（量子）回路計算量の下界証明

河内 亮周
Akinori KAWACHI
三重大学

京都大学基礎物理学研究所 量子情報ユニット
第３回量子情報スクール
2020年6月30日(火)

Overview

1. Circuit lower bounds in high complexity classes
2. Circuit lower bounds in low complexity classes
3. Quantum circuit lower bounds
4. Proof techniques for circuit lower bounds

Circuit Model (bounded fan-in)

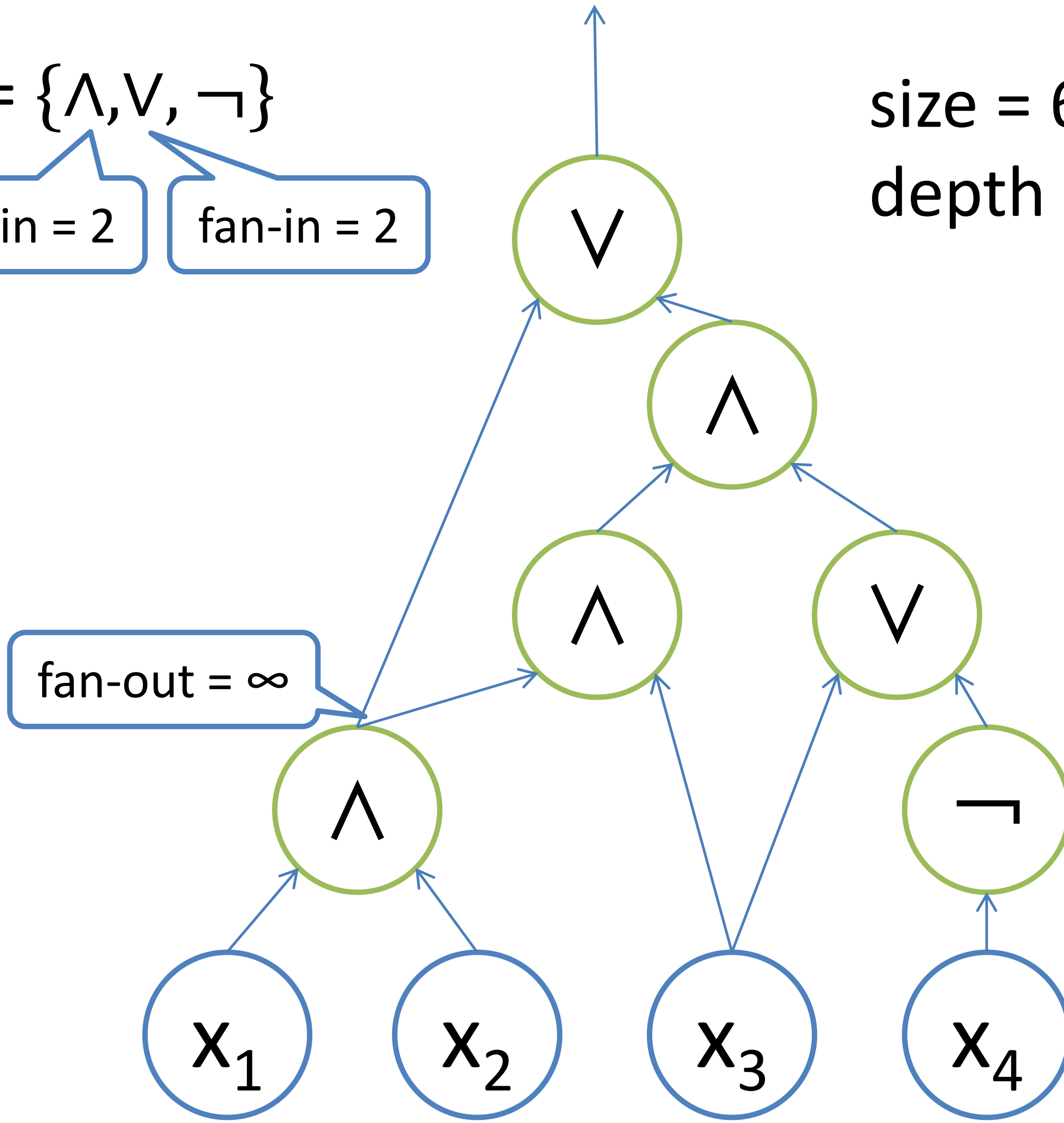
Gate set = $\{\wedge, \vee, \neg\}$

fan-in = 2

fan-in = 2

size = 6

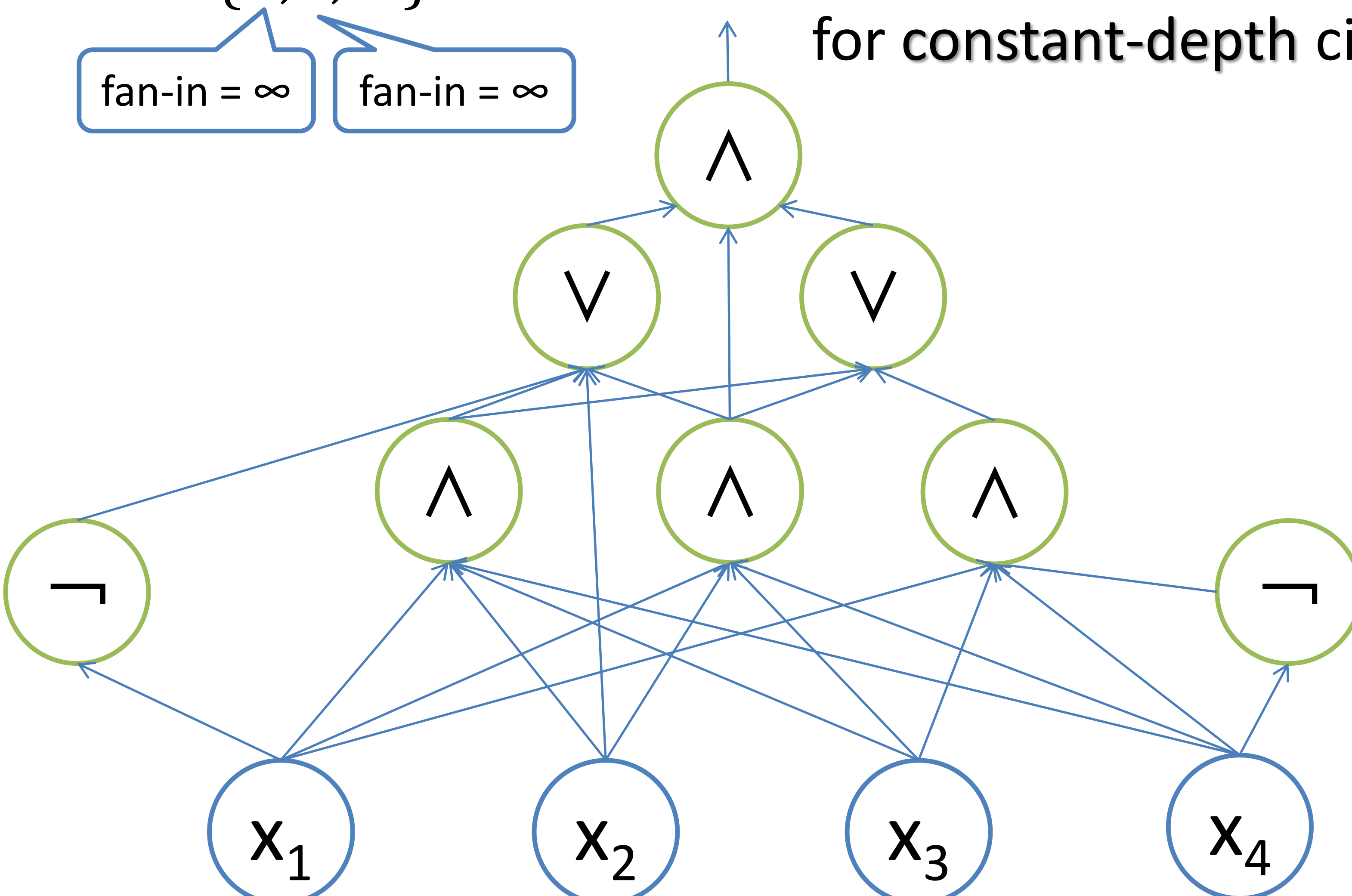
depth = 4



Circuit Model (**un**bounded fan-in)

Gate set = {∧,∨, ¬}

circuit model
for **constant-depth** circuits



Circuit Complexity

Circuit Complexity

A problem L has circuit complexity $s(n)$

= necessary and sufficient size of circuits that computes L on every input length n

Constructing circuits of size $s(n)$ for L ➔ circuit upper bounds $s(n)$

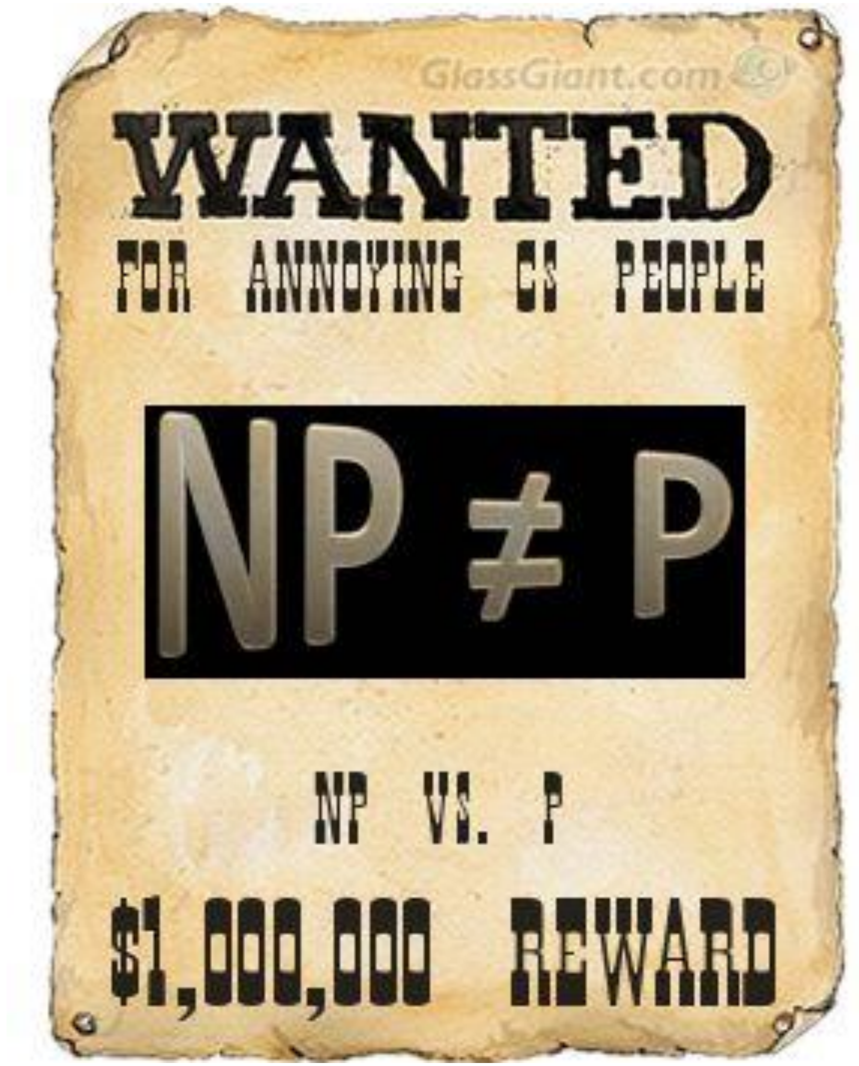
Proving no circuit of size $s(n)$ for L ➔ circuit lower bounds $s(n)$

This talk

Overview

1. Circuit lower bounds in high complexity classes
2. Circuit lower bounds in low complexity classes
3. Quantum circuit lower bounds
4. Proof techniques for circuit lower bounds

Why Circuit Lower Bounds in High Complexity Classes?



