Lower Bounds in Boolean Circuits
CS 721 Seminar Presentation

H. Gupta and K. Mittal

Department of Computer Science
Indian Institute of Technology, Bombay

2 October 2018
Outline

1. Introduction

2. Parity, AC0 and the Switching Lemma

3. Proof of the Switching Lemma
Introduction

Boolean Circuits

For every \( n \in \mathbb{N} \), a boolean circuit \( C \) is a directed acyclic graph with \( n \) sources (vertices with no incoming edges) and one sink (vertex with no outgoing edges).

All non-source vertices are called gates and are labeled with one of \( \land, \lor \) or \( \neg \) (i.e., the logical operations \( AND, OR, \) and \( NOT \) respectively).

The size of \( C \), denoted by \(|C|\), is the number of vertices in it (including the source vertices).
Let $T : \mathbb{N} \rightarrow \mathbb{N}$ be a function. A $T(n)$-size circuit family is a sequence \( \{ C_n \}_{n \in \mathbb{N}} \) of Boolean circuits, where $C_n$ is a boolean circuit on $n$ inputs, and has size $|C_n| \leq T(n)$ for every $n$.

We say that a language $L$ is in $\text{SIZE}(T(n))$ if there is a $T(n)$-size circuit family $\{ C_n \}_{n \in \mathbb{N}}$ such that for every $x \in \{0, 1\}^n$, $x \in L \iff C_n(x) = 1$. 

Circuit Families
Circuit Families

Let \( T : \mathbb{N} \rightarrow \mathbb{N} \) be a function. A \( T(n) \)-size circuit family is a sequence \( \{C_n\}_{n \in \mathbb{N}} \) of Boolean circuits, where \( C_n \) is a boolean circuit on \( n \) inputs, and has size \( |C_n| \leq T(n) \) for every \( n \).

We say that a language \( L \) is in \( SIZE(T(n)) \) if there is a \( T(n) \)-size circuit family \( \{C_n\}_{n \in \mathbb{N}} \) such that for every \( x \in \{0, 1\}^n \), \( x \in L \iff C_n(x) = 1 \).

\( P/poly \)

\( P/poly \) is the class of languages that are decidable by polynomial-sized circuit families.

\[
P/poly = \bigcup_c SIZE(n^c)
\]
Theorem

\[ P \subseteq P/\text{poly} \] (in fact we have already seen that \( BPP \subseteq P/\text{poly} \))
**Introduction**

**Theorem**

\[ P \subseteq P_{/\text{poly}} \text{ (in fact we have already seen that } BPP \subseteq P_{/\text{poly}}) \]

**Why Study Circuit Lower Bounds**

- If there is any language \( L \in NP \), that doesn’t have poly-size circuits, then \( NP \neq P \).
Introduction

Theorem

\[ P \subseteq P_{/\text{poly}} \] (in fact we have already seen that \( BPP \subseteq P_{/\text{poly}} \))

Why Study Circuit Lower Bounds

- If there is any language \( L \in NP \), that doesn’t have poly-size circuits, then \( NP \neq P \).
- Karp-Lipton [KL82] showed that if the polynomial hierarchy (PH) doesn’t collapse, then there exists an \( NP \) language that doesn’t have polynomial circuits.
Introduction

Existence of Hard Functions [Sha49]

For every $n > 1$, there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\} \text{ that cannot be computed by a circuit } C \text{ of size } 2^n/(10n)$.
Existence of Hard Functions [Sha49]

For every $n > 1$, there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

As seen in class, such lower bounds on boolean circuits can be proved using counting arguments. Unfortunately, not much better about general circuits is known. We confine ourselves to some restricted circuit classes in this talk.
**Existence of Hard Functions [Sha49]**

For every \( n > 1 \), there exists a function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) that cannot be computed by a circuit \( C \) of size \( 2^n/(10n) \).

As seen in class, such lower bounds on boolean circuits can be proved using counting arguments. Unfortunately, not much better about general circuits is known. We confine ourselves to some restricted circuit classes in this talk.

