Outline

Lecture 1: Overview of Circ LBs from Algorithms
Lecture 2-3: More on Circ LBs from Algorithms
Lecture 3: The Mysteries of the Missing String
Lecture 4: The Power of Constructing Bad Inputs
How to Prove Lower Bounds With Algorithms

Lecture 2: Overview of Circuit Lower Bounds From Circuit-Analysis Algorithms
Picking Up From Last Time

Let $\mathbb{C}$ be some circuit class (like $\text{ACC}^0$).

**Thm A [MW’18]:**

If for some $\varepsilon > 0$, Gap-$\mathbb{C}$-SAT on $2^{n^\varepsilon}$ size is in $O(2^{n-n^\varepsilon})$ time, then Quasi-NP does not have poly-size $\mathbb{C}$-circuits.

**Idea.** Show that if we assume both:

1. Quasi-NP has poly-size $\mathbb{C}$-circuits, and
2. a faster $\mathbb{C}$-SAT algorithm

Then show $\exists k \text{NTIME}[n^{\log kn}] \subseteq \text{NTIME}[o(n^{\log kn})]$

Contradicts the nondeterministic time hierarchy:

there is $L_{\text{hard}}$ in $\text{NTIME}[n^{\log kn}] \setminus \text{NTIME}[o(n^{\log kn})]$
Proof Ideas of Theorem A

Idea. Assume:

(1) Quasi-NP has poly-size \( \mathbb{C} \)-circuits, and

(2) a faster \( \mathbb{C} \)-SAT algorithm

Then show \( \exists k \ \text{NTIME}[n^{\log^k n}] \subseteq \text{NTIME}[o(n^{\log^k n})] \)

Take an \( L \) in \text{non-deterministic} \( n^{\log^k n} \) time. Given an input \( x \), we decide if \( x \in L \), by:

(A) Guessing some witness \( y \) of \( O(n^{\log^k n}) \) length.

(B) Checking \( y \) is a witness for \( x \) in \( O(n^{\log^k n}) \) time.
Proof Ideas of Theorem A

Idea. Assume:

(1) Quasi-NP has poly-size \( \mathbb{C} \)-circuits, and
(2) a faster \( \mathbb{C} \)-SAT algorithm

Then show \( \exists k \ \text{NTIME}[n^{\log^k n}] \subseteq \text{NTIME}[o(n^{\log^k n})] \)

Take an \( L \) in \text{nondeterministic} \( n^{\log^k n} \) time. Given an input \( x \), we \textbf{will} decide if \( x \in L \), by:

(A) Guessing some witness \( y \) of \( o(n^{\log^k n}) \) length.
(B) Checking \( y \) is a witness for \( x \) in \( o(n^{\log^k n}) \) time.
Proof Ideas of Theorem A

Idea. Assume:

(1) Quasi-NP has poly-size \( \mathbb{C} \)-circuits, and
(2) a faster \( \mathbb{C} \)-SAT algorithm

Then show \( \exists k \text{ NTIME}[n^{\log^k n}] \subseteq \text{NTIME}[o(n^{\log^k n})] \)

Take an \( L \) in nondeterministic \( n^{\log^k n} \) time.
Given an input \( x \), we \textbf{will} decide if \( x \in L \), by:

(A) Guessing some witness \( y \) of \( o(n^{\log^k n}) \) length.

(B) Checking \( y \) is a witness for \( x \) in \( o(n^{\log^k n}) \) time.
Guessing Short Witnesses

Easy Witness Lemma [IKW’02, MW’18]:
If $\text{NEXP (Quasi-NP)}$ has polynomial-size circuits, then all $\text{NEXP (Quasi-NP)}$ problems have “easy witnesses”

Def. An $\text{NEXP/Quasi-NP}$ problem $L$ has easy witnesses if
$\forall$ Verifiers $V$ for $L$ and $\forall x \in L$,
$\exists$ poly$(|x|)$-size circuit $D_x$ such that $V(x, Y)$ accepts,
where $Y = \text{Truth Table of circuit } D_x$.

1. Guess a witness $y$ of $o(n^{\log^k n})$ length.

1’. Guess $\text{poly}(n)$-size circuit $D_x$
Verifying Short Witnesses

2. Check $y$ is a witness for $x$ in $o(n^{\log^k n})$ time.

Assuming Quasi-NP has polynomial-size circuits, “easy witnesses” exist for every verifier $V$.