Implications of Circuit Lower Bounds

Major Strategy towards NP vs. P

Proving **circuit lower bounds** for class NP:

No poly-size circuit can compute some NP problem

⇓

solved by
poly-size circuits
≈ class P

$\mathrm{NP} \neq \mathrm{P}$

($\mathrm{NP} \not\subset \mathrm{P/poly} \rightarrow \mathrm{NP} \neq \mathrm{P}$)

Implications of Circuit Lower Bounds

Universal derandomization of randomized algorithms

Stay tuned for the next session of Shuichi's talk!

Complexity Classes

- Focus on "decision prob[lems" in this talk]
  - Answer = Yes or No
- **P** = problems which can be solved efficiently by deterministic classical algorithms

  (formally, Turing machines).
- **NP** = problems whose "witnesses" can be verified efficiently by deterministic classical algorithms.

polynomial-time (e.g. n^2-time) in input length n

algorithm = deterministic classical algorithm (unless specified otherwise)

Complexity Classes

- **P/poly** = problems solved efficiently by classical circuits.
  - $\mathrm{P} \subsetneq \mathrm{P/poly}$
- **SIZE$[s(n)]$** = problems solved by $s(n)$-size classical circuits.
  - P/poly = SIZE[poly(n)]

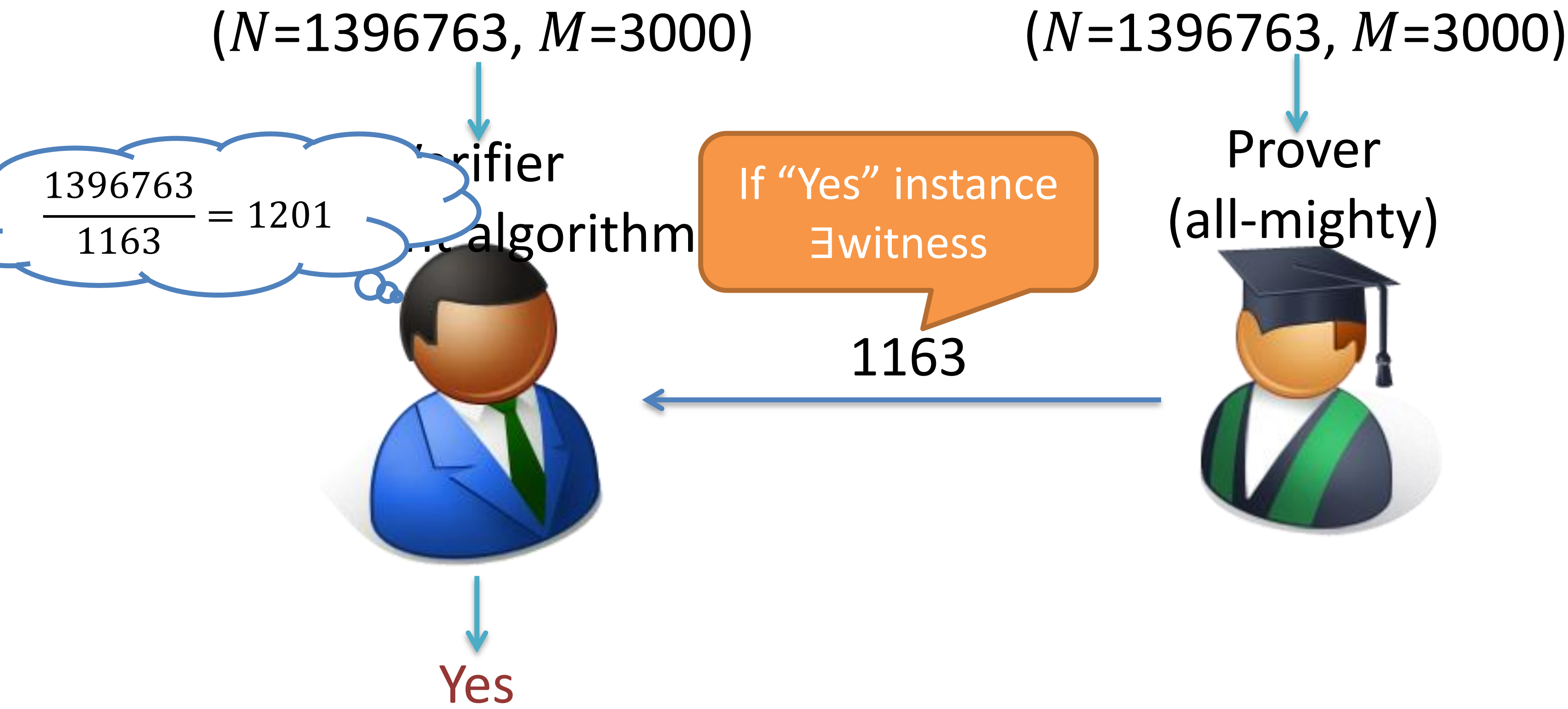
polynomial-size
in input length n

circuit = deterministic classical circuit
(unless specified otherwise)

Recap: class NP

- **NP** = problems whose "witnesses" can be verified by efficient algorithms.

Problem: N is divided by $< M$?



Recap: class NP

- **NP** = problems whose "witnesses" can be verified by efficient algorithms.

If "No" instance
no witness
$1396763 = 1163 \times 1201$

m: N is divided by $< M$?

(N=1396763, M=1000)

(N=1396763, M=1000)

erifier
nt algorithm)

$\frac{1396763}{967}$ is not int.

Prover
(all-mighty)

967

whatever Prover sends,
Verifier isn't cheated.

No

Recap: class NP

Class NP

$L \in \mathrm{NP}$

Def $\Leftrightarrow$

$x \in L \Rightarrow \exists w \colon V(x, w) = 1$

$x \notin L \Rightarrow \forall w \colon V(x, w) = 0$

$|w| = \mathrm{poly}(|x|)$

V: poly-time algorithm

Recap: NP-complete problem

Problem: SAT

Given: Boolean formula $\phi(x_1, \ldots, x_n)$

Decide: ϕ is satisfiable?

$\exists (a_1 \cdots a_n) \in \{0,1\}^n : \phi(a_1, \ldots, a_n) = 1?$

$x_1 \wedge x_2 \in \mathrm{SAT}\ (x_1 = 1, x_2 = 1)$

$x_1 \wedge \neg x_1 \notin \mathrm{SAT}$

- SAT is NP-complete problem
  - $\mathrm{SAT} \in \mathrm{P} \rightarrow \mathrm{NP} = \mathrm{P}$
  - SAT is the "hardest" in NP.

Circuit Lower Bounds for NP

The best circuit lower bound is:

Theorem [Iwama, Lachish, Morizumi & Raz (2005)]

$$\mathrm{NP} \not\subset \mathrm{SIZE}[5n]$$

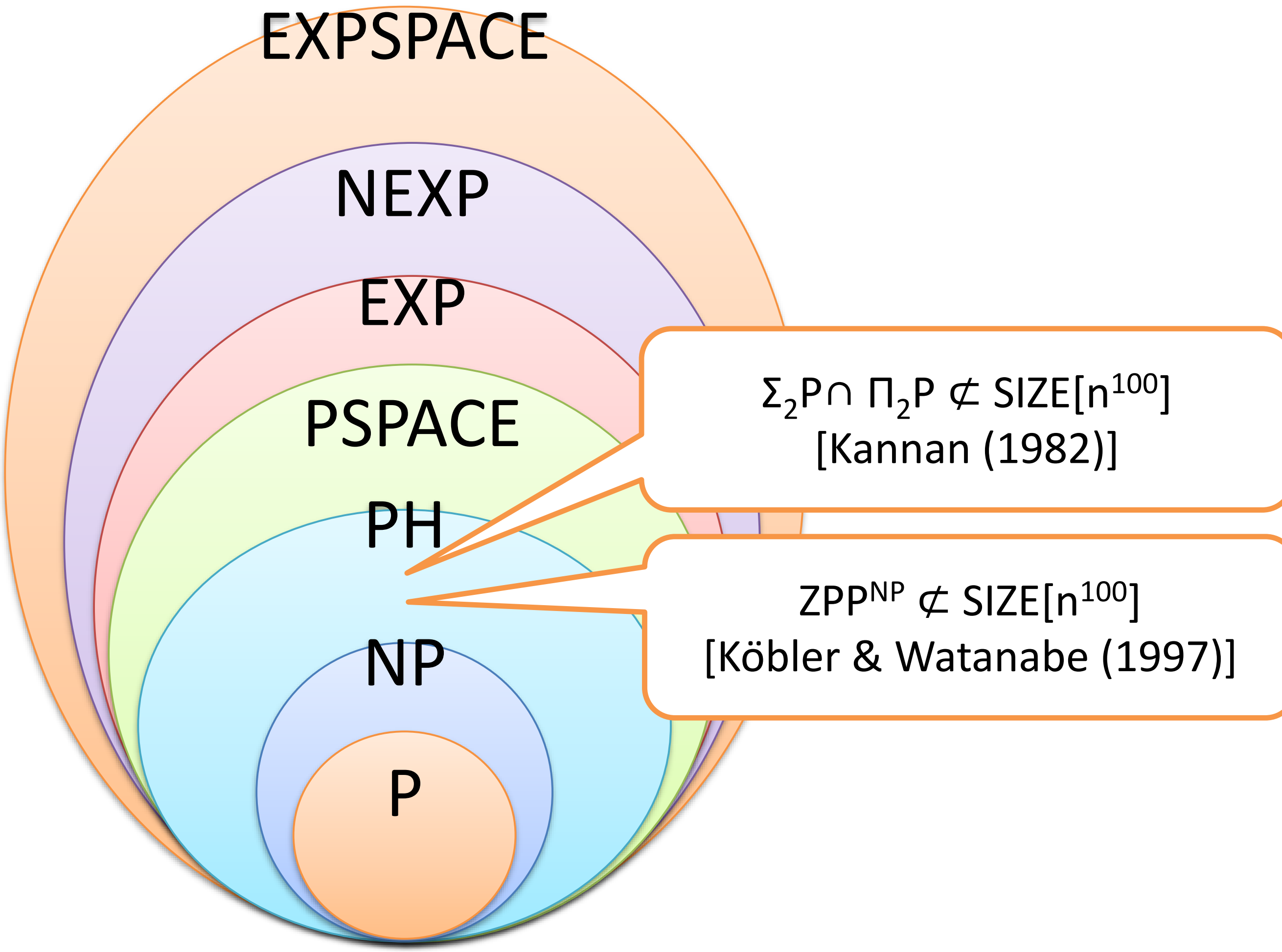
Only **linear** lower bounds!