**AC\(^i\)**

For every \( i \), a language \( L \) is in \( \text{AC}\(^i\) \) if \( L \) can be decided by a family of circuits \( \{C_n\} \) where \( C_n \) has \( \text{poly}(n) \) size, unbounded fan-in OR and AND gates, and depth \( O(\log^i(n)) \).

In particular we’ll look at \( \text{AC}\(^0\) \), i.e., constant depth circuits.
Main Theorem

Theorem ([FSS81], [Ajt83])

Let $\oplus$ be the parity function. That is, for every $x \in \{0, 1\}^n$,
$$\oplus(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \pmod{2}.$$
Main Theorem

Theorem ([FSS81], [Ajt83])

Let $\oplus$ be the parity function. That is, for every $x \in \{0, 1\}^n$,

$$\oplus(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \pmod{2}.$$ 

Then, $\oplus \notin AC^0$. 

H. Gupta and K. Mittal (IIT Bombay) 
Lower Bounds in Boolean Circuits 
2 October 2018 7 / 20
Main Theorem

Theorem ([FSS81], [Ajt83])

Let $\oplus$ be the parity function. That is, for every $x \in \{0, 1\}^n$,

$$\oplus(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \pmod{2}.$$  

Then, $\oplus \notin AC^0$.

Proof Sketch

The main tool for the proof is the concept of random restrictions.
**Theorem ([FSS81], [Ajt83])**

Let $\oplus$ be the parity function. That is, for every $x \in \{0, 1\}^n$,
$$\oplus(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \pmod{2}.$$ 
Then, $\oplus \not\in AC^0$.

**Proof Sketch**

The main tool for the proof is the concept of *random restrictions*. We first assume the existence of a constant depth circuit computing parity. Then choose a large number ($n - n^\epsilon$ for some constant $0 < \epsilon < 1$) of the variables and randomly assign them to 0 or 1.
Main Theorem

Theorem ([FSS81], [Ajt83])

Let $\oplus$ be the parity function. That is, for every $x \in \{0, 1\}^n$,
$$\oplus(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \pmod{2}.$$ 
Then, $\oplus \not\in AC^0$.

Proof Sketch

The main tool for the proof is the concept of random restrictions. We first assume the existence of a constant depth circuit computing parity. Then choose a large number ($n - n^\epsilon$ for some constant $0 < \epsilon < 1$) of the variables and randomly assign them to 0 or 1. We show that with positive probability, the restricted function is a constant, which is not possible for the parity function.
Random Restrictions and the Switching Lemma

Random Restriction

If $f$ is a function on $n$ variables and $\rho$ is a partial assignment (called a restriction) to the variables of $f$, then we denote by $f|_\rho$ the restriction of $f$ under $\rho$. 
Random Restrictions and the Switching Lemma

Random Restriction

If \( f \) is a function on \( n \) variables and \( \rho \) is a partial assignment (called a restriction) to the variables of \( f \), then we denote by \( f|_{\rho} \) the restriction of \( f \) under \( \rho \).

Håstad’s Switching Lemma [Hås86]

Suppose \( f \) is expressible as a \( k \)-DNF, and let \( \rho \) denote a random restriction that assigns random values to \( t \) randomly selected input bits. Then for every \( s \geq 2 \),

\[
Pr_{\rho} \left[ f|_{\rho} \text{ is not expressible as } s\text{-CNF} \right] \leq \left( \frac{(n-t)k^{10}}{n} \right)^{s/2}
\]
Proof of $\oplus \notin AC^0$

We assume an $AC^0$ circuit computing parity. The circuit can be simplified as follows:

1. All fan-outs are 1, i.e., the circuit is a tree.
2. All $\neg$ gates are at the input level, or equivalently the input has $2^n$ variables, with the last $n$ being the negative of the first $n$.
3. The $\lor$ and $\land$ gates alternate (each level of the tree has either only $\lor$ gates or only $\land$ gates).
4. The bottom level (above input) has $\land$ gates of fan-in 1.
Proof of $\oplus \notin AC^0$

We assume an $AC^0$ circuit computing parity. The circuit can be simplified as follows:

