Algorithms and Lower Bounds: Basic Connections

Lecture 4: NEXP not in ACC

Ryan Williams
Stanford University
Definition: ACC Circuits

An ACC circuit family \{C_n\} has the properties:
• Every \(C_n\) takes \(n\) bits of input and outputs a 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)

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

Remarks
1. The default size 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
Where does ACC come from?

Prove $P \not= NP$ by proving $NP \not\subset P/poly$.
The simple combinatorial nature of circuits should make it easier to prove impossibility results.

**Ajtai, Furst-Saxe-Sipser, Håstad (early 80’s)**
- $MOD2 \not\in AC0$ [i.e., $n^{O(1)}$ size ACC with *only* AND, OR, NOT]

**Razborov, Smolensky (late 80’s)**
- $MOD3 \not\in (AC0 \text{ with } MOD2 \text{ gates})$
- For $p \neq q$ prime, $MODp \not\in (AC0 \text{ with } MODq \text{ gates})$

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

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

\(\text{EXP}^{\text{NP}} = \text{Exponential Time with an NP oracle}\)

\(\text{NEXP} = \text{Nondeterministic Exponential Time}\)

**Theorem 1** There is a problem \(Q\) in \(\text{EXP}^{\text{NP}}\) such that for every \(d, m\) there is an \(\varepsilon > 0\) such that \(Q\) does not have \(\text{ACC}\) circuits with \(\text{MOD}_m\) gates, depth \(d\), and size \(2^{n\varepsilon}\)

**Theorem 2** There is a problem \(Q\) in \(\text{NEXP}\) such that \(Q\) does not have \(n^{\text{poly}(\log n)}\) size \(\text{ACC}\) circuits of any constant depth

**Remark** Compare with the following:

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

[K82] \(\text{NEXP}^{\text{NP}} \not\subset \text{SIZE}(n^{\text{poly}(\log n)})\)

[BFT’98] \(\text{MA-EXP} \not\subset \text{SIZE}(n^{\text{poly}(\log n)})\)
Proof Strategy for ACC Lower Bounds

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

**Theorem** (Example)
If ACC-SAT with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time (for all constant depths and moduli), then EXP$^NP$ doesn’t have $2^{n^{o(1)}}$ size ACC circuits.

2. Design faster ACC-SAT algorithms!

**Theorem** For all $d, m$ there’s an $\epsilon > 0$ such that ACC-SAT on circuits with n inputs, depth $d$, MOD$m$ gates, and $2^{n^\epsilon}$ size can be solved in $2^n - \Omega(n^\epsilon)$ time
Theorem  If ACC-SAT on circuits with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then EXP$^{NP}$ doesn’t have $2^{n^{o(1)}}$ size ACC circuits.

Proof Idea  Show that if both:

• ACC-SAT with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time
• EXP$^{NP}$ has $2^{n^{o(1)}}$ size ACC circuits

then $\text{NTIME}[2^n] \subseteq \text{NTIME}[o(2^n)]$ (a contradiction)

Work with a “compressed” version of the 3SAT problem:

Exponentially long formulas are encoded with polynomial-size circuits
**Theorem** If ACC-SAT on circuits with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then $\text{EXP}^{\text{NP}}$ isn’t in $2^{n^{o(1)}}$ size ACC.

For a circuit $C : \{0,1\}^n \rightarrow \{0,1\}$, let $\text{tt}(C)$ be its truth table: the output of $C$ on all $2^n$ assignments, in lex. order

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

**Theorem [GW, PY ’80s]** Succinct 3SAT is $\text{NEXP}$-complete.

Succinct 3SAT is in $\text{NEXP}$: evaluate circuit $C$ on all possible assignments, and solve the resulting 3SAT instance

Succinct 3SAT is $\text{NEXP}$-hard. Follows from:

“For all $L \in \text{NP}$, there’s a $\text{TIME}[\text{poly}(\log n)]$ reduction from $L$ to 3SAT”

**Padding** ⇒ “For all $L \in \text{NEXP}$, there is a $\text{TIME}[\text{poly}(n)]$ reduction from $L$ to exponentially-long 3SAT”

The $\text{TIME}[\text{poly}(n)]$ reduction can be described with a circuit!
**Theorem**  If ACC-SAT on circuits with $n$ inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then $\exp^{NP}$ isn’t in $2^{n^{o(1)}}$ size ACC.

For a 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 lex. order

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

**Lemma 1** [..., JMV’15] For all $L \in \text{NTIME}[2^n]$, there is a polytime reduction $R_L$ from $L$ to Succinct 3SAT such that:
- $x \in L \Leftrightarrow R_L(x) = C_x$ encodes a satisfiable 3CNF formula
- $C_x$ is ACC, has size $n^{10}$, and $n + 4 \log n$ inputs, where $n = |x|$

**Corollary** Succinct 3SAT for ACC circuits of $n$ inputs & $n^{10}$ size is in nondet $2^n \ poly(n)$ time but not in nondet $\frac{2^n}{n^5}$ time.

