Recent Developments on Circuit Satisfiability Algorithms

Suguru TAMAKI  Kyoto University
After the **Fine-Grained Complexity and Algorithm Design** Program (Aug. 19 – Dec. 18, 2015), many papers on Circuit SAT algorithms have been published:

- Chan-Williams, ``Deterministic APSP, Orthogonal Vectors, and More: Quickly Derandomizing Razborov-Smolensky,’’ *SODA’16*
- Chen-Santhanam, ``Satisfiability on Mixed Instances,’’ *ITCS’16*
- Chen-Santhanam-Srinivasan, ``Average-Case Lower Bounds and Satisfiability Algorithms for Small Threshold Circuits,’’ *CCC’16*
- Williams, ``Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation,’’ *CCC’16*

Sakai-Seto-T-Teruyama, “Bounded Depth Circuits with Weighted Symmetric Gates: Satisfiability, Lower Bounds and Compression,” *MFCS’16*

Alman-Chan-Williams, “Polynomial Representations of Threshold Functions and Algorithmic Applications,” *FOCS’16*


T, “A Satisfiability Algorithm for Depth Two Circuits with a Sub-Quadratic Number of Symmetric and Threshold Gates”

......
Our Problem

Circuit Satisfiability (SAT)

**Input**  Boolean Circuit
\[ C : \{0,1\}^n \rightarrow \{0,1\} \]

**Output**  Yes/No
\[ \exists x \in \{0,1\}^n, C(x) = 1 \]
\[ \rightarrow \text{Yes} \]
\[ \forall x \in \{0,1\}^n, C(x) = 0 \]
\[ \rightarrow \text{No} \]
Our Problem

Circuit Satisfiability (SAT)

**Input**  Boolean Circuit

\[ C : \{0,1\}^n \rightarrow \{0,1\} \]

**Output**  Yes/No

\[ \exists x \in \{0,1\}^n, C(x) = 1 \rightarrow \text{Yes} \]

\[ \forall x \in \{0,1\}^n, C(x) = 0 \rightarrow \text{No} \]
Our Problem

Circuit Satisfiability (SAT)

- Canonical \textbf{NP-complete} problem [Cook-Levin]
- Solved in time $2^n \times \text{poly}(|C|)$
  ($n$: \#variables, $|C|$: size of a circuit $C$)
- \boldsymbol{\mathcal{C}}\text{-SAT} input only from a circuit class $\mathcal{C}$
  e.g. $\mathcal{C} = (k\text{-})\text{CNF}, \text{AC}^0, \text{AC}^0[p], \text{ACC}^0, \text{TC}^0, \text{NC}^1$ (Formula),...
Example

CNF-SAT

$$\mathcal{C} = (k-)$$CNF formulas (conjunctions of clauses)

e.g. $$\mathcal{C} = (x_1 \lor x_2) \land (\neg x_2 \lor \neg x_3 \lor x_4) \land (\neg x_3) \land \cdots$$

CNF is $$k$$-CNF if each clause has at most $$k$$ literals

Best known algorithms

3-CNF: $$1.308^n$$ [Hertli’14, randomized]

$$1.331^n$$ [Makino-T-Yamamoto’13, deterministic]

$$k$$-CNF: $$2^{(1 - \frac{1.64}{k})n}$$ [Paturi-Pudlak-Saks-Zane’05, randomized]

$$2^{(1 - \frac{1.44}{k})n}$$ [Moser-Scheder’11, deterministic]
**Hardness of Circuit SAT?**

(Strong) **Exponential Time Hypothesis**
(``quantitatively strengthening” of $P \neq NP$)

[Impagliazzo-Paturi-Zane]

**ETH** \[ \exists \delta > 0, \text{3-CNF-SAT} \notin \text{DTIME}[2^{\delta n}] \]

**SETH** \[ \forall \epsilon > 0, \exists k, \text{k-CNF-SAT} \notin \text{DTIME}[2^{(1-\epsilon)n}] \]

ETH & SETH are widely used to prove the optimality of exact, parametrized, approximation and dynamic algorithms

