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, MODm (unbounded fan-in)

$\mathrm{MODm}(x_1,\ldots,x_t) = 1$ iff $\sum_i x_i$ 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** $\neq$ **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 $\notin$ **AC0** [i.e., $n^{O(1)}$ size **ACC** with ***only*** AND, OR, NOT]

Razborov, Smolensky (late 80's)
MOD3 $\notin$ (**AC0** with **MOD2** gates)
For p $\neq$ q prime, **MODp** $\notin$ (**AC0** with **MODq** gates)

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

Conjecture Majority $\notin$ **ACC**
Conjecture (early 90's) **NP** $\not\subset$ **ACC**
Conjecture (late 90's) **NEXP** $\not\subset$ **ACC**

ACC Lower Bounds

EXP^{NP} = Exponential Time with an NP oracle
NEXP = Nondeterministic Exponential Time

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

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

Remark Compare with the following:

[MS 70's]	$EXP^{(NP^{NP})}$ doesn't have $o(2^n/n)$ size circuits
[K82]	$NEXP^{NP} \not\subset SIZE(n^{poly(\log n)})$
[BFT'98]	$MA\text{-}EXP \not\subset SIZE(n^{poly(\log n)})$

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 $\mathbf{2^{n^{o(1)}}}$ size is in **$O(2^n/n^{10})$ time** (for all constant depths and moduli), then **EXP^{NP}** doesn't have $\mathbf{2^{n^{o(1)}}}$ 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 $\mathbf{2^{n^{\varepsilon}}}$ size can be solved in **$2^{n - \Omega(n^{\varepsilon})}$ time**

SAT Algorithms Imply Lower Bounds

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 **NTIME$[2^n] \subseteq$ 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 **EXP^{NP} isn't in $2^{n^{o(1)}}$ size ACC**.

For a circuit **$C : \{0,1\}^n \to \{0,1\}$**, 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?***

Theorem [GW, PY '80s] Succinct 3SAT is **NEXP**-complete.

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

Succinct 3SAT is NEXP-hard. Follows from:

"For all L ∈ NP, there's a TIME[poly(log n)] reduction from L to 3SAT"

Padding ⇒ "For all L ∈ NEXP, there is a TIME[poly(n)] reduction from L to exponentially-long 3SAT"

The TIME[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 $\mathbf{EXP^{NP}}$ isn't in $2^{n^{o(1)}}$ size ACC.

For a circuit $\mathbf{C : \{0,1\}^n \rightarrow \{0,1\}}$, 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 $L \in \mathbf{NTIME[2^n]}$, 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 $\mathbf{n^{10}}$, and **n + 4 log n** inputs, where n = |x|

Corollary **Succinct 3SAT for ACC circuits of n inputs & $\mathbf{n^{10}}$ 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 $\mathbf{EXP^{NP}}$ 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 $\mathbf{2^{|C|^{o(1)}}}$ **size** such that

tt(D) is a variable assignment that satisfies **tt(C)**.

Succinct 3SAT has ACC satisfying assignments

$\equiv$ *"All satisfiable formulas which are compressible have a satisfying assignment which is somewhat compressible"*

Lemma 2 If $\mathbf{EXP^{NP}}$ 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 $\mathbf{EXP^{NP}}$ isn't in $2^{n^{o(1)}}$ size ACC.

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

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

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

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

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

Now for any circuit **C'**, define the circuit **E(i) := D(C', i)**
Then **E** has $\mathbf{2^{|C|^{o(1)}}}$ size, and **tt(E)** satisfies **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 $\mathbf{EXP^{NP}}$ isn't in $2^{n^{o(1)}}$ size ACC.

An overview:

Assume "fast" ACC-SAT and small ACC circuits for $\mathbf{EXP^{NP}}$

Use to solve Succinct3SAT in 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 tt(C)**

   (which exists, by Lemma 2 and small circuits for $\mathbf{EXP^{NP}}$)

2. **Use "fast" Circuit-SAT algorithm to check that tt(D) satisfies tt(C) in $O(2^n/n^5)$ time**