We can't yet exclude the possibility
NP-complete problems could be solved by **6n-size** circuit!

Relaxation:

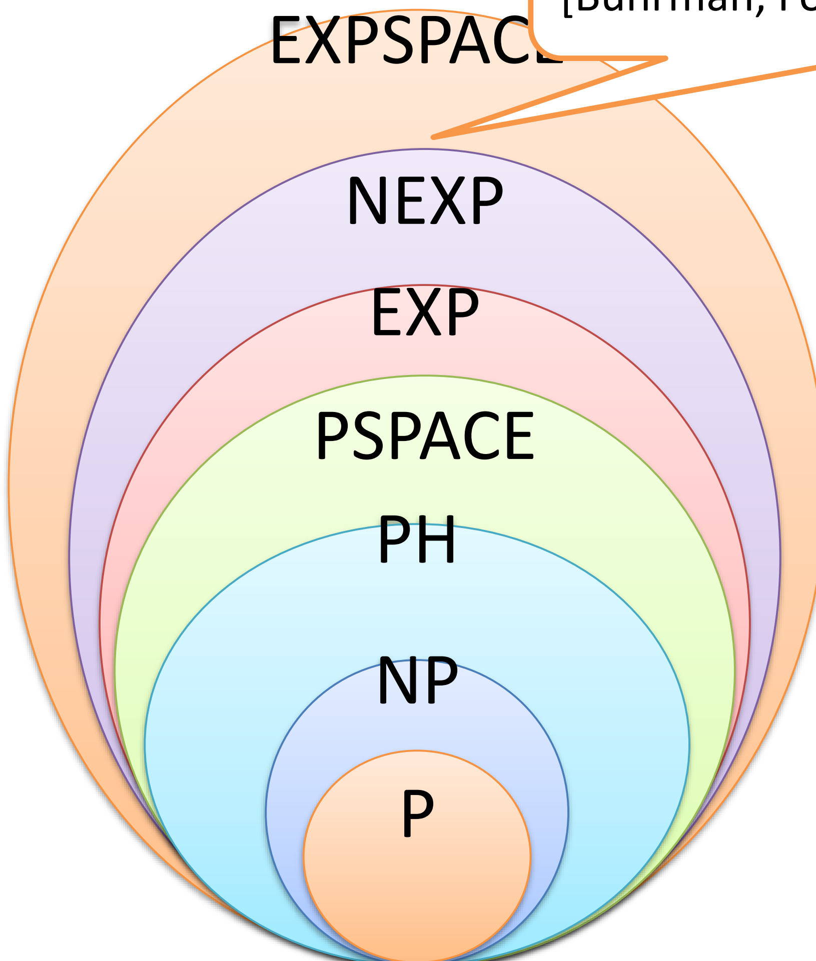
- superlinear circuit lower bounds
- circuit lower bounds in higher classes than NP

Superlinear Circuit Lower Bounds in High Complexity Classes



Superpolynomial Lower Bounds in High Complexity Classes

$MA_{EXP} \not\subset P/poly$
[Buhrman, Fortnow & Thierauf (1998)]



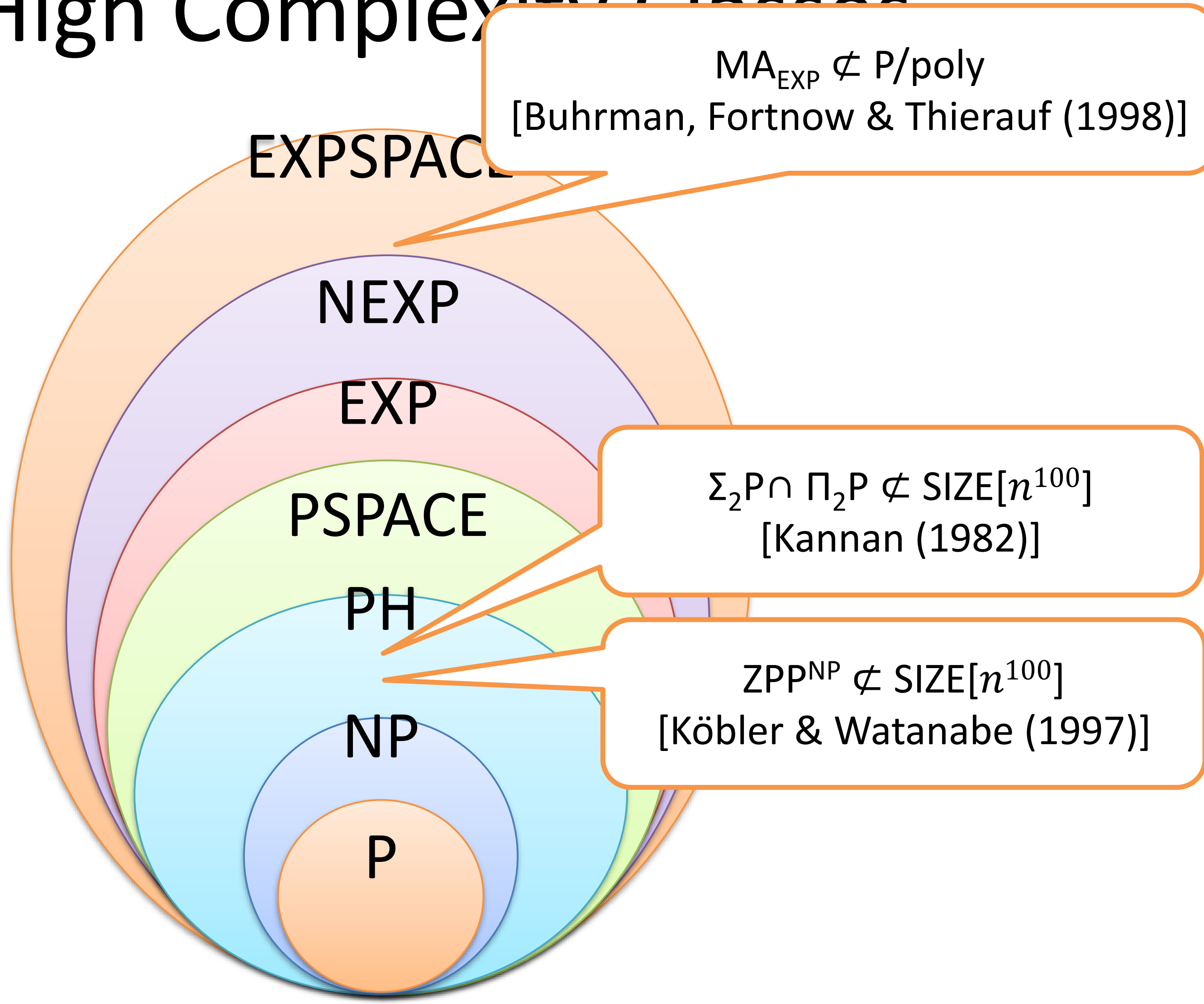
Complexity Classes

- **PH** (Polynomial-time Hierarchy) = $\mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^{\cdots}}}$
  - Generalization of class NP.
  - c.f. $\boldsymbol{\Sigma_2}\mathbf{P} = \mathrm{NP}^{\mathrm{NP}}$ = problems verified by polynomial-time algorithms with NP oracle
    - NP oracle = black box solving any NP problem in 1 step.
- **PSPACE** = problems solved by polynomial-space (poly(n)-space) algorithms.
  - No time bounds.

Complexity Classes

- **EXP** = problems solved by exponential-time ($2^{\text{poly}(n)}$-time) algorithms.
  - Exponential-time analogue of class P
- **NEXP** = problems verified by exponential-time algorithms.
  - Exponential-time analogue of class NP
- **EXPSPACE** = problems solved by exponential-space ($2^{\text{poly}(n)}$-space) algorithms.
  - Exponential-space analogue of class PSPACE