We choose a verifier $V$ for $L \in NTIME[n^{\log^k n}]$ so that:

- Checking $V(x, y)$ accepts for $|x| = n$ is equivalent to
  - Solving UNSAT on a $\mathbb{C}$-circuit with $2^{m^\varepsilon}$ size and $m = \log^{k+1}(n) + 4\log(n)$ inputs

Then, $2^{m-m^\varepsilon}$ time for $\mathbb{C}$-UNSAT $\Rightarrow o(n^{\log^k n})$ time to decide $L$. 
Verifying Short Witnesses

2. Check $y$ is a witness for $x$ in $o(n^\log^k n)$ time.

Assuming Quasi-NP has polynomial-size circuits, “easy witnesses” exist for every verifier $V$.

We can also choose a verifier $V$ for $L \in NTIME[n^\log^k n]$ so that:

Checking $V(x, y)$ accepts $\iff$

Distinguishing unsatisfiable circuits from circuits with many satisfying assignments (Uses a version of the PCP Theorem!)

Then, $2^{n-n^\varepsilon}$ time for Gap-$\mathbb{C}$-UNSAT $\Rightarrow o\left(n^\log^k n\right)$ time to decide $L$.
Now: Time for Details
Definition: ACC Circuit Family

An **ACC circuit family** \( \{ C_n \} \) has the properties:

- Every \( C_n \) takes \( n \) bits of input and outputs one bit
- There is a fixed \( d \) such that every \( C_n \) has depth at most \( d \)
- There is a fixed \( m \) such that the gates of \( C_n \) are \( \text{AND, OR, NOT, MOD}_{m} \) (unbounded fan-in)

\[
\text{MOD}_{m}(x_1, \ldots, x_t) = 1 \iff \sum_i x_i \text{ is divisible by } m
\]

**Remarks**

1. The default size (#gates) of \( C_n \) is **polynomial in** \( n \)
2. **Strength:** this is a **non-uniform** model of computation (can compute some undecidable languages!)
3. **Weakness:** ACC circuits can be efficiently simulated by **constant-layer neural networks (a.k.a. TC0)**
Definition: ACC Circuit Family

An **ACC circuit family** \( \{ C_n \} \) has the properties:

- Every \( C_n \) takes \( n \) bits of input and outputs one bit
- There is a fixed \( d \) such that every \( C_n \) has depth at most \( d \)
- There is a fixed \( m \) such that the gates of \( C_n \) are AND, OR, NOT, MOD\( m \) (unbounded fan-in)

\[
\text{MOD}_m(x_1, \ldots, x_t) = 1 \text{ iff } \sum_i x_i \text{ is divisible by } m
\]

Example:

Note: These circuits become very complex, already for certain fixed \( d \) and \( m \).

**OPEN:** Does every problem in EXP have polynomial-size MOD6 circuits of depth 3 (?!)

ACC does have some surprising power:

[CW’22] For every \( \epsilon > 0 \), every symmetric Boolean fn has \( 2^{n\epsilon} \) size depth-3 ACC circuits
Where does ACC come from?

**Dream of the 1980s:** Prove $P \neq NP$ by proving $NP \not\subset P/poly$.

Unlike Turing Machines, logic circuits are fixed, “simple” devices. This should make it easier to prove impossibility results.

**Ajtai, Furst-Saxe-Sipser, Håstad (early 80’s)**

**MOD2 $\not\in$ AC0** [poly-size ACC with only AND, OR, NOT, no MODm]

**Razborov, Smolensky (late 80’s)**

**MOD3 $\not\in$ (AC0 with MOD2 gates)**

For $p \neq q$ prime, $\text{MOD}_p \not\in$ (AC0 with MOD$q$ gates)

**Barrington (late 80’s)** Suggested ACC as the next natural step

**Conjecture** Majority $\not\in$ ACC

**Conjecture (early 90’s)** NP $\not\in$ ACC

**Conjecture (late 90’s)** NEXP $\not\in$ ACC
**ACC Lower Bounds**

\[ \text{EXP}^{\text{NP}} = \text{Exponential Time with an NP oracle} \quad \text{[think: SAT oracle]} \]

\[ \text{NEXP} = \text{Nondeterministic Exponential Time} \]

**Theorem [W’11]** There is an \( f \in \text{EXP}^{\text{NP}} \) such that for every \( d, m \) there is an \( \epsilon > 0 \) such that \( f \) does not have \( \text{ACC} \) circuits with \( \text{MOD}m \) gates, depth \( d \), and size \( 2^n^{\epsilon} \).

**Theorem [W’11]** There is an \( f \in \text{NEXP} \) such that for all \( d, m, k \), \( f \) does not have \( n^{\log^k n} \) size \( \text{ACC} \) circuits of depth \( d \) with \( \text{MOD}m \) gates.

**Remark** Compare with:

[MS 70’s] \( \text{EXP}^{(\text{NP}^{\text{NP}})} = \text{EXP}^{\Sigma_2 P} \) doesn’t have \( o(2^{n/n}) \) size circuits.