(cf. ``fine-grained complexity” theory)
General Research Goals

1. Design non-trivial algorithms for a stronger circuit class

Non-trivial:
- super-polynomially (or exponentially) faster than $2^n$

Stronger circuit class:
- $(k-)\text{CNF} \subset \text{AC}^0 \subset \text{AC}^0[p] \subset \text{ACC}^0 \subset \text{TC}^0 \subset \text{NC}^1 \subset \ldots \subset \text{CKT}$

2. If non-trivial algorithms exist for $C$-SAT, then
- improve the running time
- prove the difficulty of improvement
- additional properties: deterministic, counting, P-space,...
Why Algorithms for Circuit SAT?

1. Useful

- SAT can encode many combinatorial problems efficiently
- (sometimes) inspire algorithms for other problems
e.g. All-Pairs Shortest Paths (APSP) [Williams’14,...]
2. Connections to circuit lower bounds (This Talk)

- [black box] non-trivial $C$-SAT algorithm $\Rightarrow$ NEXP $\not\subseteq C$
  [Williams’10,11,...] (NEXP: nondeterministic exponential time)

- [white box] analysis of $C$-SAT algorithm
  $\Rightarrow$ average-case lower bounds
  [Santhanam’10, Seto-T’12, Chen-Kabanets-Kolokolova-Shaltiel-Zuckerman’14,...]
Boolean Circuit Complexity

studies the power of circuit classes defined using several parameters:

\#gates: \(O(n), O(n^2), \ldots, \text{poly}(n), \ldots, n^{\text{polylog} n}, \ldots, 2^{n^\varepsilon}, \ldots\)

Gate type: AND, OR, NOT, XOR, MOD, MAJORITY, THRESHOLD...

Structure:
• fanin: 2, 3, ..., const, ..., unbounded
• fanout: 1, 2, ..., const, ..., unbounded
• depth: 2, 3, ..., const, ..., log n, ..., polylog n, ...
• graph class: tree (=formula, fan-out=1), planar, ...
(k-)CNF ∈ AC⁰ ⊂ AC⁰[p] ⊂ ACC⁰ ⊆ TC⁰ ⊆ NC¹ ⊆ ... ⊆ CKT

- (k-)CNF: conjunction of disjunctions of (at most k) literals
- AC⁰: constant-depth, unbounded-fan-in, AND/OR/NOT
- AC⁰[p]: AC⁰ + mod p gates (p: prime power)
- ACC⁰: AC⁰ + mod m gates (m: integer ≥ 2)
- TC⁰: constant-depth, unbounded-fan-in, linear threshold (THR) gates: sgn(∑ₙᵢ₌₁ wᵢxᵢ − θ)
- NC¹: fan-in 2, fan-out 1, AND/OR/NOT(/XOR)
- CKT: fan-in 2, AND/OR/NOT(/XOR)
- C¹ ∘ C₂: composition of C₁ and C₂  e.g. CNF = AND ∘ OR

(Note: assume #gates = poly(n) unless otherwise specified)
Typical question:
Can a family of Boolean circuits \{C_n\} from a circuit class \(\mathcal{C}\) compute a problem \(X\)? (\(C_n\) is an \(n\)-variate circuit for each \(n\))

Important case (\(\text{NP} \subseteq \text{CKT}\)?):
- \(\mathcal{C}=\text{CKT}\): circuits with \(\text{poly}(n)\) number of gates
- \(X=\text{NP-complete problem, e.g., SAT, clique,}...\)

If the answer is no, then \(P \neq \text{NP}\) (by \(P \subsetneq \text{CKT}\))
(Approach to \(P\) vs. \(\text{NP}\) [Sipser’s Program])
\[ \ldots \subset AC^0[p] \subset ACC^0 \subset ACC^0 \circ \text{THR} \subset TC^0 \subset NC^1 \subset \ldots \subset CKT \]

Non-trivial $C$-SAT algorithm:

- $ACC^0 \circ \text{THR}$ with a super-poly \#gates [Williams’14]
- depth 2 $TC^0$ with a linear \#wires [Impagliazzo-Paturi-Schneider’13,...]