Circuit Lower Bounds in High Complexity Classes



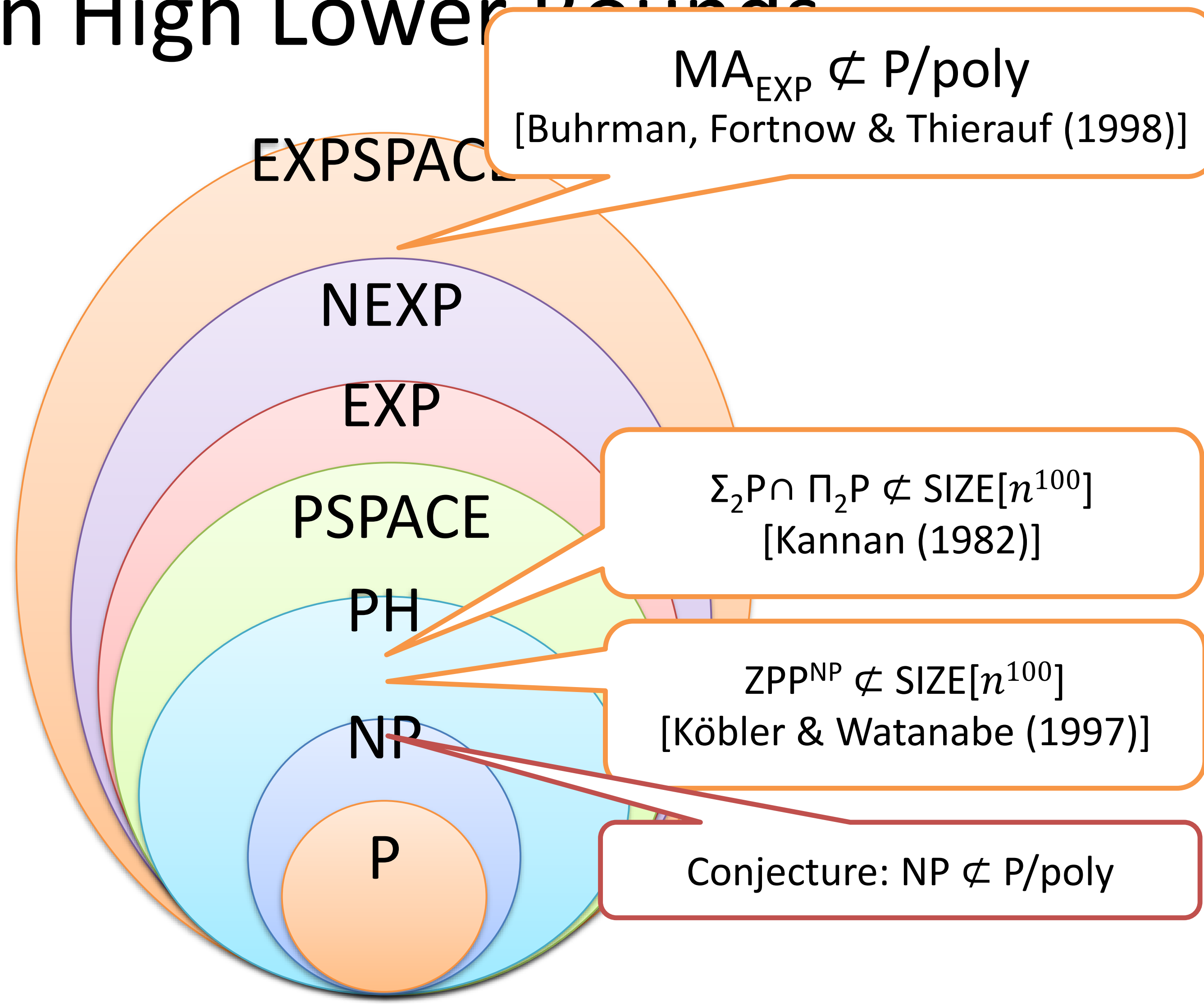
Complexity Classes

- $\mathbf{\Sigma_2 P}$ = $\mathrm{NP}^{\mathrm{NP}}$, $\mathbf{\Pi_2 P}$ = complement class of $\mathbf{\Sigma_2 P}$
- **ZPP** (Zero-error Probabilistic Polynomial-time) = problems solved by expected polynomial-time randomized algorithm with zero error
- **$\mathbf{ZPP}^{\mathbf{NP}}$** = problems solved by expected polynomial-time randomized algorithm with zero error with NP oracle

Complexity Classes

- **MA** (Merlin-Arthur) = problems which can be verified by polynomial-time randomized algorithms with high probability.
  - Randomized analogue of class NP
- **MA_{EXP}** = problems which can be verified by exponential-time randomized algorithms with high probability.
  - Exponential-time analogue of class MA

Circuit Lower Bounds in High Lower Bounds



Breakthrough from Algorithm Design

poly-size
constant-depth circuits
with modulo gates

Theorem [Williams (2014)]

NEXP $\not\subset$ ACC^0

Pro

Given a circuit C of class $\mathbb{C}$ (e.g., P/poly, ACC^0), decide whether C is satisfiable.

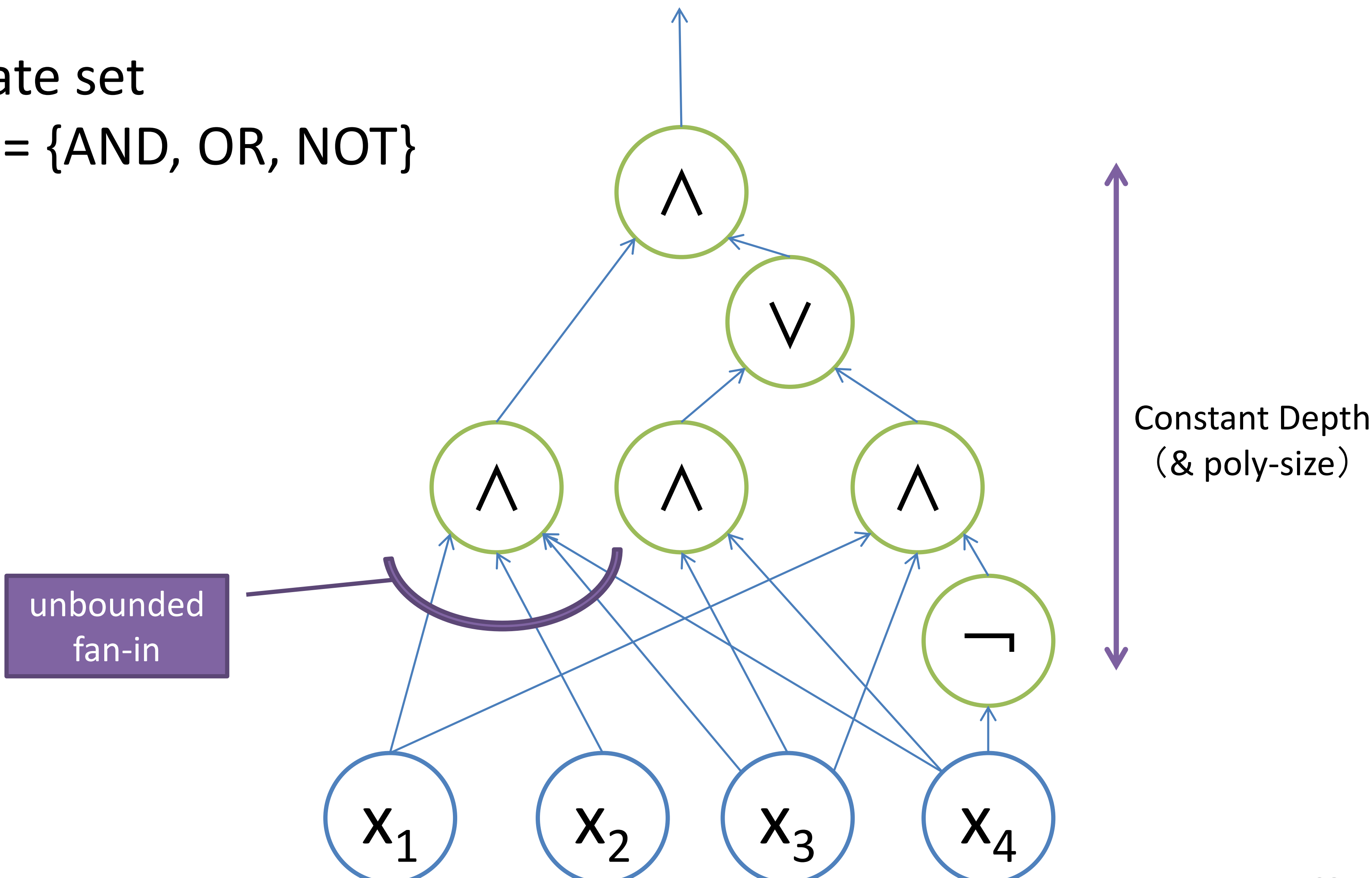
1st step: $\exists(2^n/\mathrm{superpoly}(n))$-time algorithm for $\mathbb{C}$-CKT-SAT

➔ NEXP $\not\subset \mathbb{C}$

2nd step: $(2^n/\mathrm{superpoly}(n))$-time algorithm for ACC^0-CKT-SAT

AC^0

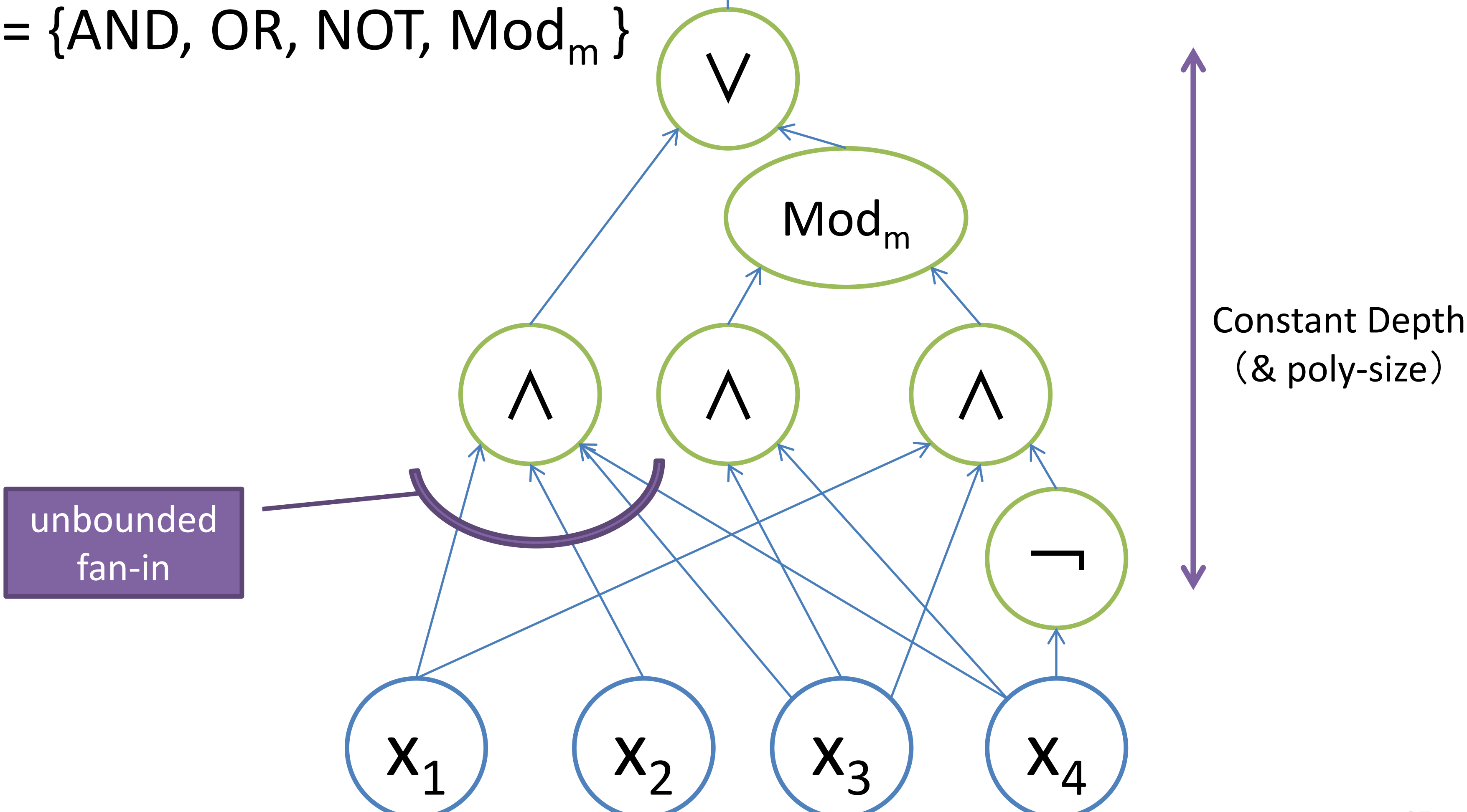
Gate set
= {AND, OR, NOT}



ACC^0 (AC^0 with counter)

$\mathrm{Mod}_m(x) = 1$
iff $m \mid wt(x)$

Gate set
= {AND, OR, NOT, Mod_m }



Breakthrough from Algorithm Design

Theorem [Williams (2014)]

NEXP $\not\subset$ ACC^0

Pro

Given a circuit C of class $\mathbb{C}$ (e.g., P/poly, ACC^0), decide whether C is satisfiable.