[K82] \( \text{NEXP}^{\text{NP}} = \Sigma_2 E \) doesn’t have \( n^{\log^k n} \)-size circuits for all \( k \).
ACC Lower Bounds

**Quasi-NP** = Nondeterministic $n^{polylog n}$ Time

**Theorem [MW’18]** There is an $f$ in **Quasi-NP** such that for all $d, m, k$, $f$ does not have $n^k$ size ACC circuits of depth $d$ with MOD$m$ gates

Has since been extended in multiple ways!
(stronger circuit classes, average-case hardness, etc etc)

We’ll outline a different result, and then sketch how to extend it.

**Theorem** There is an $f$ in $E^{NP} = \text{TIME}^{\text{SAT}[2^{O(n)}]}$ such that for all $d, m$, there is an $\varepsilon > 0$ such that $f$ does not have $2^{n^\varepsilon}$ size ACC circuits of depth $d$ with MOD$m$ gates
Design a faster ACC-SAT algorithm

The Algorithm: For every $d, m$, there is an $\epsilon > 0$ such that ACC-SAT on circuits with $n$ inputs, $2^{n\epsilon}$ size, depth $d$, and MOD$m$ gates is solvable in $2^{n-n^\epsilon}$ time.

Show that faster ACC-SAT algorithms imply lower bounds against ACC

The LB Connection: If $\mathcal{C}$-SAT on circuits with $n$ inputs and $2^{n^\epsilon}$ size is in $O(2^n/n^{10})$ time, then $E^{NP}$ doesn’t have $2^{n^\epsilon}$ size $\mathcal{C}$-circuits.

This algorithm has changed little in the past 9 years...

The connections have strengthened considerably!
Algorithm for SAT on ACC Circuits

Ingredients:

1. **Old representation** [Yao’90, Beigel-Tarui’94, Green et al’95]
   For every ACC function \( f : \{0,1\}^* \to \{0,1\} \) and every \( n \), we can write \( f_n : \{0,1\}^n \to \{0,1\} \) as:
   \[
   f_n(x_1, \ldots, x_n) = g(h(x_1, \ldots, x_n)),
   \]
   where
   - \( h \) is a multilinear polynomial of at most \( K \) monomials,
     \( h(a) \in \{0, \ldots, K\} \) for all \( a \in \{0,1\}^n \)
   - \( K \) is not “too large” (quasi-polynomial in circuit size)
   - \( g : \{0, \ldots, K\} \to \{0,1\} \) is a fixed “simple” function

2. **“Fast Fourier Transform” for multilinear polynomials:**
   Given a multilinear polynomial \( h \) in its coefficient representation, the value \( h(a) \) can be computed over all points \( a \in \{0,1\}^n \) in \( 2^n \, \text{poly}(n) \) time.

\[ K \leq 2^{O(\log s \, \text{poly}(d \, r))} \]
where \( s = \text{size}, \ d = \text{depth}, \ r = \# \text{ prime divisors of } m \)
Fast Multipoint Evaluation

**Theorem:** Given the $2^n$ coefficients of a multilinear polynomial $h$ in $n$ variables, $h(a)$ can be computed on all points $a \in \{0, 1\}^n$ in $2^n \text{poly}(n)$ time.

Can write: $h(x_1, \ldots, x_n) = x_1 h_1(x_2, \ldots, x_n) + h_2(x_2, \ldots, x_n)$

Want a $2^n$ table $T$ that contains the value of $h$ on all $2^n$ points.

**Algorithm:** If $n = 1$ then return $T = [h(0), h(1)]$

Recursively compute the $2^{n-1}$-length table $T_1$ for the values of $h_1$, and the $2^{n-1}$-length table $T_2$ for the values of $h_2$

Return the table $T = (T_2)(T_1 + T_2)$ of $2^n$ entries

Running time has the recurrence $R(2^n) \leq 2 \cdot R(2^{n-1}) + 2^n \text{poly}(n)$

**Corollary:** We can evaluate $g$ of $h$ on all $a \in \{0, 1\}^n$, in only $2^n \text{poly}(n)$ time
Theorem: For every $d, m$, there is an $\varepsilon > 0$ such that ACC-SAT on circuits with $n$ inputs, $2^{n\varepsilon}$ size, depth $d$, and MOD$m$ gates is solvable in $2^{n-n^\varepsilon}$ time.

Proof:

- Take an OR of all assignments to the first $n^\varepsilon$ inputs of $C$.
- For small $\varepsilon > 0$, $h$ is evaluated on all $2^n - n^\varepsilon$ assignments in $2^{n-n^\varepsilon}$ poly(n) time.

