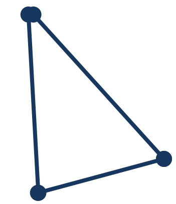
The Quantum Query Complexity of Read-Many Formulas

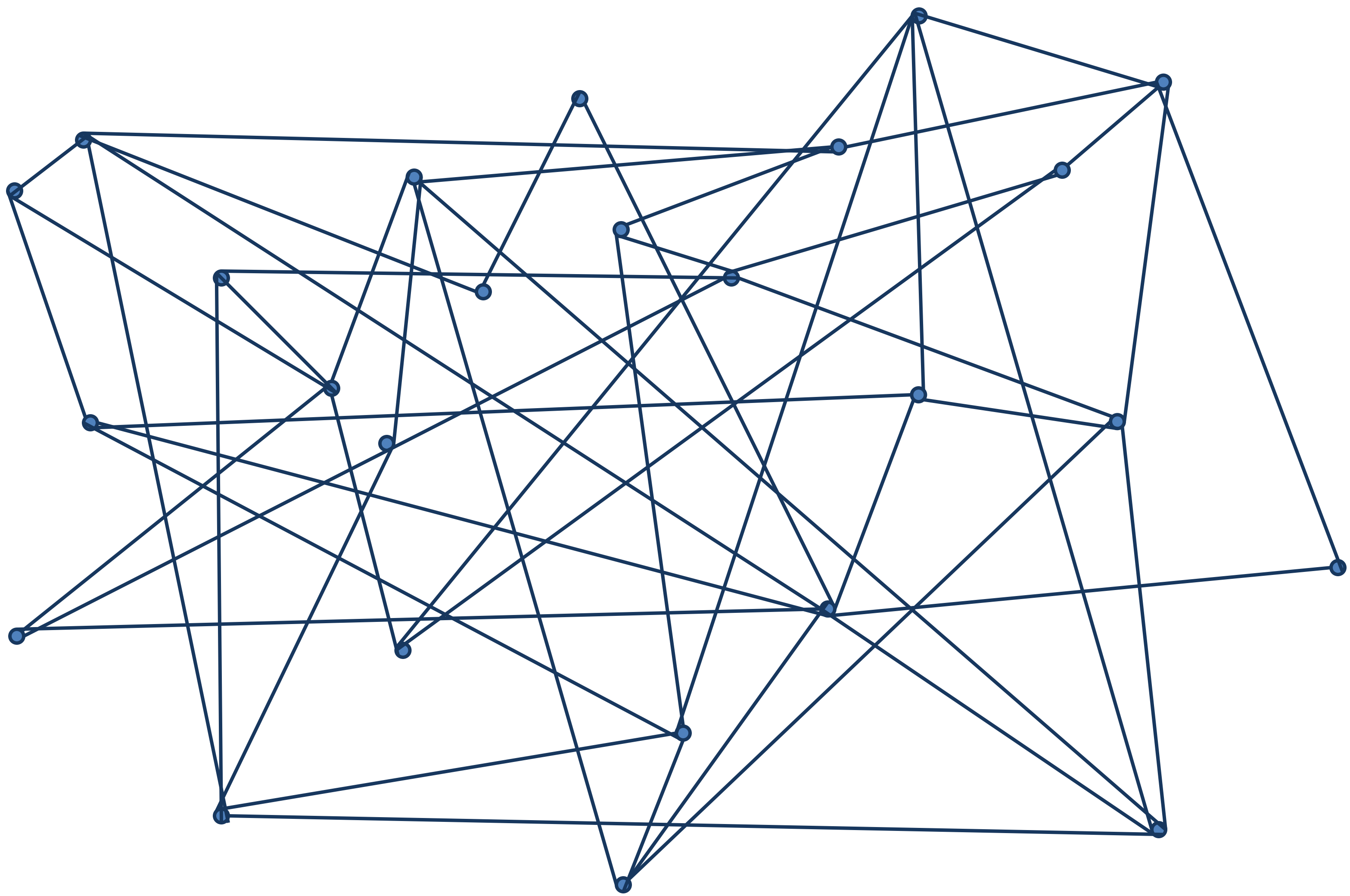
Andrew Childs, Shelby Kimmel, Robin Kothari

arXiv:1112.0548

Is there a Triangle?