1. All fan-outs are 1, i.e., the circuit is a tree.
Proof of $\oplus \not\in AC^0$

We assume an $AC^0$ circuit computing parity. The circuit can be simplified as follows:

1. All fan-outs are 1, i.e., the circuit is a tree.
2. All $\neg$ gates are at the input level, or equivalently the input has $2n$ variables, with the last $n$ being the negative of the first $n$. 
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

We assume an $AC^0$ circuit computing parity. The circuit can be simplified as follows:

1. All fan-outs are 1, i.e., the circuit is a tree.
2. All $\neg$ gates are at the input level, or equivalently the input has $2n$ variables, with the last $n$ being the negative of the first $n$.
3. The $\lor$ and $\land$ gates alternate (each level of the tree has either only $\lor$ gates or only $\land$ gates).
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

We assume an $AC^0$ circuit computing parity. The circuit can be simplified as follows:

1. All fan-outs are 1, i.e., the circuit is a tree.
2. All $\neg$ gates are at the input level, or equivalently the input has $2n$ variables, with the last $n$ being the negative of the first $n$.
3. The $\lor$ and $\land$ gates alternate (each level of the tree has either only $\lor$ gates or only $\land$ gates).
4. The bottom level (above input) has $\land$ gates of fan-in 1.
Proof of $\oplus \not\in AC^0$

Let $n^b$ be the upper bound on the number of gates in the circuit as described above with depth at most $d$ (for constants $b$ and $d$).
Proof of $\oplus \not\in \mathsf{AC}^0$

Let $n^b$ be the upper bound on the number of gates in the circuit as described above with depth at most $d$ (for constants $b$ and $d$). We’ll do a series of random restrictions, reducing the depth of the circuit by 1 and keeping the fan-in constant at the bottom level in each step (with high probability).
Proof of $\oplus \not\in AC^0$

Let $n^b$ be the upper bound on the number of gates in the circuit as described above with depth at most $d$ (for constants $b$ and $d$). We’ll do a series of random restrictions, reducing the depth of the circuit by 1 and keeping the fan-in constant at the bottom level in each step (with high probability).

Let $n_i$ denote the number of unrestricted variables after step $i$. In step $i + 1$, we restrict $n_i - \sqrt{n_i}$ more variables randomly. Since $n_0 = n$, we have that $n_i = n^{1/2^i}$.
Proof of \( \oplus \not\in AC^0 \)

Let \( n^b \) be the upper bound on the number of gates in the circuit as described above with depth at most \( d \) (for constants \( b \) and \( d \)). We’ll do a series of random restrictions, reducing the depth of the circuit by 1 and keeping the fan-in constant at the bottom level in each step (with high probability).

Let \( n_i \) denote the number of unrestricted variables after step \( i \). In step \( i + 1 \), we restrict \( n_i - \sqrt{n_i} \) more variables randomly. Since \( n_0 = n \), we have that \( n_i = n^{1/2^i} \).

Also, let \( k^i = 10b2^i \). We’ll show that with high probability, after the step \( i \), we’re left with a depth \( d - i \) circuit with fan-in at most \( k_i \) at the bottom level.
Proof of $\oplus \not\in AC^0$

Suppose that the bottom level has $\land$ gates and the one above it has $\lor$ gates. The function computed by each $\lor$ gate is a $k_i$-DNF.
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

Suppose that the bottom level has $\land$ gates and the one above it has $\lor$ gates. The function computed by each $\lor$ gate is a $k_i$-DNF. By the Switching Lemma, the probability that such a function is expressible as a $k_{i+1}$-CNF is at least $1 - \left(\frac{k_i^{10}}{n^{1/2i+1}}\right)^{k_{i+1}/2}$, which is at least $1 - 1/(10n^b)$ for large enough $n$. 
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

Suppose that the bottom level has $\land$ gates and the one above it has $\lor$ gates. The function computed by each $\lor$ gate is a $k_i$-DNF. By the Switching Lemma, the probability that such a function is expressible as a $k_{i+1}$-CNF is at least $1 - \left(\frac{k_i^{10}}{n^{1/2^{i+1}}}\right)^{k_{i+1}/2}$, which is at least $1 - 1/(10n^b)$ for large enough $n$.