- $AC^0$: constant-depth, unbounded-fan-in, AND/OR/NOT
- $ACC^0$: $AC^0 + \text{mod } m$ gates ($m$: integer $\geq 2$)
- $TC^0$: constant-depth, unbounded-fan-in, linear threshold (THR) gates: $\text{sgn}(\sum_{i=1}^{n} w_i x_i - \theta)$
- $C_1 \circ C_2$: composition of $C_1 \circ C_2$
Non-trivial $C$-SAT algorithm:

- $\text{ACC}^0 \circ \text{THR}$ with a super-poly #gates [Williams’14]
- depth 2 $\text{TC}^0$ with a linear #wires [Impagliazzo-Paturi-Schneider’13,...]

New Results! (Note: #gates $\leq$ #wires)

(1) depth $d$ $\text{TC}^0$ with #wires $\ll n^{1+1/2^{O(d)}}$ [Chen-Santhanam-Srinivasan’16]
(2) depth 2 $\text{TC}^0$ with #gates $\ll n^{1.99}$ [Alman-Chan-Williams’16, T]
(3) depth 3 $\text{TC}^0$ with #gates $\ll n^{1.19}$, where the top gate is unweighted majority [ACW’16]
(4) Extensions of (2) to $\text{ACC}^0 \circ \text{THR} \circ \text{THR}$ and (3) to $\text{MAJ} \circ \text{AC}^0 \circ \text{THR} \circ \text{AC}^0 \circ \text{THR}$ with corresponding restrictions [ACW’16]
... ⊂ AC⁰[p] ⊂ ACC⁰ ⊆ ACC⁰∘THR ⊆ TC⁰ ⊆ NC¹ ⊆ ... ⊆ CKT

Lower Bounds for C:

- majority, mod q ∉ AC⁰[p] [Razborov’87, Smolensky’87]
- NEXP ∉ ACC⁰∘THR [Williams’14]
- parity ∉ depth d TC⁰ with #wires ≪ n^(1+1/3d)
  inner product ∉ depth d TC⁰ with #gates ≪ n
  [Impagliazzo-Paturi-Saks’93, Gröger-Turán’91]

- AC⁰[p]: AC⁰ + mod p gates (p: prime power)
- ACC⁰: AC⁰ + mod m gates (m: integer ≥ 2)
- TC⁰: constant-depth, unbounded-fan-in,
  linear threshold (THR) gates: sgn(Σᵢ₌₁ⁿ wᵢxᵢ − θ)
- C₁∘C₂: composition of C₁∘C₂
Lower Bounds for $C$:

- $\text{NEXP} \not\subseteq \text{ACC}^0 \circ \text{THR}$ [Williams’14]
- $\text{parity} \not\in \text{depth } d \text{ TC}^0$ with $\#\text{wires} \ll n^{1+1/3d}$
  - inner product $\not\in \text{depth } d \text{ TC}^0$ with $\#\text{gates} \ll n$
  [Impagliazzo-Paturi-Saks’93, Gröger-Turán’91]

New Results! [Kane-Williams’16]

1. $\exists L \in \text{P}$ s.t. $L \not\in \text{depth } 2 \text{ TC}^0$ with $\#\text{wires} \ll n^{2.49}$ or $\#\text{gates} \ll n^{1.49}$
2. $\exists L \in \text{P}$ s.t. $L \not\in \text{depth } 3 \text{ TC}^0$ with $\#\text{wires} \ll n^{2.49}$ or $\#\text{gates} \ll n^{1.49}$

where the top gate is unweighted majority
Lower Bounds for $\mathcal{C}$:

