observations :**
  1. The resulting formula would compute restriction function on Y_i.Also the induced formula would not have any leaves labelled by 0 and 1.
  2. Now since the resulting formula is a tree, the number of internal nodes is one less than the number of leaf nodes i.e $l(Y_i) - 1$.

Nechiporuk's Method

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.
- Consider fixing variables in $\{Y_1, \ldots, Y_b\} \setminus Y_i$.
- **Observations :**
  1. The resulting formula would compute restriction function on Y_i.Also the induced formula would not have any leaves labelled by 0 and 1.
  2. Now since the resulting formula is a tree, the number of internal nodes is one less than the number of leaf nodes i.e $l(Y_i) - 1$.
  3. The total number of distinct boolean valued functions taking 2 binary inputs is equal to $2^4 = 16$.

Nechiporuk's Method

Proof (Cont.)

- Consider F be the formula computing f. Let $l(Y_i)$ be the number of leaves labelled by variables of Y_i in F.
- Consider fixing variables in $\{Y_1, \ldots, Y_b\} \setminus Y_i$.
- **Observations :**
    1. The resulting formula would compute restriction function on Y_i.Also the induced formula would not have any leaves labelled by 0 and 1.
    2. Now since the resulting formula is a tree, the number of internal nodes is one less than the number of leaf nodes i.e $l(Y_i) - 1$.
    3. The total number of distinct boolean valued functions taking 2 binary inputs is equal to $2^4 = 16$.
- Thus from the above two observations the number of distinct boolean functions computable on variables on Y_i is $16^{l(Y_i)-1} \leq 16^{l(Y_i)}$.

Nechiporuk's Method

Proof (Cont.)

- We now have $N_f(Y_i) = \binom{2^c}{b-1} + 1$ and number of distinct boolean function is bounded from above by $16^{l(Y_i)}$.

Nechiporuk's Method

Proof (Cont.)

- We now have $N_f(Y_i) = \binom{2^c}{b-1} + 1$ and number of distinct boolean function is bounded from above by $16^{l(Y_i)}$.Combining this two we get the following lower bound for $l(Y_i)$

$$l(Y_i) \geq log\binom{2^c}{b-1}$$

for all i.

Nechiporuk's Method

Proof (Cont.)

- Now the size of the formula is lower bounded by the number of leaves i.e

$$
\begin{aligned}
|F| &\geq \sum_{i=1}^{b} l(Y_i) \\
&\geq \sum_{i=1}^{b} log(\binom{2^c}{b-1}) \\
&\geq \sum_{i=1}^{b} (b-1) log(\frac{2^c}{b-1}) \ [\binom{n}{k} \geq (\frac{n}{k})^k] \ [2] \\
&\geq b(b-1) log(\frac{2^c}{b-1}) \geq b(b-1) log(\frac{2^c}{b})
\end{aligned}
$$

Nechiporuk's Method

Proof (Cont.)

- We now have $|F| \geq b(b-1)log(\frac{2^c}{b})$.

Nechiporuk's Method

Proof (Cont.)

- We now have $|F| \geq b(b-1)log(\frac{2^c}{b})$.
- Substisuting the value of $c = log(n)$ and $b = \frac{n}{log(n)}$ we get

$$|F| = \Omega(\frac{n^2}{\log^2(n)} \log(\log(n)))$$

Proof (Cont.)

- We now have $|F| \geq b(b-1)log(\frac{2^c}{b})$.
- If we have $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some $k \in \mathbb{N}$. Now substituting these into our above equation gives us

$$|F| = \Omega(\frac{n^2}{\log(n)})$$

$\square$

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.
- Nechiporuk method provides a tight lower bound for Block Estimation functions with $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some for $k \in \mathbb{N}$ i.e there is a formula in full binary basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ infact we can give a formula in de-Morgan basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ computing the above function.

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.
- Nechiporuk method provides a tight lower bound for Block Estimation functions with $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some for $k \in \mathbb{N}$ i.e there is a formula in full binary basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ infact we can give a formula in de-Morgan basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ computing the above function.
  1. So in our circuit we will make b^2 comparisons(consider them as a black box computing whether the two blocks are equal) between any two blocks.

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.
- Nechiporuk method provides a tight lower bound for Block Estimation functions with $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some for $k \in \mathbb{N}$ i.e there is a formula in full binary basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ infact we can give a formula in de-Morgan basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ computing the above function.
  1. So in our circuit we will make b^2 comparisons(consider them as a black box computing whether the two blocks are equal) between any two blocks.
  2. Now in such comparison between the blocks we make c comparisons(here comparison means AND operations) between corresponding elements in the block.