Now we can merge two layers (both with $\land$ gates), reducing the depth by 1, and keeping a fan-in of $k_{i+1}$ at the bottom layer.
Proof of $\oplus \not\in AC^0$

Suppose that the bottom level has $\land$ gates and the one above it has $\lor$ gates. The function computed by each $\lor$ gate is a $k_i$-DNF. By the Switching Lemma, the probability that such a function is expressible as a $k_{i+1}$-CNF is at least $1 - \left( \frac{k_i^{10}}{n^{1/2^{i+1}}} \right)^{k_{i+1}/2}$, which is at least $1 - 1/(10n^b)$ for large enough $n$.

Now we can merge two layers (both with $\land$ gates), reducing the depth by 1, and keeping a fan-in of $k_{i+1}$ at the bottom layer.

A symmetric reasoning applies if the bottom level has $\lor$ gates.
Proof of $\oplus \not\in AC^0$

Suppose that the bottom level has $\wedge$ gates and the one above it has $\vee$ gates. The function computed by each $\vee$ gate is a $k_i$-DNF. By the Switching Lemma, the probability that such a function is expressible as a $k_i+1$-CNF is at least $1 - \left( \frac{k_i^{10}}{n^{1/2+i+1}} \right)^{k_i+1/2}$, which is at least $1 - 1/(10n^b)$ for large enough $n$.

Now we can merge two layers (both with $\wedge$ gates), reducing the depth by 1, and keeping a fan-in of $k_i+1$ at the bottom layer.

A symmetric reasoning applies if the bottom level has $\vee$ gates.

We continue this switch and merge process for $d - 2$ steps. Note that we apply the Lemma at most once per each of the $n^b$ gates of the circuit. By union bound, we succeed with probability at least $9/10$. 

H. Gupta and K. Mittal (IIT Bombay)  
Lower Bounds in Boolean Circuits  
2 October 2018 11 / 20
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

Finally we are left with a depth 2 circuit with fan-in $k = k_{d-2}$. That is, either a $k$-CNF or a $k$-DNF formula. But, such a formula can be made a constant by fixing at most $k$ of the variables ($k$ is a constant here).
Proof of Main Theorem

Proof of $\oplus \not\in AC^0$

Finally we are left with a depth 2 circuit with fan-in $k = k_{d-2}$. That is, either a $k$-CNF or a $k$-DNF formula. But, such a formula can be made a constant by fixing at most $k$ of the variables ($k$ is a constant here).

Since the parity function is not constant under any restriction of less than $n$ variables, we have a contradiction.
Håstad’s Switching Lemma [Hås86]

Suppose $f$ is expressible as a $k$-DNF, and let $\rho$ denote a random restriction that assigns random values to $t$ randomly selected input bits. Then for every $s \geq 2$,

$$\Pr_{\rho}[f|\rho \text{ is not expressible as } s\text{-CNF}] \leq \left(\frac{(n-t)k^{10}}{n}\right)^{s/2}$$
Proof of the Switching Lemma

Proof of Lemma

Let $R_t$ be the set of all restrictions on $t$ variables. Then $|R_t| = \binom{n}{t}2^t$. 

H. Gupta and K. Mittal (IIT Bombay)  
Lower Bounds in Boolean Circuits  
2 October 2018  
14 / 20
Proof of Lemma

Let \( R_t \) be the set of all restrictions on \( t \) variables. Then \( |R_t| = \binom{n}{t} 2^t \).