$\approx$





Optimal Quantum Algorithm Unknown:

Problem	**Lower Bound (QQC)**	**Upper Bound (QQC)**
Triangle Problem n vertex graph $\rightarrow$ triangle? [Belovs, '11]	$\Omega(n)$	$O(n^{35/27})$
K-distinctness n integers $\rightarrow \geq k$ of them equal? [Belovs and Lee, 2011]	$\Omega(n^{2/3})$	$O(n^{k/(k+1)})$
Boolean Matrix Product Verification A, B, C are $n \times n$ Boolean matrices $\rightarrow A \times B = C$? [Buhrman and Spalek, 2006]	$\Omega(n)$	$O(n^{1.5})$

Optimal Quantum Algorithm Unknown:

Problem
Triangle Problem n vertex graph → triangle? [Belovs, ‘11]
K-distinctness n integers →≥ k of them equal? [
Boolean Matrix Product Verification A, B, C are $n \times n$ Boolean matrices → $A \times B = C$?

Boolean Formulas

Shelby Kimmel (that's me!)
Robin Kothari and Andrew Childs (University of Waterloo)

Problem:

How hard to evaluate Boolean formulas with a quantum computer? (in general and for some specific problems)

Result:

Optimal algorithm for general Boolean formulas

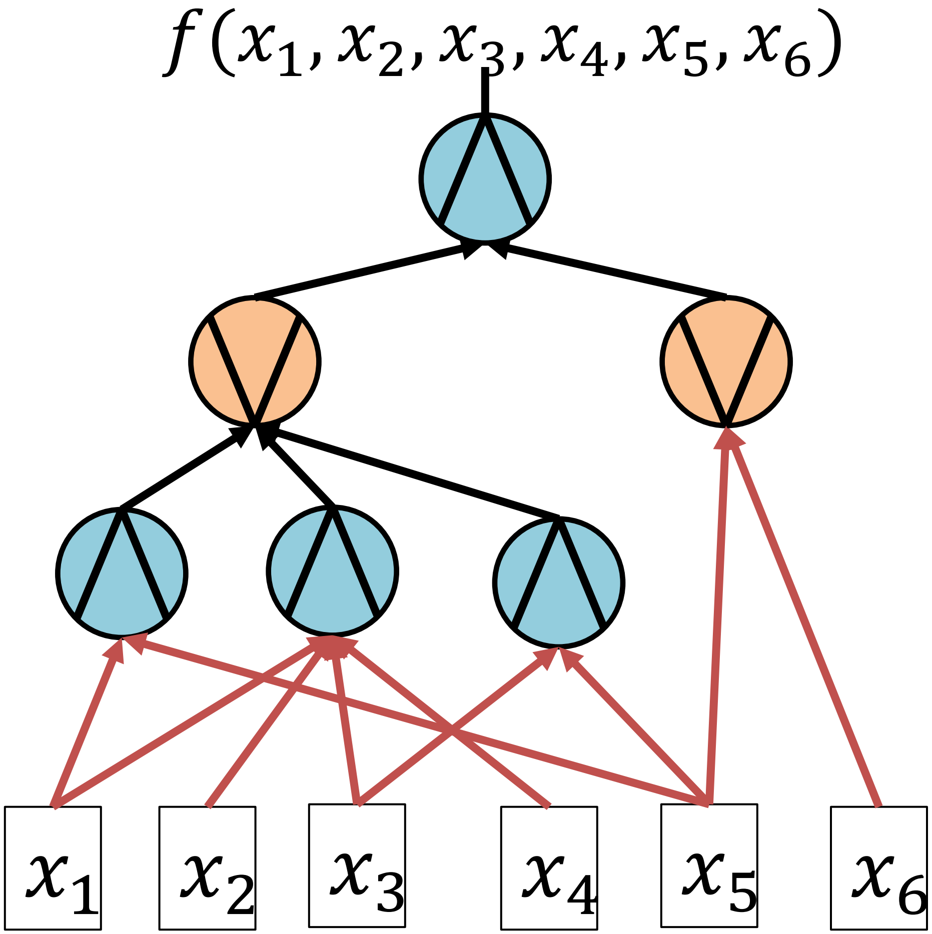
Also:

- Almost optimal algorithm for constant depth Boolean formulas
- Better bounds for Boolean Matrix Product Verification
- Applications to classical circuit complexity

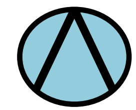
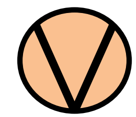
Outline

1. Intro to Boolean formulas and quantum query complexity (QQC)
2. Optimal algorithm for Boolean formulas
3. Applications and Extensions
   a) Constant depth formulas
   b) Boolean Matrix Product Verification
   c) Classical Circuit Complexity