- $\text{NEXP} \not\subseteq \text{ACC}^0 \circ \text{THR}$ [Williams’14]
- $\text{parity} \not\in \text{depth } d \text{ TC}^0$ with $\#\text{wires} \ll n^{1+1/3d}$
  inner product $\not\in \text{depth } d \text{ TC}^0$ with $\#\text{gates} \ll n$
  [Impagliazzo-Paturi-Saks’93, Gröger-Turán’91]

New Results! (consequences of non-trivial SAT algorithms)

(1) $\exists L \in \text{E}^{\text{NP}}$ s.t. $L \not\in \text{depth 2 TC}^0$ with $\#\text{gates} \ll n^{1.99}$ [Alman-Chan-Williams’16, T]

cf. $\exists L \in \text{P}$, $\#\text{gates} \ll n^{1.49}$ [Kane-Williams’16]

(2) Extensions of (1) to $\text{ACC}^0 \circ \text{THR} \circ \text{THR}$ with corresponding restrictions [ACW’16]
Algorithm Design Paradigms

Paradigm 1: Restrict and Simplify (Divide and Conquer)

Step 1  Construct a partition (or covering) $D_1 \cup D_2 \cup \cdots = \{0,1\}^n$ based on the input $\mathcal{C}$

Step 2  For each $D_i$, simplify $\mathcal{C}|_{D_i}$

and check the satisfiability of it

(recursively or by another algorithm)

Typically $D_i$ is a subcube, affine subspace, solution space of polynomial equations etc.
Algorithm Design Paradigms

Paradigm 2: Polynomial Method (Low Rank Decomposition)

Step 1  Take a ``big OR'' of the input $C$:
$$C'(y) := \bigvee_{a \in \{0,1\}^{n'}} C(y, a)$$

Step 2  Represent $C'$ as $C'(y', y'') = h(\sum_{i=1}^{r} f_i(y') g_i(y''))$
for some $f_i, g_i: \{0,1\}^{(n-n')/2} \to R$ and $h: R \to \{0,1\}$

Step 3  Define $2^{(n-n')/2} \times r$ and $r \times 2^{(n-n')/2}$ matrices
$A_{y',i} := f_i(y'), B_{i,y''} := g_i(y'')$
Note that $(AB)_{y',y''} = \sum_{i=1}^{r} f_i(y') g_i(y'')$
i.e. the truth table of $C'$ is obtained via rectangular matrix multiplication (with $h$ applied later)
A Glance at Some Recent $C$-SAT algorithms
Some Recent $C$-SAT algorithms

Attempts to handle $TC^0$ (probably $C \not\subseteq ACC^0 \circ THR$)

- depth 2 $TC^0$ with a sub-quadratic #gates
- $AC^0$ with a limited #symmetric gates: $g(\sum_{i=1}^{n} x_i)$

Faster algorithms within $AC^0[p]$ ($C \subseteq ACC^0 \circ THR$)

- Systems of degree $k$ Polynomial Equations over $GF(q)$
- $AND \circ XOR \circ AND \circ XOR$
- $AC^0[p]$
Depth 2 $\text{TC}^0$ with a sub-quadratic \#gates
Motivation

Think about interesting $\mathcal{C} \subset \text{TC}^0 \setminus \text{ACC}^0 \circ \text{THR}$
i.e. $\mathcal{C} = \text{depth 2 TC}^0$ with #gates $= \omega(n)$
(Note: $\mathcal{C} \not\subseteq \text{ACC}^0 \circ \text{THR}$ is unknown)

Open question until 2015

$\exists L \in \text{E}^{\text{NP}}$ s.t. $L \not\in \text{depth-2 TC}^0$ with #gates $= O(n)$
(E$^{\text{NP}}$: languages computable by $2^{O(n)}$-time Turing machines with NP oracle)

Possible Attack

Non-trivial deterministic algorithms for $\mathcal{C} \Rightarrow \text{E}^{\text{NP}} \not\subset \text{C}$
[Ben-Sasson-Viola’14]
Recent $\mathcal{C}$-SAT algorithms (1)