1st step: $\exists$(2^n/superpoly(n))-time algorithm for $\mathbb{C}$-CKT-SAT

➔ NEXP $\not\subset$ $\mathbb{C}$

2nd step: (2^n/superpoly(n))-time algorithm for ACC^0-CKT-SAT

Breakthrough from Algorithm Design

ACC^0 circuit +
linear threshold gates
at bottom layer

Theorem [Williams (2018)]

$$\mathrm{NEXP} \not\subset \mathrm{ACC}^0 \circ \mathrm{THR}$$

Improvement

Non-trivially faster algorithm for ACC^0∘THR-CKT-SAT (2nd step)

Breakthrough from Algorithm Design

Theorem [Murray & Williams (2018)]

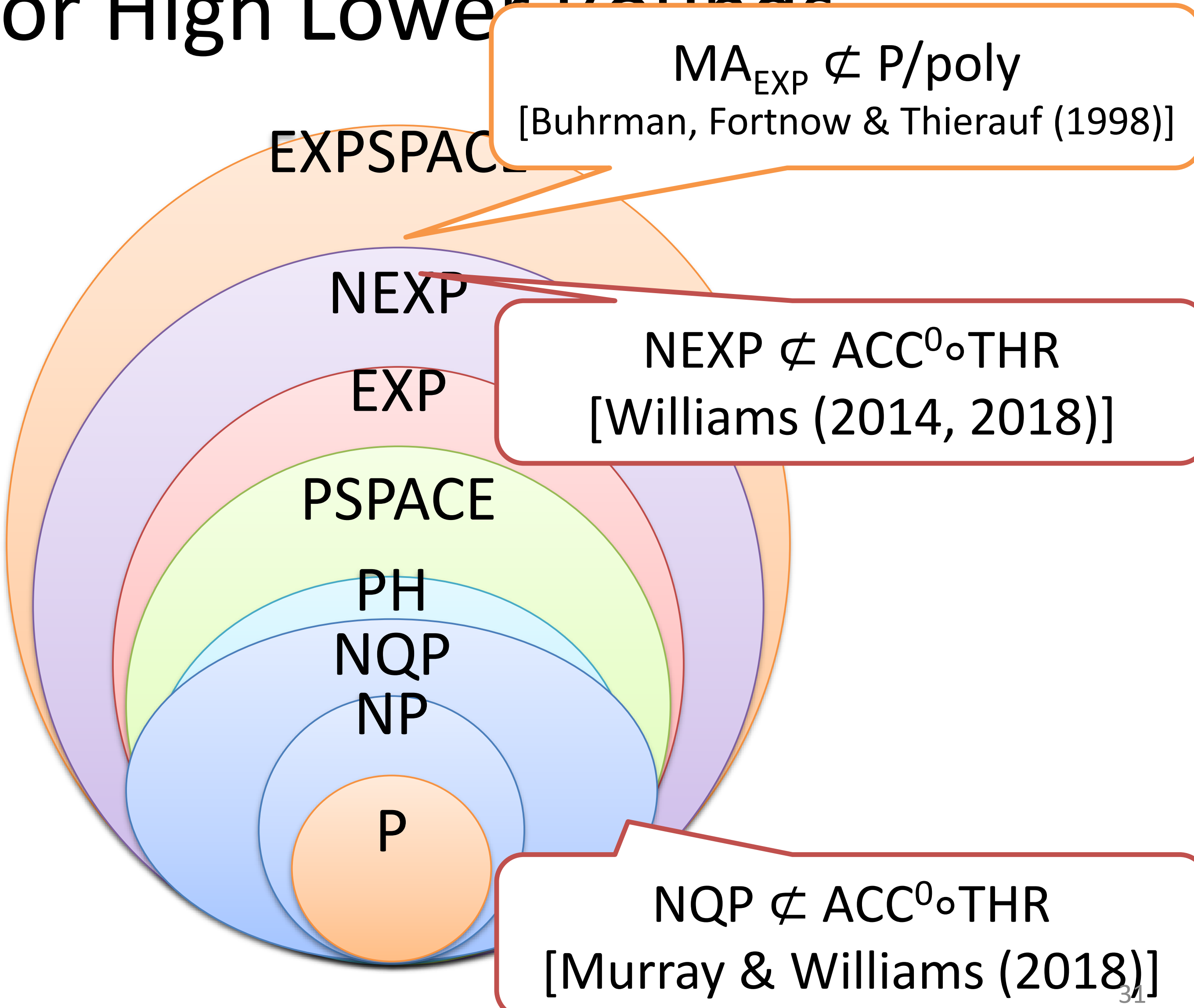
$$\mathrm{NQP} \not\subset \mathrm{ACC}^0 \circ \mathrm{THR}$$

$n^{\mathrm{polylog}\, n}$-time version of NP

Improvement

NEXP can be replaced with NQP (1st step)

Circuit Lower Bounds for High Lower Bounds



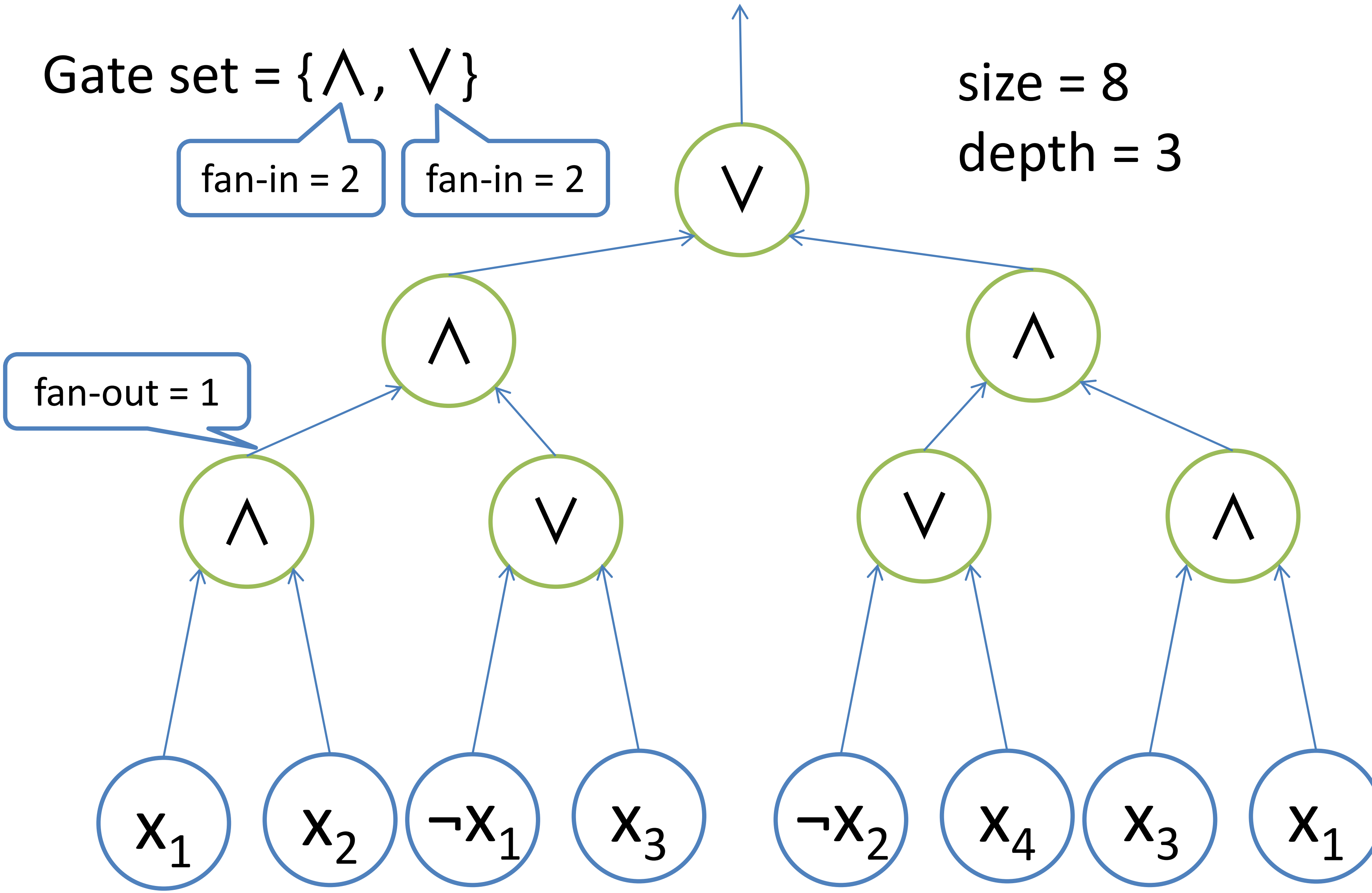
Overview

1. Circuit lower bounds in high complexity classes
2. Circuit lower bounds in low complexity classes
3. Quantum circuit lower bounds
4. Proof techniques for circuit lower bounds

Circuit Lower Bounds for Low Complexity Classes

- Computational power of restricted circuits?
  - Boolean formulas
    - de Morgan formulas
    - Formulas over full binary basis
  - Low-depth (shallow) circuits
    - constant-depth circuits
    - $O(\log(n))$-depth circuits