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.
- Nechiporuk method provides a tight lower bound for Block Estimation functions with $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some for $k \in \mathbb{N}$ i.e there is a formula in full binary basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ infact we can give a formula in de-Morgan basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ computing the above function.
  1. So in our circuit we will make b^2 comparisons(consider them as a black box computing whether the two blocks are equal) between any two blocks.
  2. Now in such comparison between the blocks we make c comparisons(here comparison means AND operations) between corresponding elements in the block.
  3. The total complexity is $\mathcal{O}(c \cdot b^2 + 2^{log(c \cdot b^2)}) = \mathcal{O}(\frac{n^2}{log(n)})$

Nechiporuk's Method

- As promised we implicitly proved theorem 2 (Nechiporuk's bound) in proof of theorem 1.
- Nechiporuk method provides a tight lower bound for Block Estimation functions with $n = 2k \cdot 2^k$, $b = 2^k$ and $c = 2k$ for some for $k \in \mathbb{N}$ i.e there is a formula in full binary basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ infact we can give a formula in de-Morgan basis of size $\mathcal{O}(\frac{n^2}{log(n)})$ computing the above function.
- Nechiporuk bound is the best known lower bound on the size of formulae over full binary basis

Can we do better?

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the Nechiporuk's method ?

Can we do better?

- In this section we are going to answer the question on what is the best lower bound given by the Nechiporuk's method ?
- It turns out that Nechiporuk's method cannot provide us with a lower bound better than $\Omega(\frac{n^2}{log(n)})$.

Can we do better?

- Before we prove the above statement let's state by what we mean by "lower bound given by Nechiporuk's Method". In the above proof as mentioned before the pivotal tool used is theorem 2.

Can we do better?

- Before we prove the above statement let's state by what we mean by "lower bound given by Nechiporuk's Method". In the above proof as mentioned before the pivotal tool used is theorem 2.
  Theorem 2 says that consider any disjoint subsets $Y_1, Y_2, \ldots, Y_m$ of X then the size of the formula is lower bounded by the sum of number of unique restriction on these Y_i's i.e

$$|F| \geq \frac{1}{4}\sum_{i=1}^{m} log(N_f(Y_i))$$

Can we do better?

- Before we prove the above statement let's state by what we mean by "lower bound given by Nechiporuk's Method". In the above proof as mentioned before the pivotal tool used is theorem 2.
  Theorem 2 says that consider any disjoint subsets $Y_1, Y_2, \ldots, Y_m$ of X then the size of the formula is lower bounded by the sum of number of unique restriction on these Y_i's i.e

  $$|F| \geq \frac{1}{4}\sum_{i=1}^{m} log(N_f(Y_i))$$

  So when we say "lower bound given by Nechiporuk's method" we actually mean lower bound given by theorem 2 i.e the lower bound given by the RHS of the above equation . Hence to see what is the best lower bound given by Nechiporuk's method we have to find the maximum value of the above expression over the choice of the subsets Y_i's. .

Can we do better?

Lemma 1

Nechiporuk's Method cannot give a lower bound better than $\Omega(\frac{n^2}{log(n)})$

Proof.

- As discussed before it suffices to proving that maximum value attained by $\sum_{i=1}^{m} log(N_f(Y_i))$ cannot be greater than $\frac{n^2}{log(n)}$ upto a constant factor(depends on n but not on the choice of subsets) , where Y_i's are disjoint subset of X and maximum is taken over all the choices of disjoint subsets.

Can we do better?

Lemma 1

Nechiporuk's Method cannot give a lower bound better than $\Omega(\frac{n^2}{log(n)})$

Proof.

- As discussed before it suffices to proving that maximum value attained by $\sum_{i=1}^{m} log(N_f(Y_i))$ cannot be greater than $\frac{n^2}{log(n)}$ upto a constant factor(depends on n but not on the choice of subsets) , where Y_i's are disjoint subset of X and maximum is taken over all the choices of disjoint subsets.
- Now as $N_f(Y_i)$ is counting unique restriction functions $g : \{0,1\}^{Y_i} \rightarrow \{0,1\}$ of f on Y_i, hence $N_f(Y_i) \leq 2^{2^{|Y_i|}}$ (which is the total number of boolean function on $\{0,1\}^{Y_i}$) .

Can we do better?

Lemma 1

Nechiporuk's Method cannot give a lower bound better than $\Omega(\frac{n^2}{log(n)})$

Proof.