(Otherwise, we’d contradict the nondet. time hierarchy!)
**Theorem** If ACC-SAT on circuits with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then $\text{EXP}^{\text{NP}}$ isn’t in $2^{n^{o(1)}}$ size ACC.

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

Say that **Succinct 3SAT has ACC satisfying assignments** if for every $C$ such that $\tt(C)$ is a satisfiable 3CNF, there is an ACC circuit $D$ of $2^{|C|^{o(1)}}$ size such that $\tt(D)$ is a variable assignment that satisfies $\tt(C)$.

**Succinct 3SAT has ACC satisfying assignments**
≡ “All satisfiable formulas which are compressible have a satisfying assignment which is somewhat compressible”

**Lemma 2** If $\text{EXP}^{\text{NP}}$ has $2^{n^{o(1)}}$ size ACC circuits then **Succinct 3SAT has ACC satisfying assignments**
**Theorem** If ACC-SAT on circuits with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then $\text{EXP}^{\text{NP}}$ isn’t in $2^{n^{o(1)}}$ size ACC.

**Succinct 3SAT:** *Given a circuit $C$, is $\text{tt}(C)$ a satisfiable 3CNF?*

**Lemma 2** If $\text{EXP}^{\text{NP}}$ has $2^{n^{o(1)}}$ size ACC circuits then Succinct 3SAT has ACC satisfying assignments

**Proof** The following can be computed in $\text{EXP}^{\text{NP}}$:

*On input $(C, i)$, use an NP oracle and binary search to find the lexicographically first satisfying assignment to $\text{tt}(C)$. Output the $i$-th bit of this assignment.*

**By assumption:** there is a $2^{|C|^{o(1)}}$ size ACC circuit $D(C, i)$ which outputs the $i$-th bit of a satisfying assignment to $\text{tt}(C)$.

Now for any circuit $C'$, define the circuit $E(i) := D(C', i)$ Then $E$ has $2^{|C|^{o(1)}}$ size, and $\text{tt}(E)$ satisfies $\text{tt}(C')$
**Theorem**  If ACC-SAT on circuits with $n$ inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time, then $\text{EXP}^{\text{NP}}$ isn’t in $2^{n^{o(1)}}$ size ACC.

**An overview:**

Assume “fast” ACC-SAT and small ACC circuits for $\text{EXP}^{\text{NP}}$  
Use to solve Succinct3SAT in $\text{NTIME}[2^n/n^5]$  
(contradiction!)

Outline of Succinct3SAT algorithm:

**Given a Succinct3SAT instance** $C$ (an ACC circuit)

1. Guess a small ACC circuit $Y$ encoding a satisfying assignment for the exponentially-long 3CNF $\text{tt}(C)$  
   (which exists, by Lemma 2 and small circuits for $\text{EXP}^{\text{NP}}$)

2. Use “fast” Circuit-SAT algorithm to check that $\text{tt}(D)$ satisfies $\text{tt}(C)$ in $O(2^n/n^5)$ time
Given Succinct3SAT instance **C** (an ACC circuit of *n* inputs)

**Nondeterministically guess** ACC circuit **Y** of $2^{n^{o(1)}}$ size

**Y(j)** is intended to output the *j*-th bit of a satisfying assignment for **φ**

Construct the following circuit **D** of $2^{n^{o(1)}}$ size:

Output 1 iff the assignment encoded by **Y** does not satisfy the *i*-th clause of **φ**

Outputs assignments to the variables *a*, *b*, *c* of **φ**

Outputs the *i*-th clause of 3CNF **φ**

Using ACC-SAT algorithm: determine satisfiability of **D** in $o(2^n)$ time!
Proof Strategy for ACC Lower Bounds

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

**Theorem** (Example)
If ACC-SAT with n inputs and $2^{n^{o(1)}}$ size is in $O(2^n/n^{10})$ time (for all constant depths and moduli), then $\text{EXP}^{\text{NP}}$ doesn’t have $2^{n^{o(1)}}$ size ACC circuits.

2. Design faster ACC-SAT algorithms!

**Theorem** For all $d$, $m$ there’s an $\varepsilon > 0$ such that ACC-SAT on circuits with n inputs, depth $d$, MODm gates, and $2^{n^\varepsilon}$ size can be solved in $2^n - \Omega(n^\varepsilon)$ time
Ingredients for Solving ACC SAT

1. A known representation of ACC
   [Yao ’90, Beigel-Tarui’94] Every ACC function 
   \( f : \{0,1\}^n \rightarrow \{0,1\} \) can be expressed in the form
   \[
   f(x_1, \ldots, x_n) = g(h(x_1, \ldots, x_n))
   \]
   - \( h \) is a multilinear polynomial with \( K \) monomials,
     \( h(x_1, \ldots, x_n) \in \{0, \ldots, K\} \) for all \( (x_1, \ldots, x_n) \in \{0,1\}^n \)
   - \( K \) is not “too large” (quasipolynomial in circuit size)
   - \( g : \{0, \ldots, K\} \rightarrow \{0,1\} \) can be an arbitrary function

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

**Theorem** For all $d$, $m$ there’s an $\epsilon > 0$ such that $\text{ACC}[m]$ SAT with depth $d$, $n$ inputs, $2^{n^\epsilon}$ size can be solved in $2^n - \Omega(n^\epsilon)$ time.