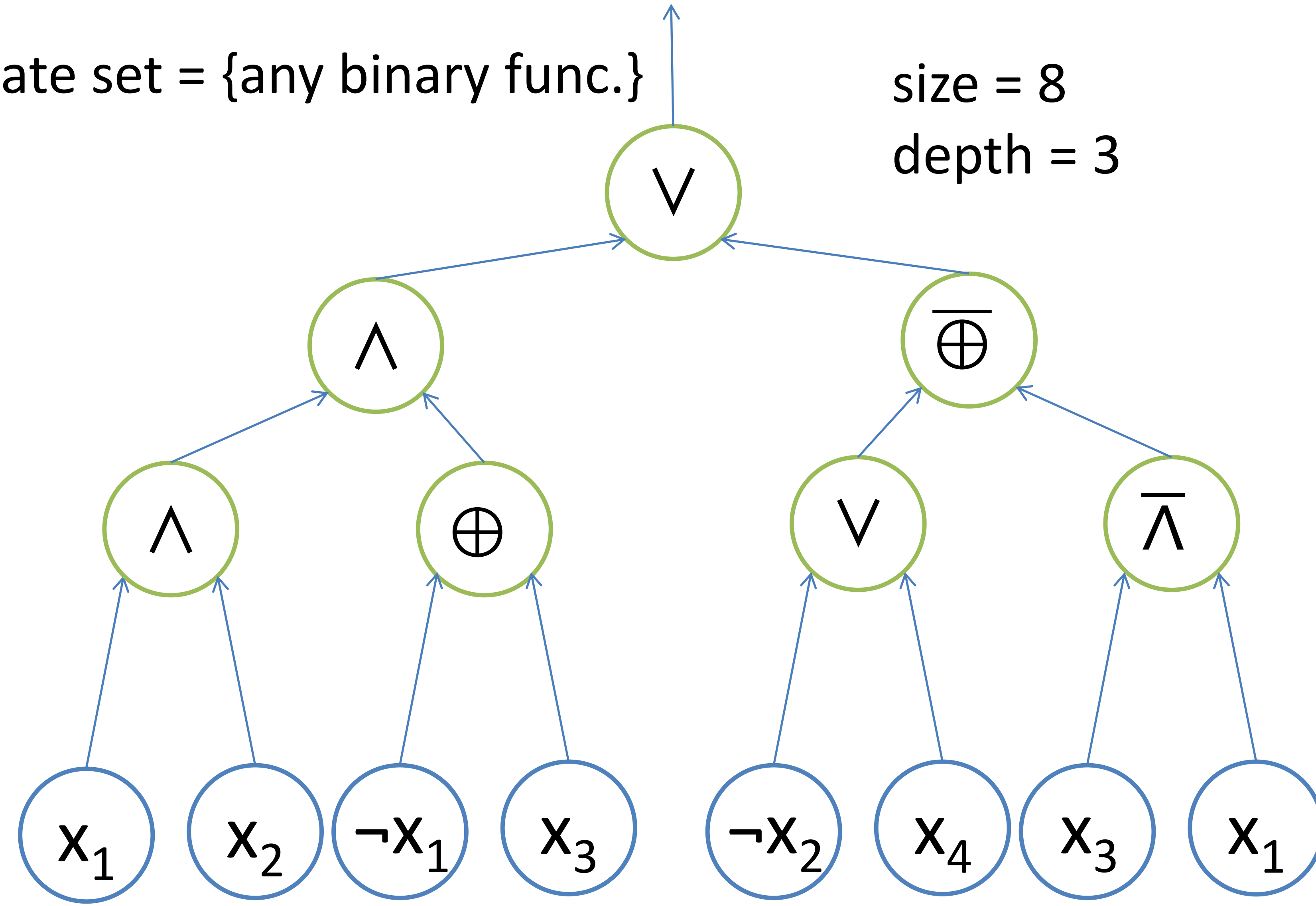
Boolean Formula (de Morgan)



Boolean Formula (Full Binary Basis)

Gate set = {any binary func.}

size = 8
depth = 3

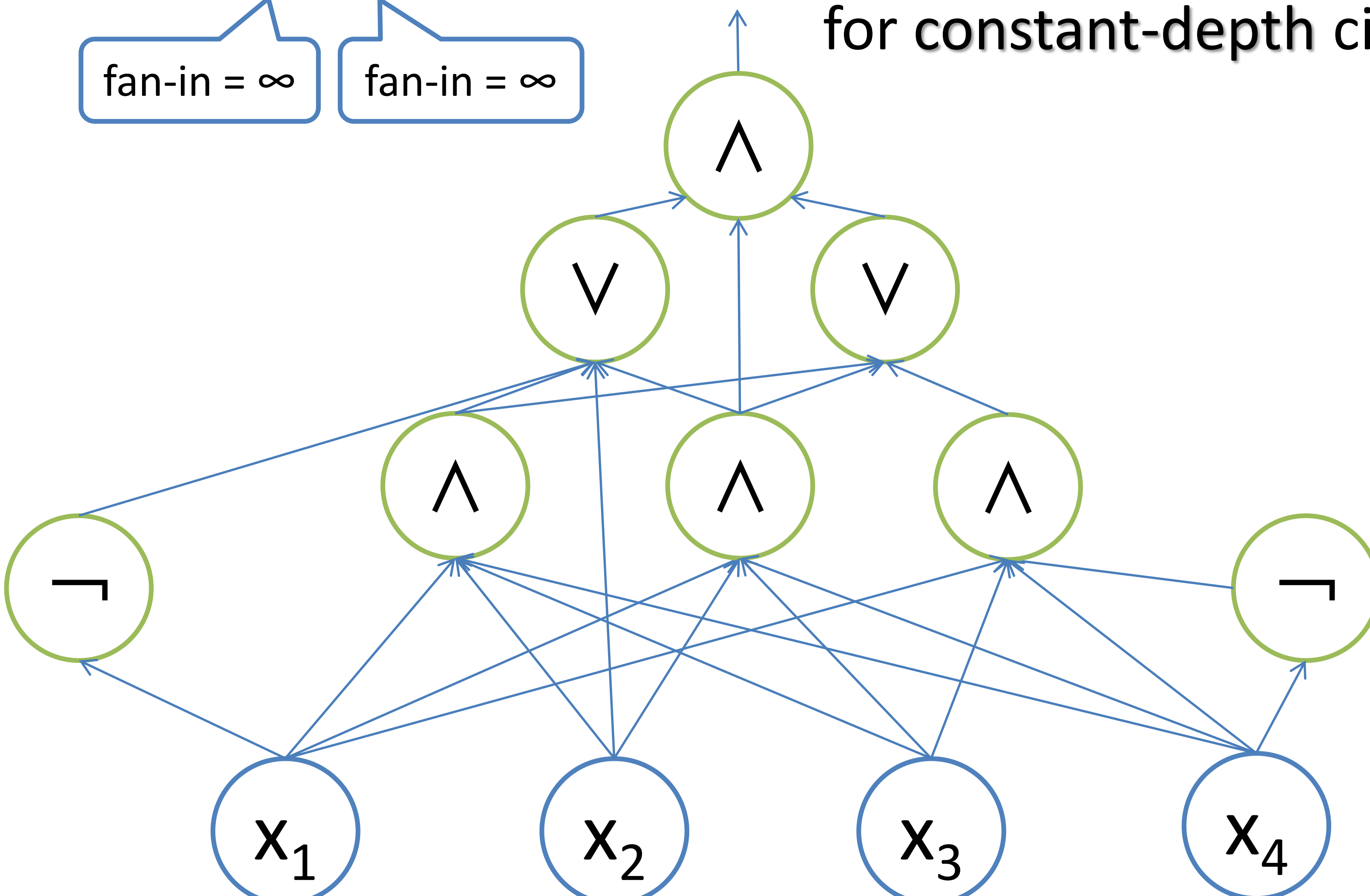


Circuit Model (**un**bounded fan-in)

Gate set = {$\wedge$, $\vee$, $\neg$}

fan-in = ∞ fan-in = ∞

circuit model
for **constant-depth** circuits



Low-Depth Circuit Classes

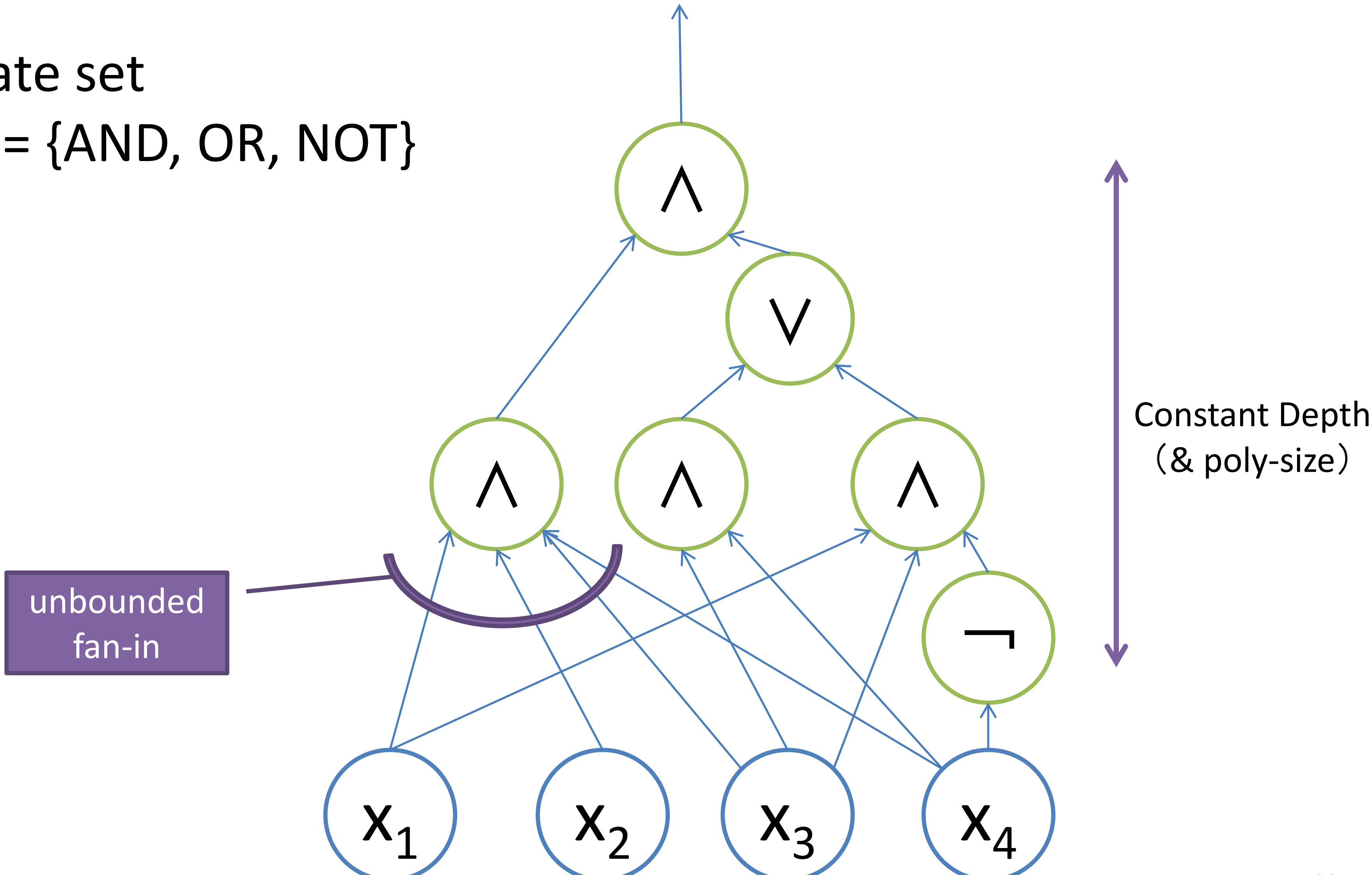
- AC^i = problems solved by $O(\log^i n)$-depth poly-size circuit of unbounded fan-in
- NC^i (Nick's Class) = problems solved by $O(\log^i n)$-depth poly-size circuit of bounded fan-in

Nicholas Pippenger

出典: https://www.hmc.edu/mathematics/people/faculty/nicholas-pippenger/

AC^0

Gate set
= {AND, OR, NOT}



Why Circuit Lower Bounds for Low Complexity Classes?