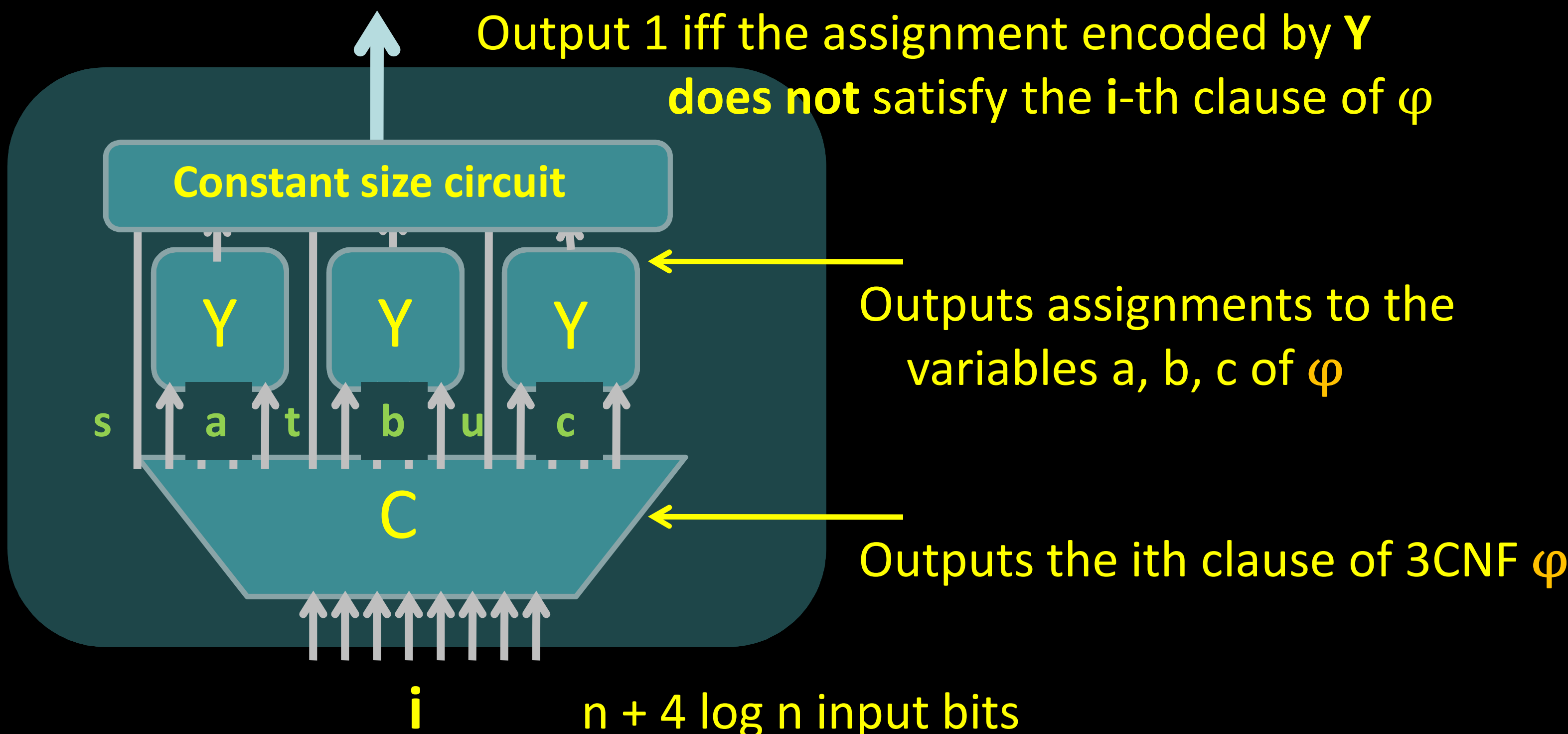
Fast Algorithm for Succinct3SAT

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:



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 $\mathbf{2^{n^{o(1)}}}$ size is in **$O(2^n/n^{10})$ time** (for all constant depths and moduli), then **$\mathrm{EXP}^{\mathrm{NP}}$** doesn't have $\mathbf{2^{n^{o(1)}}}$ 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 $\mathbf{2^{n^{\varepsilon}}}$ 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 **$f : \{0,1\}^n \to \{0,1\}$** 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)*
   - **$g : \{0,\ldots,K\} \to \{0,1\}$** can be an arbitrary function