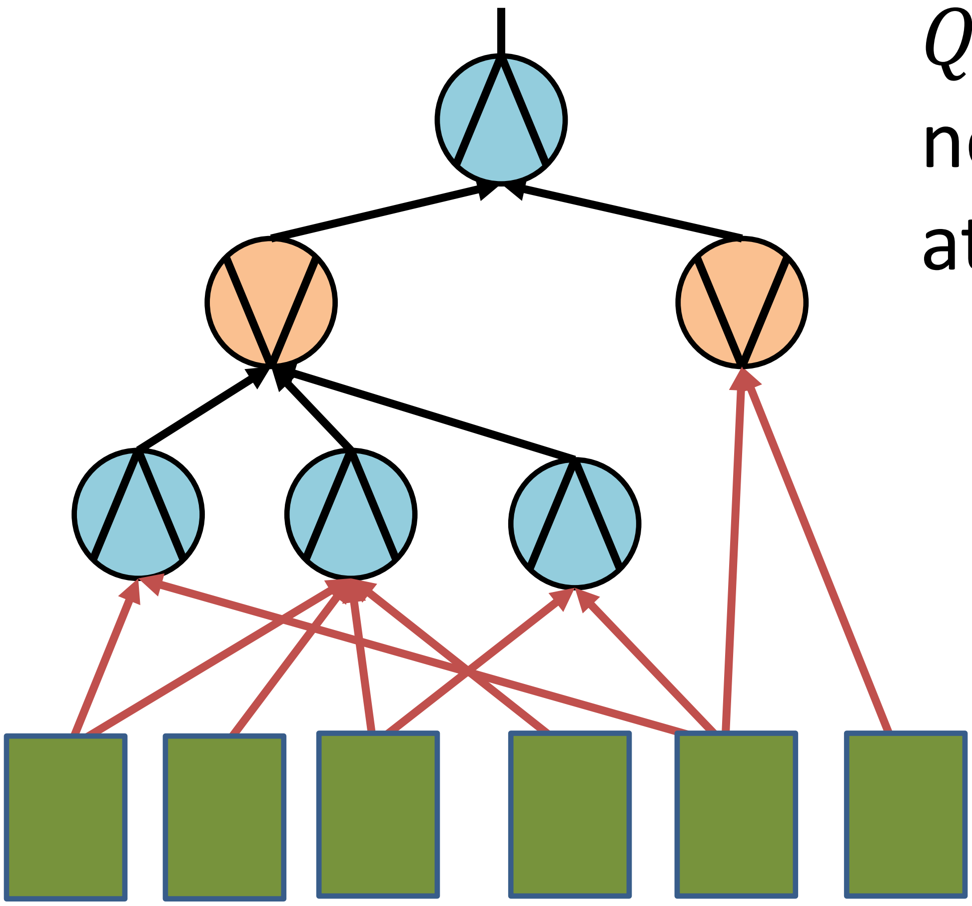
General Boolean Formula



Boolean: $x_i \in \{0,1\}$

- Unbounded Fan-in
  - AND 
  - OR 
- No fanout of gates
- Fanout of inputs OK
- n =# of inputs
- S =# of input edges
- G =# of gates

Quantum Query Complexity

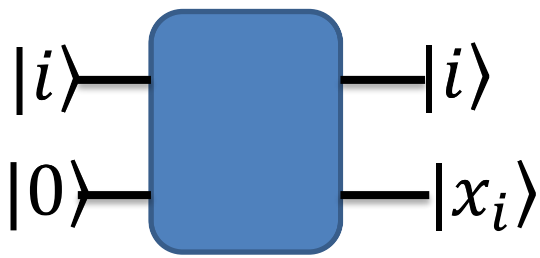


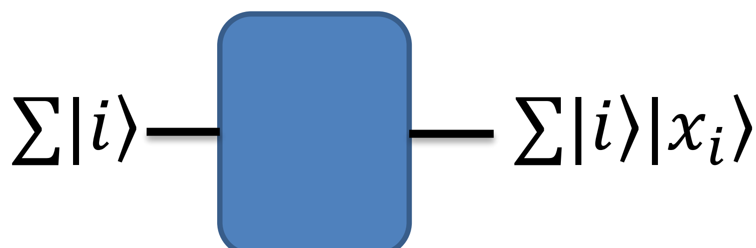
$Q(f)$ = # of inputs need to "query," look at (quantumly)

Quantum Query Complexity
(How hard is a problem?)