- Relaxation for circuit lower bounds
  - Too difficult to prove lower bounds in general circuit models!
  - Towards understanding of proof techniques in successful cases for weaker circuit models.
- P vs. NC^1 conjecture
  - Is every P problem parallelizable?
    - NC^1 problem is $O(\log(n))$-time solvable by parallel computation.
    - poly-size Boolean formulas $\equiv$ NC^1 circuits

Parity

Problem: Parity

Given: n-bit string $x \in \{0,1\}^n$

Decide: #1 of x is odd or not.

i.e., $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$?

Remark: Parity $\in \mathrm{NC}^1$

Some restricted circuits cannot compute Parity!

Formula Lower Bounds

The lower bound of Parity for de Morgan formulas:

Theorem [Khrapchenko (1971)]

$$\mathrm{L}_{\mathrm{dM}}(\mathrm{Parity}) \geq n^2$$

$\mathrm{L}_{\mathrm{dM}}(f)$ = size of minimum de Molgan formula computing f

It is known $\mathrm{L}_{\mathrm{dM}}(\mathrm{Parity}) \leq n^2$ [Tarui (2010)], i.e., the bound is tight.

Formula Lower Bounds

The best known lower bound for de Morgan formulas:

Theorem [Tal (2017)]

$$\mathrm{L}_{\mathrm{dM}}(\mathrm{KR}) = \Omega\left(\frac{n^3}{\log n \cdot (\log\log n)^2}\right)$$

$\mathrm{L}_{\mathrm{dM}}(\mathrm{f})$ = size of minimum de Molgan formula computing f

KR:$\{0,1\}^n \to \{0,1\}$ is some explicit function in P.

([Komargodski & Raz (2013)], [Komargodski, Raz & Tal (2013)])

Formula Lower Bounds

The best known lower bound for formulas over full binary basis:

Theorem [Nechiporuk (1966)]

$$L_{\text{full}}(\text{ED}) = \Omega\left(\frac{n^2}{\log n}\right)$$

$L_{\text{full}}(f)$ = size of minimum formula over full binary basis computing f

It is known $L_{\text{full}}(\text{ED}) = O(n^2/\log n)$, i.e., the bound is tight.

AC^0 circuit vs. Parity

Theorem [Ajtai (1983), Furst, Saxe & Sipser (1984)]

Parity $\notin \text{AC}^0$

Theorem [Smolensky (1987)]

Parity $\notin \text{AC}^0[\text{Mod}_p]$ for any prime $p > 2$

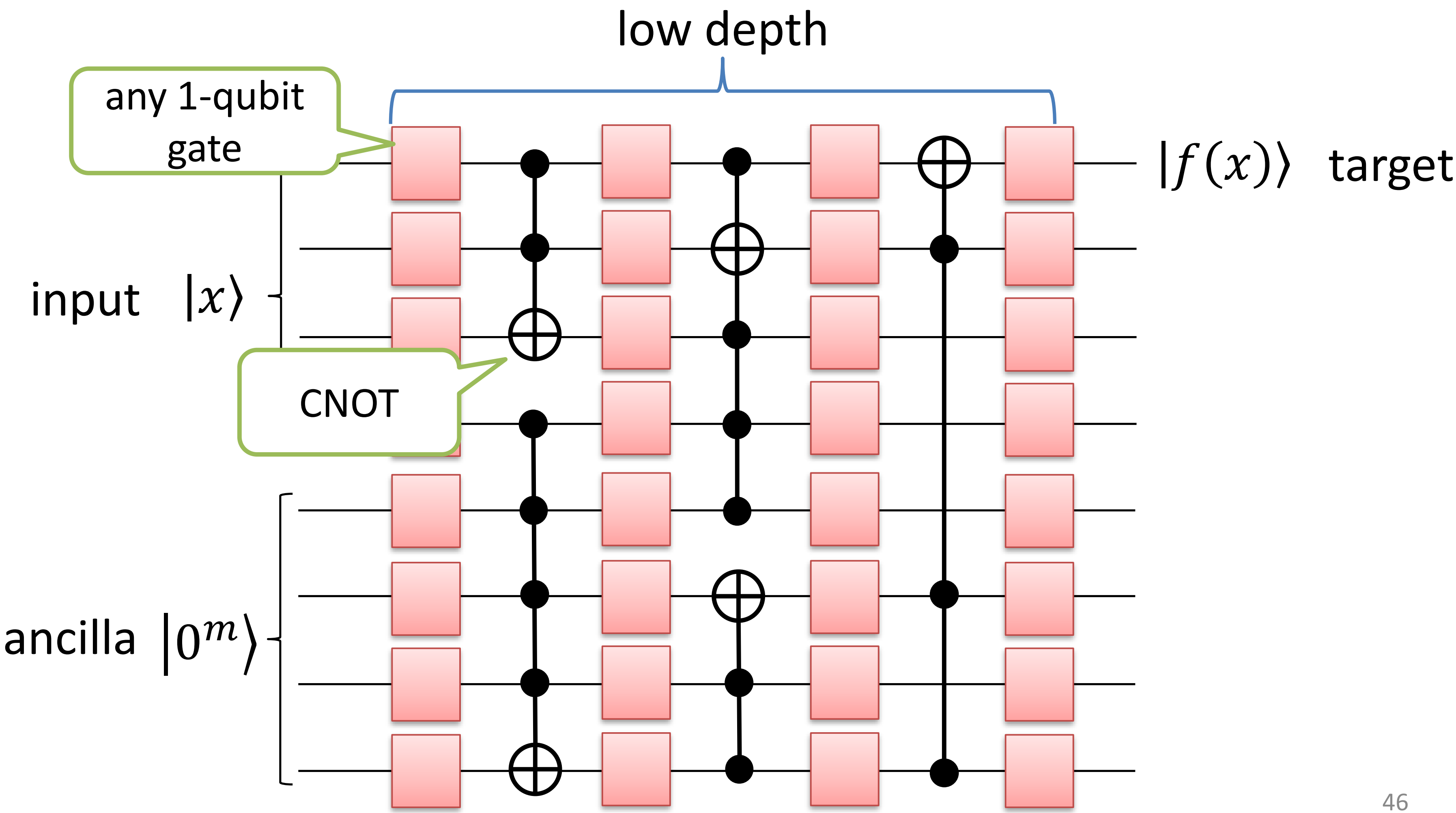
The power of $\text{AC}^0[\text{Mod}_m]$ was NOT known for a composite m until Williams' result $\text{NEXP} \not\subset \text{ACC}^0$.

Overview

1. Circuit lower bounds in high complexity classes
2. Circuit lower bounds in low complexity classes
3. Quantum circuit lower bounds
4. Proof techniques for circuit lower bounds

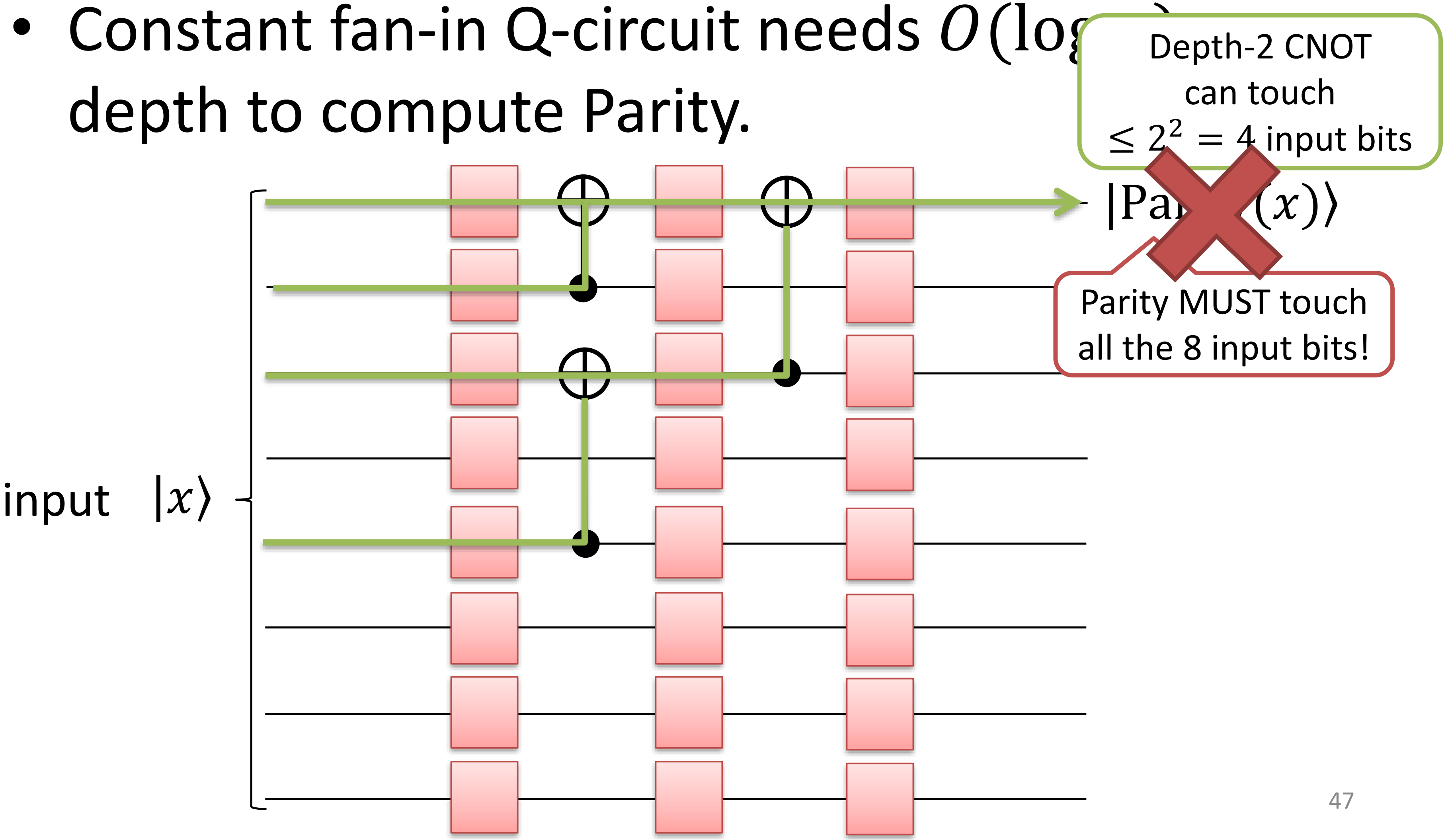
QAC^0 circuit

Gate set = {arbitrary 1-qubit gate, (generalized) CNOT}



Can shallow quantum circuit compute Parity?