**Theorem** [Alman-Chan-Williams’16, T]:

Let $\mathcal{C}$= depth 2 $\text{TC}^0$ with #gates = $m$

There is a deterministic algorithm for $\#\mathcal{C}$-SAT that runs in time $\text{poly}(n,m) \cdot 2^{n-\mu(n,m)}$,

where $\mu(n,m) = \Omega\left(\frac{n}{m^{1/2+o(1)}}\right)^c$, $\exists c > 0$

**Corollary:**

$\exists L \in E^{\text{NP}}$ s.t. $L \notin \mathcal{C}$ holds with $m = O(n^{1.99})$

Note: These results are incomparable with [Chen-Santhanam-Srinivasan’16,Kane-Williams’16]
Sketch of the Proof by [T]

based on the **Polynomial Method** in Circuit Complexity

- follow the **framework** for $\text{ACC}^0 \circ \text{THR}$ [Williams‘14]
- **probabilistic polynomial** for THR [Srinivasan‘13]
- **PRG** for space-bounded computation [Nisan‘92]
- some **transformation** techniques due to [Maciel-Therien‘98], [Beigel‘92]
- **fast evaluation algorithm** for $\text{SYM} \circ \text{SYM}$ [Williams‘14]

Some details later
$\text{AC}^0$ with a limited $\#$symmetric gates
Motivation

Think about some interesting $\mathcal{C} \subset TC^0 \setminus ACC^0 \circ THR$
i.e. $\mathcal{C} = \text{``AC}^0 \text{ with } t(n) \text{ weighted symmetric gates''}$
(Note: $\mathcal{C} \not\subseteq ACC^0 \circ THR$ is unknown)

Definition:

- $f: \{0,1\}^n \rightarrow \{0,1\}$ is symmetric (SYM)
  if $\exists g: \mathbb{Z} \rightarrow \{0,1\}, f = g(\sum_{i=1}^{n} x_i)$

- $f: \{0,1\}^n \rightarrow \{0,1\}$ is weighted symmetric
  if $\exists g: \mathbb{Z} \rightarrow \{0,1\}, \exists w_i \in \mathbb{Z}, f = g(\sum_{i=1}^{n} w_i x_i)$

AND, OR, parity, mod $m$, majority are symmetric
THR ($\text{sgn}(\sum_{i=1}^{n} w_i x_i - \theta)$) is weighted symmetric
Motivation

\[ \mathcal{C} = \text{``AC}^0 \text{ with } t(n) \text{ weighted symmetric gates''} \]

Interesting?

- contains Max SAT when depth 2, \( t(n) = 1 \) (THR\(^\circ\)OR)
  non-trivial algorithm for weighted Max 3-SAT is open (cf. \( 2^{0.791n} \) time algorithm for Max 2-SAT [Williams‘04])

- Lower bounds:
  generalized inner product (GIP) \( \not\in \text{AC}^0 \) with
  \[ \#\text{symmetric gates} = n^{1-o(1)} \text{ or } \#\text{THRs} = n^{1/2-o(1)} \]
  [...,Lovett-Srinivasan'11]
Recent $\mathcal{C}$-SAT algorithms (2)

Theorem [Sakai-Seto-T-Teruyama]:
Let $\mathcal{C}= \text{``AC}^0 \text{ with } t(n) \text{ weighted symmetric gates''}$
where $t(n) = n^{o(1)}$, $|w_i| = 2^{n^{o(1)}}$
There is a non-trivial deterministic algorithm for $\#\mathcal{C}$-SAT
(Note: we assume evaluation of symmetric gate is easy)

Special Case:
Max SAT can be solved in deterministic time $2^{n-n^{1/O(k)}}$
when $\#\text{clauses} = O(n^k)$
(Note: Max $k$-SAT $\Rightarrow$ $\#\text{clauses} = O(n^k)$)
By-products