- As discussed before it suffices to proving that maximum value attained by $\sum_{i=1}^{m} log(N_f(Y_i))$ cannot be greater than $\frac{n^2}{log(n)}$ upto a constant factor(depends on n but not on the choice of subsets) , where Y_i's are disjoint subset of X and maximum is taken over all the choices of disjoint subsets.
- Now as $N_f(Y_i)$ is counting unique restriction functions $g : \{0,1\}^{Y_i} \rightarrow \{0,1\}$ of f on Y_i, hence $N_f(Y_i) \leq 2^{2^{|Y_i|}}$ (which is the total number of boolean function on $\{0,1\}^{Y_i}$) .
- Now notice that once we fix all variables except Y_i then the restriction function g of f on Y_i , is fixed and given by $g(Y) = f(Y_1, \ldots, Y_{i-1}, Y, \ldots, Y_n)$. Thus $N_f(Y_i) \leq 2^{n-|Y_i|}$.

Can we do better?

Lemma 1

Nechiporuk's Method cannot give a lower bound better than $\Omega(\frac{n^2}{log(n)})$

Proof.

- As discussed before it suffices to proving that maximum value attained by $\sum_{i=1}^{m} log(N_f(Y_i))$ cannot be greater than $\frac{n^2}{log(n)}$ upto a constant factor(depends on n but not on the choice of subsets) , where Y_i's are disjoint subset of X and maximum is taken over all the choices of disjoint subsets.
- Now as $N_f(Y_i)$ is counting unique restriction functions $g : \{0,1\}^{Y_i} \rightarrow \{0,1\}$ of f on Y_i, hence $N_f(Y_i) \leq 2^{2^{|Y_i|}}$ (which is the total number of boolean function on $\{0,1\}^{Y_i}$) .
- Now notice that once we fix all variables except Y_i then the restriction function g of f on Y_i , is fixed and given by $g(Y) = f(Y_1, \ldots, Y_{i-1}, Y, \ldots, Y_n)$. Thus $N_f(Y_i) \leq 2^{n-|Y_i|}$.

Can we do better?

Proof (Cont.)

- This tells us that $N_f(Y_i) \leq \min(2^{n-|Y_i|}, 2^{2^{|Y_i|}})$.

Can we do better?

Proof (Cont.)

- This tells us that $N_f(Y_i) \leq \min(2^{n-|Y_i|}, 2^{2^{|Y_i|}})$.
- Let's say $Y_1, \ldots, Y_m$ are subsets of X of size $s_1, \ldots, s_m$. Now,

$$\sum_{i=1}^{m} log(N_f(Y_i)) \leq \sum_{i=1}^{m} log(\min(2^{n-s_i}, 2^{2^{s_i}}))$$

Can we do better?

Proof (Cont.)

- This tells us that $N_f(Y_i) \leq \min(2^{n-|Y_i|}, 2^{2^{|Y_i|}})$.
- Let's say $Y_1, \ldots, Y_m$ are subsets of X of size $s_1, \ldots, s_m$. Now,

$$\sum_{i=1}^{m} log(N_f(Y_i)) \leq \sum_{i=1}^{m} log(\min(2^{n-s_i}, 2^{2^{s_i}}))$$

- Now I claim that for positive integers $s_1, \ldots, s_n, n$ s.t $\sum_{i=1}^{m} s_i \leq n$ then

$$\sum_{i=1}^{m} log(\min(2^{n-s_i}, 2^{2^{s_i}})) = \mathcal{O}(\frac{n^2}{log(n)})$$

holds.

Can we do better?

Proof (Cont.)

- Now I claim that for positive integers $s_1, \ldots, s_n, n$ s.t $\sum_{i=1}^{m} s_i \leq n$ then

$$\sum_{i=1}^{m} log(\min(2^{n-s_i}, 2^{2^{s_i}})) = \mathcal{O}(\frac{n^2}{log(n)})$$

holds

- Using the above inequality it is easy to see that

$$\sum_{i=1}^{m} log(N_f(Y_i)) = \mathcal{O}(\frac{n^2}{log(n)})$$

.

□

References

- https://sites.math.rutgers.edu/~sk1233/courses/topics-S13/lec2.pdf
- https://drive.google.com/file/d/1_MLcS0OAiAV7QtMUMmOsEAywrNEhTng1/view?usp=sharing
- http://www.math.toronto.edu/rossman/2429-L3.pdf
- **Krapchenko's Method Addendum:** https://drive.google.com/file/d/1czNtjozo1LHgLp9mlAr4b5ntFvr983On/view?usp=sharing