Denote \( B \) as the set of bad restrictions, i.e. \( B = \{ \rho \in R_t \text{ such that } f|_{\rho} \text{ is not expressible as an } s\text{-CNF} \} \).
Proof of the Switching Lemma

Proof of Lemma

Let $R_t$ be the set of all restrictions on $t$ variables. Then $|R_t| = \binom{n}{t}2^t$. Denote $B$ as the set of bad restrictions, i.e. $B = \{\rho \in R_t$ such that $f|_\rho$ is not expressible as an $s$-CNF\}. To prove the Lemma, we show that $B$ is small.
Proof of Lemma

Let $R_t$ be the set of all restrictions on $t$ variables. Then $|R_t| = \binom{n}{t}2^t$. Denote $B$ as the set of bad restrictions, i.e. $B = \{\rho \in R_t \text{ such that } f|_{\rho} \text{ is not expressible as an } s\text{-CNF}\}$. To prove the Lemma, we show that $B$ is small. This is done by constructing a one-one mapping from $B$ to $R_{t+s} \times \{0, 1\}^\ell$ for some $\ell = O(s \log k)$. This gives us

$$\frac{|B|}{|R_t|} \leq \frac{|R_{t+s} \times \{0, 1\}^\ell|}{|R_t|} = \frac{\binom{n}{t+s}2^{t+s}2^{O(s \log k)}}{\binom{n}{t}2^t} = \frac{\binom{n}{t+s}k^{O(s)}}{\binom{n}{t}}$$
Proof of the Switching Lemma

Proof of Lemma

Before we continue, we give a few definitions:

1. Let a *min-term* of a function $f$ be a partial assignment to $f$’s variables that makes $f$ output 1 irrespective of the assignments to the remaining variables. Thus every clause in a $k$-DNF formula for $f$ yields a size-$k$ min-term of $f$.
Proof of the Switching Lemma

Proof of Lemma

Before we continue, we give a few definitions:

1. Let a \textit{min-term} of a function $f$ be a partial assignment to $f$’s variables that makes $f$ output 1 irrespective of the assignments to the remaining variables. Thus every clause in a $k$-DNF formula for $f$ yields a size-$k$ min-term of $f$.

2. A \textit{max-term} is a partial assignment to $f$’s variables that makes $f$ output 0 regardless of the other variables. Thus every clause in a $k$-CNF formula for $f$ yields a size-$k$ max-term of $f$. 
Before we continue, we give a few definitions:

1. Let a *min-term* of a function $f$ be a partial assignment to $f$’s variables that makes $f$ output 1 irrespective of the assignments to the remaining variables. Thus every clause in a $k$-DNF formula for $f$ yields a size-$k$ min-term of $f$.

2. A *max-term* is a partial assignment to $f$’s variables that makes $f$ output 0 regardless of the other variables. Thus every clause in a $k$-CNF formula for $f$ yields a size-$k$ max-term of $f$.

We’ll assume that all min/max terms are minimal, i.e., no subset of them is a min/max term respectively.
Proof of the Switching Lemma

Proof of Lemma

We note that a function that is not expressible as an $s$-CNF must have at-least one max-term of length at-least $s + 1$. 
Proof of Lemma

We note that a function that is not expressible as an $s$-CNF must have at-least one max-term of length at-least $s + 1$.

This is seen from the fact that for any function $f$,

$$\neg f = \bigvee \text{ (clause corresponding to max-term of } f \text{)}$$
Proof of Lemma

We note that a function that is not expressible as an $s$-CNF must have at-least one max-term of length at-least $s + 1$.

This is seen from the fact that for any function $f$,

$$\neg f = \bigvee \text{(clause corresponding to max-term of } f\text{)}$$

Further Proof on Board!
Lower Bounds for Circuits with Counters: \( \text{ACC}0 \)
An exponentially stronger circuit class than \( \text{AC}^0 \).

Benjamin Rossman [Ros18]

**The Average Sensitivity of Bounded-Depth Formulas**
Computational Complexity 2018
Thank You!
M. Ajtai.
Σ₁¹ formulae on finite structure.

Merrick L. Furst, James B. Saxe, and Michael Sipser.
Parity, circuits, and the polynomial-time hierarchy.

Johan Håstad.
Almost optimal lower bounds for small depth circuits.
R. Karp and R. Lipton.
Turing machines that take advice.

Benjamin Rossman.
The average sensitivity of bounded-depth formulas.

C. E. Shannon.
The synthesis of two-terminal switching circuits.