Average-case lower bounds

\[ \exists L \in \mathbb{P} \text{ s.t. } \Pr[L_n(x) = C(x)] \leq \frac{1}{2} + \frac{1}{n^{\omega(1)}} \]

if \( C \in C = \text{``AC}^0 \text{ with } t(n) \text{ weighted symmetric gates''} \)

where \( t(n) = n^{o(1)} \), \(|w_i| = 2^{n^{o(1)}}\)

Worst-case lower bounds

\[ \exists L \in \mathbb{P} \text{ s.t. } L \notin \text{MAJ} \circ C, \text{ where } C \text{ is as above} \]

Remark (for specialists)

These seem to be the first lower bounds for such \( C \) bypassing communication complexity arguments
Lemma:
Let $\mathcal{C} = (\text{weighted SYM}) \circ \text{AND}$
where $\#\text{ANDs} = m$, $|w_i| \leq w$
There is a deterministic algorithm for $\#\mathcal{C}$-SAT
that runs in time $\text{poly}(n, m, \log w)2^{n - \mu(n, m, w)}$
where $\mu(n, m, w) = (n/\log(mw))^{\Omega(\log n/ \log m)}$

Based on ``Restrict and Simplify“ & DP

(Note: Theorem follows from Lemma and transformation $\text{AC}^0$ with symmetric gates $\Rightarrow$ $\text{SYM} \circ \text{AND}$ using
[Beigel-Reingold-Spielman'91,Beigel'92,Beame-Impagliazzo-Srinivasan'12])
Faster algorithms within $\text{ACC}^0$
(k-)CNF ⊂ AC₀ ⊂ AC₀[p] ⊂ ACC₀

$C$: $C$-SAT in time $T$, condition

1. $k$-CNF: $2^{n(1-\mu(k))}, \mu(k) = 1/O(k)$
   [Paturi-Pudlak-Zane’97,...]

2. CNF: $2^{n(1-\mu(c))}, \mu(c) = 1/O(\log c), \#\text{clauses} = cn$
   [Schuler’05,Calabro-Impagliazzo-Paturi’06,...]

3. AC₀: $2^{n(1-\mu(c,d))}, \mu(c, d) = 1/O(\log c + d \log d)^{d-1}$
   depth $d$, #gates = $cn$ [Impagliazzo-Matthews-Paturi’12]

4. ACC₀: $2^{n-\mu(n,d)}, \mu(n, d) = n^{1/2^{O(d)}}$
   depth $d$, #gates = $2^{n^{o(1)}}$ [Williams’11]
Recent $C$-SAT algorithms (3)

**Theorem** [Lokshtanov-Paturi-T-Williams-Yu]:

$C$: $C$-SAT in time $T$, condition

1. Systems of degree-$k$ Polynomial Equations over $GF(q)$
   
   ($=\text{AND} \circ \text{XOR} \circ \text{AND}_k \supset k\text{-CNF} = \text{AND} \circ \text{OR}_k$ if $q = 2$):
   
   $q^{n(1-\mu(k))}, \mu(k) = 1/O(k)$

2. $\text{AND} \circ \text{XOR} \circ \text{AND} \circ \text{XOR}$ ($\supset \text{CNF} = \text{AND} \circ \text{OR}$):
   
   $2^{n(1-\mu(c))}, \mu(c) = 1/O(\log c)$, $\#\text{ANDs} = cn$ at depth 3

3. $AC^0[p]$ ($\supset AC^0$):
   
   $2^{n(1-\mu(d,m))}, \mu(d, m) = 1/O(\log m)^{d-1}$

   depth $d$, $\#\text{gates} = m$
Proof Sketch

based on the **Polynomial Method** in Circuit Complexity

- probabilistic polynomial for AND/OR [Razborov’87, Smolensky’87]
  
- fast evaluation algorithm for polynomial

- degree reduction for AND\(\circ\)XOR\(\circ\)AND\(\circ\)XOR
  
  extending Schuler’s width reduction for CNF
  
  (``Restrict and Simplify``, where each partition is a solution space of low-degree polynomial equations)
Algorithms via Polynomial Method
Polynomial Method