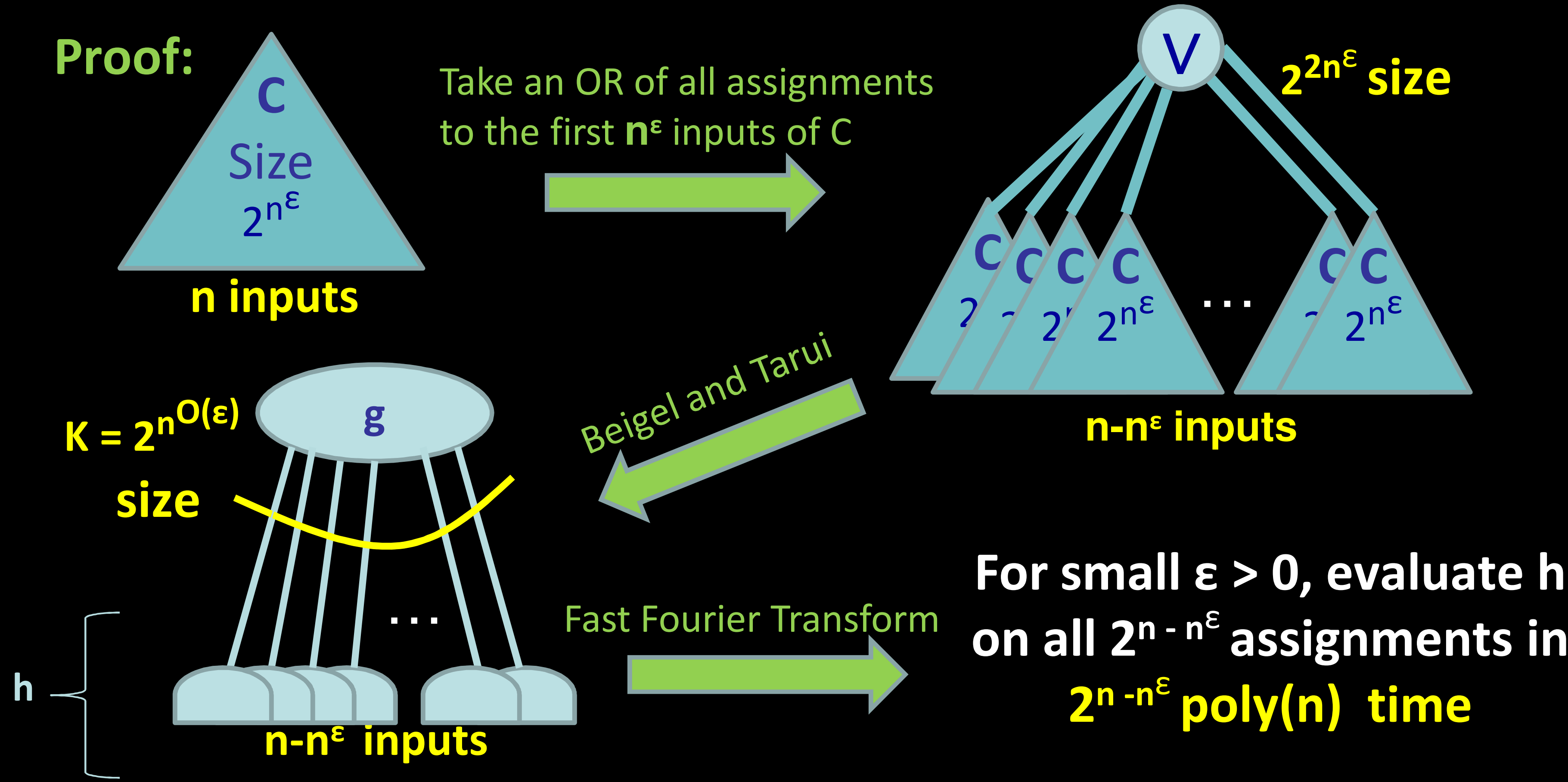
2. **"Fast Fourier Transform" for multilinear polynomials:**

   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 poly(n)** time.

ACC Satisfiability Algorithm

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

Proof:



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 $\subset$ P/poly** then Succinct 3SAT has poly-size circuits encoding satisfying assignments.

The proof applies work on "hardness versus randomness"

1. If **EXP $\subset$ 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:

$$\textbf{EXP = MA} \subseteq \textbf{io-NTIME}[2^n]/n \subseteq \textbf{io-SIZE}(n^k)$$

(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^{NP}, but use the following lemma:

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

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

Corollary If **NEXP** $\subset$ **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 ½ of its assignments are satisfying. Determine which.**
If this is in $O(2^n/n^{10})$ time then NEXP $\not\subseteq$ 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 $L \in \mathrm{NTIME}(2^n)$,

there is a reduction S_L from L to **MAX CSP** such that:

$x \in L \Rightarrow$ All constraints of $S_L(x)$ are satisfiable

$x \notin L \Rightarrow$ At most ½ of the constraints are satisfiable

1. $|S_L(x)| = 2^n$ **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 poly(n) + $2^{poly(\log S)}$ 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 EXP $\not\subset$ ACC

Observation: Let **L** $\in$ **TIME[4^n] \ 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 EXP lower bounds. But perhaps **NP $\neq$ uniform-ACC**

Q: Is there L $\in$ 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*** $\boldsymbol{poly(s) + 2^n\, poly(n)}$ ***time?***

- **Find more connections between algorithms and lower bounds!**