- Constant fan-in Q-circuit needs $O(\log n)$ depth to compute Parity.



Quantum Circuit Lower Bounds for Parity

Conjecture:

No poly-size QAC^0 circuit of unbounded ancilla can compute Parity.

Theorem [Fang, Fenner, Green & Zhang (2006)]

No depth-$o(\log n)$ QAC^0 circuit of $\boldsymbol{o(n)}$ ancilla qubits can compute Parity.

Theorem [Pade, Fenner, Grier & Thierauf (2020)]

No **depth-2** QAC^0 circuit of **unbounded** ancilla qubits can compute Parity.

Quantum Supremacy in Shallow circuits

Theorem [Bravyi, Gosset & Koenig (2018)]

∃search problem (named "2D hidden linear function"):

- const-depth Q-circuit of bounded fan-in gates can solve,
- no $o(\log n)$-depth circuit of bounded fan-in gates can solve.

Improved by [Le Gall (2019)], [Coudron, Stark & Vidick (2018)], [Bene Watts, Kothari, Shaeffer & Tal (2019)]

Overview

1. Circuit lower bounds in high complexity classes
2. Circuit lower bounds in low complexity classes
3. Quantum circuit lower bounds
4. Proof techniques for circuit lower bounds

Techniques for Circuit Lower Bounds in High Complexity Classes

- Karp-Lipton collapse argument
  - $\Sigma^2\mathrm{P} \cap \Pi^2\mathrm{P} \not\subset \mathrm{SIZE}[n^{100}]$ [Kannan (1982)]
  - $\mathrm{ZPP}^{\mathrm{NP}} \not\subset \mathrm{SIZE}[n^{100}]$ [Köbler & Watanabe (1997)]

- Algorithm design approaches
  - Constructing non-trivially fast CKT-SAT algorithms [Williams (2013)]

Generalization of NP

Class NP

$L \in \mathrm{NP}$

$\overset{\text{Def}}{\Longleftrightarrow}$
$x \in L \Longrightarrow \exists w\colon\ R(x, w) = 1$
$x \notin L \Longrightarrow \forall w\colon\ R(x, w) = 0$

$|w| = \mathrm{poly}(|x|)$
R: poly-time comp.

e.g., $\mathrm{SAT} \in \mathrm{NP}$

$\phi(x_1, \dots, x_n) \in \mathrm{SAT} \Longleftrightarrow \exists a_1, \dots, a_n\ \phi(a_1, \dots, a_n) = 1$

Generalization of NP

Class Σ_2P

$L \in \Sigma_2\mathrm{P}$

Def

$x \in L \Rightarrow \exists w_1 \forall w_2 : R(x, w_1, w_2) = 1$

$x \notin L \Rightarrow \forall w_1 \exists w_2 : R(x, w_1, w_2) = 0$

$|w_1|, |w_2| = \mathrm{poly}(|x|)$

R: poly-time comp.

e.g., $\Sigma_2\mathrm{SAT} \in \Sigma_2\mathrm{P}$

$\phi(x_1, \dots, x_n, y_1, \dots, y_m) \in \Sigma_2\mathrm{SAT}$

$\Leftrightarrow \exists a_1, \dots, a_n, \forall b_1, \dots, b_n\ \phi(a_1, \dots, a_n, b_1, \dots, b_m) = 1$

Generalization of NP

Class Σ_kP

$L \in \Sigma_k\mathrm{P}$

Def $\Leftrightarrow$ $x \in L \Rightarrow$

$$\exists w_1 \forall w_2 \cdots \exists w_k : R(x, w_1, \ldots, w_k) = 1$$

$x \notin L \Rightarrow$

$$\forall w_1 \exists w_2 \cdots \forall w_k : R(x, w_1, \ldots, w_k) = 0$$

$|w_1|, \ldots, |w_k| = \mathrm{poly}(|x|)$
R: poly-time comp.

Polynomial-Time Hierarchy

Class PH

$$\mathrm{PH} = \bigcup_{k \in \mathbb{N}} \Sigma_k \mathrm{P}$$

Karp-Lipton Collapse Argment

1. PH $\not\subset$ SIZE$[n^{100}]$
2. Case-Analysis
   1. NP $\not\subset$ SIZE$[n^{300}]$ ➔ Done!
   2. NP $\subset$ SIZE$[n^{300}]$ ➔ By Karp-Lipton Theorem,

      PH collapses to some class : PH = $\mathbb{C}$.

      Then, PH = $\mathbb{C} \not\subset$ SIZE$[n^{100}]$.

PH has (superlinearly) hard problems.

Theorem [Kannan (1982)]

No n^{100}-size circuit can compute some $\Sigma^4\mathrm{P}$ problem.

Problem: HARD

Given: n-bit string $x \in \{0,1$

Decide: $f_{\mathrm{HARD}}(x) = 1$?

f_{HARD} is function which no n^{100}-size circuit can compute.

$\forall C \in \{n^{100}\text{–size circuit}\}$
$\exists z \in \{0,1\}^n$:
$C(z) \neq f_{\mathrm{HARD}}(z)$

Caveat: This is not precise definition, which is complicated from technical reasons.

Collapse of PH

Theorem [Karp & Lipton (1982)]

Some n^{300}-size circuit C^* can compute SAT and C^* can be simulated by class-$\mathbb{C}$ computation

$\rightarrow$ PH = $\mathbb{C}$

Argument for CLBs

Case 1

SAT has **no** n^{300}-size circuit $\rightarrow$ NP $\not\subset$ SIZE[n^{300}]

Case 2

SAT has n^{300}-size circuit C^* $\rightarrow$ PH = $\mathbb{C}$ $\not\subset$ SIZE[n^{100}]

if C^* can be simulated in $\mathbb{C}$!

Circuit Lower Bounds from Karp-Lipton Collapse Argments

Theorem [Kannan (1982)]

No n^{100}-size circuit can compute some $\Sigma^2\mathrm{P} \cap \Pi^2\mathrm{P}$ problem.

Theorem [Köbler & Watanabe (1997)]

No n^{100}-size circuit can compute some $\mathrm{ZPP}^{\mathrm{NP}}$ problem.

Techniques for Circuit Lower Bounds

- Random restriction [Furst, Saxe, & Sipser (1984)]
  - $\mathrm{Parity} \notin \mathrm{AC}^0$
  - Variant applies to quantum circuit lower bound for Parity [Fang, Fenner, Green, Homer & Zhang (2003)]

- Razborov-Smolensky argument [Razborov (1987), Smolensky (1987)]
  - $\mathrm{Parity} \notin \mathrm{AC}^0$
    - $\mathrm{Parity}(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$
  - $\mathrm{Parity} \notin \mathrm{AC}^0[\mathrm{Mod}_3]$
    - $\mathrm{AC}^0[\mathrm{Mod}_3]$ = AC^0 that allows Mod_3 gates

Razborov-Smolensky Argument

1. Parity: $\{+1,-1\}^n \to \{+1,-1\}$ (in Fourier basis) is high-deg poly.

$$\mathrm{Parity}(x_1, \dots, x_n) = x_1 x_2 \cdots x_n$$

2. AC^0 circuit is well-approximable by low-deg poly.
(Domain conversion is easy: $x' = 2x - 1$ for $x \in \{0,1\}, x' \in \{+1,-1\}$)

3. Suppose AC^0 circuit can compute Parity.
➔ Parity has impossibly good approx.
w/ low-deg poly.

Contradiction!

Note: this can show $\mathrm{Parity} \notin \mathrm{AC}^0[\mathrm{Mod}_3]$, too.

Polynomial Representations

- Polynomial representations (over $\{0,1\}^n$)
  - $\mathrm{AND}(x_1,, \ldots, x_n) = x_1 \cdots x_n$
  - $\mathrm{OR}(x_1,, \ldots, x_n) = 1 - (1 - x_1) \cdots (1 - x_n)$

degree n

- $(1 - \epsilon)$-approx. polynomial representations
  - Random subset $\{x_{i_1}, \ldots, x_{i_m}\}$ of size $m = \epsilon^{-1} \log n$
  - $\widetilde{\mathrm{AND}}(x_1,, \ldots, x_n) = x_{i_1} \cdots x_{i_m}$
  - $\widetilde{\mathrm{OR}}(x_1,, \ldots, x_n) = 1 - \left(1 - x_{i_1}\right) \cdots \left(1 - x_{i_m}\right)$

degree $\epsilon^{-1} \log n$

- $\Pr\left[\mathrm{AND}(x) \neq \widetilde{\mathrm{AND}}(x)\right] \leq \epsilon$

degree $(\epsilon^{-1} \log n)^2$

- $\Pr\left[\mathrm{AND}(\mathrm{OR}(x), \ldots) \neq \widetilde{\mathrm{AND}}\left(\widetilde{\mathrm{OR}}(x), \ldots\right)\right] \leq 2\epsilon$
- Depth-d s-size circuit can be $\Omega(1)$-approximated by deg-$O\left((\log s)^{2d}\right)$ polynomial.

Algorithm Design Approaches

- [Williams (2010, 2014), Murray & Williams (2018)]
  - Constructing fast algorithms for CKT-SAT yields CLBs!
- [Impagliazzo & Kabanets (2004), Gutfreund & K (2010)]
  - Derandomizing some randomized algorithms yields CLBs!
- [Kabanets et al. (2013)]
  - Compressing truth tables yields CLBs!
- [Fortnow & Klivans (2004), Klivans et al. (2013)]
  - Constructing good learning algorithms yields CLBs!

Concluding Remarks

- See my survey papers:
  - K, “Proving Circuit Lower Bounds in High Uniform Classes,” Interdisciplinary Information Sciences 20(1): 1-26, 2014.
  - K, “Circuit Lower Bounds from Learning-theoretic Approaches,” Theoretical Computer Science, 733: 83-98, 2018.
- New techniques beyond barrier results?
  - Relativization barrier [Baker, Gill & Solovay (1975)]
  - Natural-proof barrier [Razborov & Rudich (1997)]
  - Algebrization barrier [Aaronson & Wigderson (2009)]