Example [Razborov’87, Smolensky’87]:
1. $\text{AC}^0[p]$ can be well approximated by a low-degree $\text{GF}(p)$ polynomial
2. majority, mod $q$ cannot be well approximated by a low-degree $\text{GF}(p)$ polynomial

$1+2 \Rightarrow \text{majority, mod } q \notin \text{AC}^0[p]$

item 1 is useful in algorithm design
(``sparse“ suffices instead of ``low-degree“ in many cases)
Polynomial Method

Definition:
Let $f : \{0,1\}^n \rightarrow \{0,1\}$

A distribution $P$ over polynomials is an $\epsilon$-error probabilistic polynomial for $f$ if $\forall x, \Pr_{p \sim P}[p(x) \neq f(x)] \leq \epsilon$

$\deg(P) \leq d$ if $\Pr_{p \sim P}[\deg(p) \leq d] = 1$

$\epsilon$-error probabilistic $C$-circuit is defined analogously
Algorithm for $\mathcal{C}$-SAT

Input: $C: \{0,1\}^n \rightarrow \{0,1\}, C \in \mathcal{C}$

Step 1. Define $C': \{0,1\}^{n-n'} \rightarrow \{0,1\}, C' \in \text{OR} \circ \mathcal{C}$ as

$$C'(y) \coloneqq \bigvee_{a \in \{0,1\}^{n'}} C(y, a) \quad \text{(Note: } |C'| \approx 2^{n'}|C|\text{)}$$

Step 2. Construct $(1/3)$-error probabilistic polynomial $p$ for $C'$ in time $T(n, n', |C|)$

Step 3. Evaluate $p(y)$ for all $y \in \{0,1\}^{n-n'}$ in time $T'(n, n', |C|)$

Step 4. repeat 2-3 $O(n)$ times to reduce error probability

Output: truth table $V$ of $C'$ such that

$$\forall y, \text{Pr}[V(y) \neq C'(y)] \leq 2^{-2n}$$

Running Time: $O(n(T(n, n', |C|) + T'(n, n', |C|)))$
Ingredients for THR$\circ$THR (1/3)

Lemma [Razborov’87, Smolensky’87]:
There exists $\epsilon$-error probabilistic polynomial for AND/OR of degree \( \log(1/\epsilon) \) and it is efficiently samplable

Lemma [Srinivasan’13]:
There exists $\epsilon$-error probabilistic polynomial for THR of degree \( \sqrt{n\ \text{polylog}(n/\epsilon)} \) and it is efficiently samplable
Ingredients for THR$\circ$THR (2/3)

Lemma:
$$\text{XOR} \circ \text{AND} \circ \text{XOR} \circ \text{AND} \subseteq \text{XOR} \circ \text{AND}$$

Lemma [Maciel-Therien’98]:
$$\text{THR} \subseteq \text{AC}^0[2] \circ \text{SYM}$$

Lemma [Beigel’92]:
$$\text{AND} \circ \text{SYM} \subseteq \text{weighted SYM}$$
Lemma [Williams’14]:

Let $C: \{0,1\}^n \rightarrow \{0,1\}$, $C \in \text{SYM} \circ (\text{weighted SYM})$ where

- top gate has fan-in $u$
- bottom gates have maximum weight $w$

such that $uw \leq 2^{0.1n}$

Then, truth table of $C$ can be generated in time $\text{poly}(n)2^n$
Algorithm for \( \text{THR} \circ \text{THR-SAT} \)

**Input:** \( C: \{0,1\}^n \to \{0,1\}, C \in \text{THR} \circ \text{THR} \)

**Step 1.** Define \( C': \{0,1\}^{n'} \to \{0,1\}, C \in \text{OR} \circ C \) as

\[
C'(y) := \bigvee_{a \in \{0,1\}^{n'}} C(y, a) \quad (\text{Note: } |C'| \approx 2^{n'}|C|)
\]