**Proof:**
- $K = 2^{n^{O(\epsilon)}}$
- Take an OR of all assignments to the first $n^\epsilon$ inputs of $C$
- Beigel and Tarui
- Fast Fourier Transform
- For small $\epsilon > 0$, evaluate $h$ on all $2^{n-n^\epsilon}$ assignments in $2^{n-n^\epsilon}\text{poly}(n)$ time.
**Theorem** If ACC SAT with n inputs, \(n^{O(1)}\) size is in \(O(2^n/n^{10})\) time, then **NEXP doesn’t have** \(n^{O(1)}\) size ACC circuits.

Proceed just as with EXP\(^NP\), but use the following lemma:

**Lemma [IKW’02]** If NEXP \(\subseteq P/poly\) then Succinct 3SAT has poly-size circuits encoding satisfying assignments.

The proof applies work on “hardness versus randomness”

1. If EXP \(\subseteq P/poly\) then EXP = MA [BFNW93]

2. If Succinct 3SAT does *not* have polysize SAT assignment circuits, then in \(i.o.-NTIME[2^n]/n\) we can *guess a function with high circuit complexity and verify it* – *just guess a satisfying assignment to a hard Succinct3SAT instance*!

Can derandomize MA infinitely often with n bits of advice:

\[\text{EXP} = \text{MA} \subseteq \text{i.o.-NTIME}[2^n]/n \subseteq \text{i.o.-SIZE}(n^k)\]

(this is a contradiction)
**Theorem** If ACC SAT with n inputs, $n^{O(1)}$ size is in $O(2^n/n^{10})$ time, then **NEXP doesn’t have** $n^{O(1)}$ size ACC circuits.

Proceed just as with EXP$^\text{NP}$, but use the following lemma:

**Lemma [IKW’02]** If NEXP $\subseteq P/poly$ then Succinct 3SAT has poly-size circuits encoding satisfying assignments.

**Lemma** If P $\subseteq$ ACC then all poly-size *unrestricted* circuit families have equivalent poly-size ACC circuit families.

**Corollary** If NEXP $\subseteq$ ACC then Succinct 3SAT has poly-size ACC circuits encoding satisfying assignments.

This is all we need for the previous proof to go through. Also works for quasipolynomial size circuits.
Weak Derandomization Suffices

**Theorem 2** Suppose we are given a circuit C with n inputs, and are promised that it is either *unsatisfiable*, or at least $\frac{1}{2}$ of its assignments are satisfying. Determine which. If this is in $O(2^n/n^{10})$ time then $\text{NEXP} \not\subseteq \text{P/poly}$.

**Proof Idea:** Same as before, but replace the reduction from L to Succinct3SAT with a succinct PCP reduction.

**Lemma 3 [BGHSV’05]** For all $L \in \text{NTIME}(2^n)$,

there is a reduction $S_L$ from L to $\text{MAX CSP}$ such that:

- $x \in L \implies$ All constraints of $S_L(x)$ are satisfiable
- $x \notin L \implies$ At most $\frac{1}{2}$ of the constraints are satisfiable

1. $|S_L(x)| = 2^n \text{ poly}(n)$
2. The i-th constraint of $S_L(x)$ is computable in poly(n) time.
Remark on a Nice Property of ACC

Thm: Given an ACC circuit C of size $S$ and $n$ inputs, the truth table of C can be produced in $2^n \text{poly}(n) + 2^{\text{poly}(\log S)}$ time.

The main result of this lecture is that this property suffices to separate NEXP from ACC.

Morally, this property should be enough to get $\text{EXP} \not\subseteq \text{ACC}$

Observation: Let $L \in \text{TIME}[4^n] \setminus \text{TIME}[3^n]$. Then the truth table of $L \cap \{0,1\}^n$ cannot be produced in $o(3^n)$ time.

The non-uniformity of ACC prevents us from directly proving $\text{EXP}$ lower bounds. But perhaps $\text{NP} \neq \text{uniform-ACC}$

Q: Is there $L \in \text{TIME}[3^n]$ such that generating the $2^n$-length truth table of $L$ on $n$-bit inputs requires $\omega(3^n)$ time?
Future Progress

• Replace NEXP with simpler complexity classes
  May need to improve on exhaustive search for more complex problems

**Open Problem** *Does faster* COUNTING *of satisfying assignments for circuits imply stronger lower bounds?*

• Replace ACC with stronger circuits
  Design SAT algorithms for stronger circuits!
  Using PCP Theorem: can weaken the hypotheses

**Open Problem** *Can Boolean formulas of size $s$ be evaluated on all $n$-variable assignments in $\text{poly}(s) + 2^n \text{poly}(n)$ time?*

• Find more connections between algorithms and lower bounds!