Output “SAT” $\iff \exists a \in \{0, 1\}^{n-n^\varepsilon}$ s.t. $g(h(a)) = 1$. 

Beigel and Tarui

Fast Multipoint Eval
The LB Connection: If ACC-SAT on circuits with \(n\) inputs and \(2^{n^\epsilon}\) size is in \(O(2^n/n^{10})\) time, then \(E^{NP}\) doesn’t have \(2^{n^\epsilon}\) size ACC-circuits.

Given circuit \(C: \{0, 1\}^n \rightarrow \{0, 1\}\), let \(tt(C)\) be its truth table: the output of \(C\) on all \(2^n\) assignments, in lexicographical order

**Succinct 3SAT:** Given a circuit \(C\), does \(tt(C)\) encode a satisfiable 3CNF?

Key Idea: Succinct 3SAT is NEXP-complete, in a very strong way...

**Lemma 1** Succinct 3SAT for AC0 circuits of \(n\) inputs and \(n^{10}\) size is solvable in nondeterministic \(2^n \ poly(n)\) time but not in nondeterministic \(\frac{2^n}{n^5}\) time.

Upper bound: Evaluate the AC0 circuit on all \(2^n\) inputs, get a \(2^n\)-length 3CNF instance, guess and check a SAT assignment, in \(2^n \ poly(n)\) time

Lower bound: [JMV’13] Every \(L \in NTIME[2^n]\) can be reduced in poly-time to a Succinct 3SAT instance which is AC0, \(m = n + 4\log(n)\) inputs, \(n^{10}\) size

So, if Succinct3SAT is in \(2^m/m^5\) time, then \(L\) can be decided in time \(o(2^n)\)

Contradicts the nondeterministic time hierarchy theorem!
The LB Connection: If $E^{NP}$ has $2^{n^\epsilon}$ size ACC-circuits and ACC-SAT on circuits with $n$ inputs and $2^{n^\epsilon}$ size is in $O(2^n/n^{10})$ time, then contradiction.

**Succinct 3SAT:** Given a circuit $C$, does $tt(C)$ encode a satisfiable 3CNF?

**Lemma 1** Succinct 3SAT for ACC circuits of $n$ inputs and $n^{10}$ size is solvable in nondeterministic $2^n \text{poly}(n)$ time but not in nondeterministic $\frac{2^n}{n^{5}}$ time.

**Goal:** Use ACC circuits for $E^{NP}$ & the ACC-SAT algorithm, to solve Succinct 3SAT faster.

Say that **Succinct 3SAT has “succinct” SAT assignments** if for every $C$ (of $n$ inputs and $n^{10}$ size) such that $tt(C)$ encodes a satisfiable 3CNF $F$, there is an ACC circuit $D$ of $2^{n^{10\epsilon}}$ size such that $tt(D)$ encodes a variable assignment $A$ that satisfies $F$.

(Imagine $F$ has variables $x_1, ..., x_{2^n}$. Then $D(i)$ outputs a 0-1 assignment to variable $x_i$ in $F$)

*If a succinct SAT assignment exists, we only have to guess a witness of length $2^{n^{10\epsilon}}$*

**Lemma 2** If $E^{NP}$ has $2^{n^\epsilon}$ size ACC circuits then Succinct 3SAT has “succinct” SAT assignments.
**The LB Connection:** If $E^{NP}$ has $2^{n^e}$ size ACC-circuits and ACC-SAT on circuits with $n$ inputs and $2^{n^e}$ size is in $O(2^n/n^{10})$ time, then *contradiction*

**Succinct 3SAT:** Given a circuit $C$, does $tt(C)$ encode a satisfiable 3CNF?

**Lemma 2** If $E^{NP}$ has $2^{n^e}$ size ACC circuits then Succinct 3SAT has “succinct” SAT assignments

**Proof** The following is an $E^{NP}$ procedure:

*On input $(C, i)$, where $i \in \{1, \ldots, 2^n\}$, $C$ has $n$ inputs & $n^{10}$ size*

1. Compute $F = tt(C)$, think of $F$ as a 3CNF formula.
2. Use a SAT oracle and search-to-decision for SAT, to find the lexicographically first SAT assignment to $F$.
3. Output the $i$-th bit of this assignment.

$E^{NP}$ has $2^{n^e}$ size ACC circuits $\Rightarrow$ there is a $2^{|C|^e} \leq 2^{n^{10e}}$ size ACC circuit $D(C, i)$ which outputs the $i$-th bit of a satisfying assignment to $F = tt(C)$.

Now for any circuit $C'$ of $n^{10}$ size, define the circuit $E(i) := D(C', i)$

Then $E$ has $2^{n^{10e}}$ size, and the assignment $tt(E)$ satisfies $tt(C')$