**Step 2.** Construct \((1/3)\)-error probabilistic circuit \( C'' \) for \( C' \)

\[
C'' \in (\text{XOR} \circ \text{AND}) \circ (\text{XOR} \circ \text{AND}) \circ \cdots \circ (\text{XOR} \circ \text{AND}) \circ \text{SYM}
\]

**Step 3.** Transform \( C'' \) into \( C''' \in \text{XOR} \circ (\text{weighted SYM}) \)

**Step 4.** Evaluate \( C'''(y) \) for all \( y \in \{0,1\}^{n-n'} \)

**Step 5.** Repeat 2-4 \( O(n) \) times to reduce error probability

**Output:** truth table of \( C' \)

(Note: need more work for deterministic algorithms)
Recent $\mathcal{C}$-SAT algorithms (1)

Theorem [Alman-Chan-Williams’16,T]:
Let $\mathcal{C} = \text{depth 2 } \text{TC}^0$ with \#gates $= m$
There is a deterministic algorithm for \#$\mathcal{C}$-SAT
that runs in time $\text{poly}(n,m) \cdot 2^{n - \mu(n,m)}$,
where $\mu(n,m) = \Omega\left(n/m^{1/2+o(1)}\right)^c$, $\exists c > 0$

Corollary:
$\mathsf{E}^{\mathsf{NP}} \not\subseteq \mathcal{C}$ holds with $m = O(n^{1.99})$
Let $f : \{0,1\}^n \rightarrow \{0,1\}$

A distribution $P$ over polynomials is an $\epsilon$-error probabilistic polynomial for $f$
if $\forall x$, $\Pr_{p \sim P}[p(x) \neq f(x)] \leq \epsilon$

Lemma [Srinivasan’13]:
There exists $\epsilon$-error probabilistic polynomial for THR
of degree $\sqrt{n} \text{polylog}(n/\epsilon)$ and it is efficiently samplable

We cannot improve the lemma under the above definition
of $\epsilon$-error probabilistic polynomial
Further Improvement?

All we want to do in the polynomial method is:

**Step 1** Take a "big OR" of the input $C$:

$$C'(y) := \bigvee_{a \in \{0,1\}^{n'}} C(y, a)$$

**Step 2** Represent $C'$ as

$$C'(y', y'') := h(\sum_{i=1}^{r} f_i(y') g_i(y''))$$

for some $f_i, g_i: \{0,1\}^{(n-n')/2} \rightarrow R$ and $h: R \rightarrow \{0,1\}$

Typically, $\sum_{i=1}^{r} f_i(y') g_i(y'')$ is a sum of monomials and $h$ is a threshold or modulo function

Try different representations, e.g. $\{1,2\}$ instead of $\{0,1\}$, $R =$ complex numbers, $\sum_{i=1}^{r} f_i(y') g_i(y'')$ as a trigonometric polynomial,... etc.
Conclusion
Some Recent $\mathcal{C}$-SAT algorithms

Attempts to handle $\text{TC}^0$ (probably $\mathcal{C} \not\subseteq \text{ACC}^0 \circ \text{THR}$)

- depth 2 $\text{TC}^0$ with a sub-quadratic \#gates
- $\text{AC}^0$ with a limited \#symmetric gates: $g(\sum_{i=1}^{n} x_i)$

Faster algorithms within $\text{AC}^0[p]$ ($\mathcal{C} \subseteq \text{ACC}^0 \circ \text{THR}$)

- Systems of degree $k$ Polynomial Equations over $\text{GF}(q)$
- $\text{AND} \circ \text{XOR} \circ \text{AND} \circ \text{XOR}$
- $\text{AC}^0[p]$
Open Questions

- without polynomial method? (in polynomial space?)
- other algebraic representations useful?
  - e.g. Barrington’s theorem, randomized encoding, span program,…
- other lower bound techniques useful?
  - e.g. communication complexity, proof complexity, mathematical programming,…
- lower bound for \( C \leftrightarrow \) non-trivial \( C \)-SAT algorithm?
  (cf. [Carmosino-Impagliazzo-Kabanets-Kolokolova,CCC'16])

Thank you!