Goal: Determine the value of $f(x_1, \ldots, x_n)$ for a known function f, with an black box for *x*

Quantum Black Box



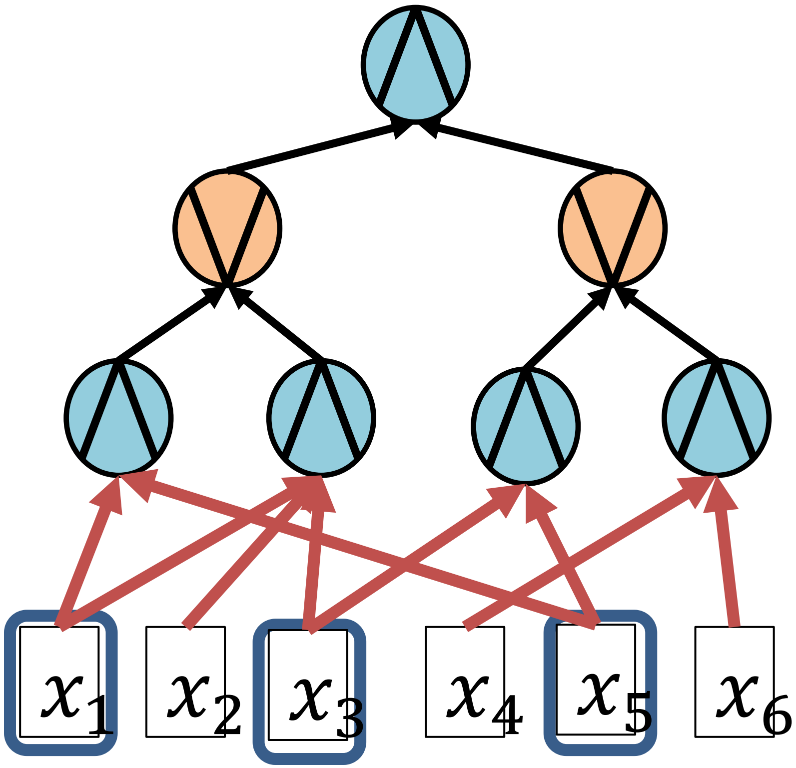


Only care about # of uses of Black Box (queries)

$\mathrm{Q}(f)$
(bounded error quantum query complexity)

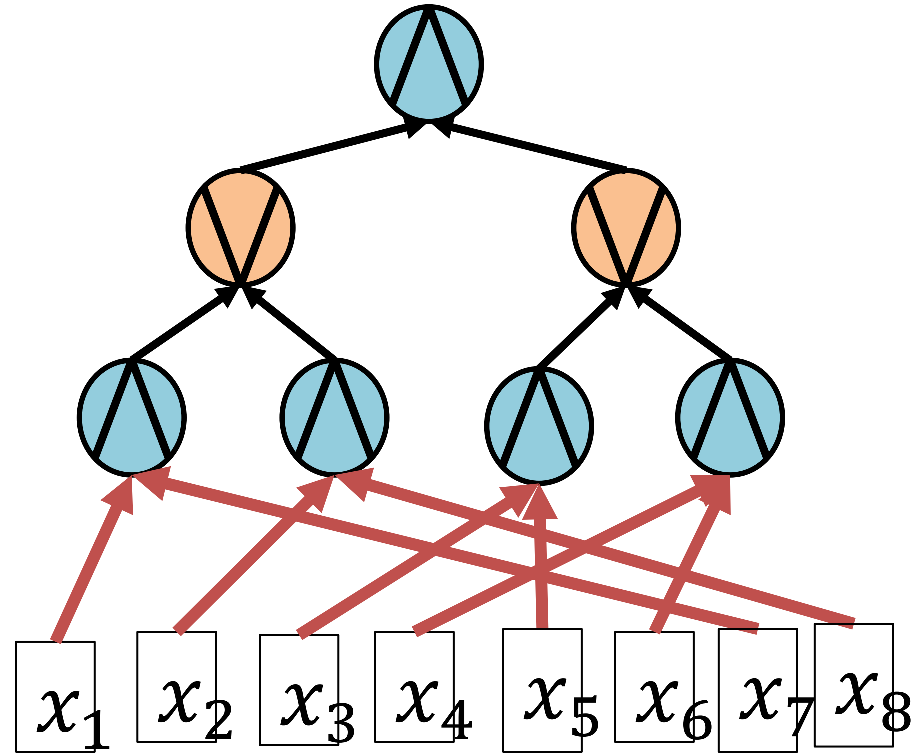
General vs. Read-Once Formulas

“Read-many” = general



$Q(f) =?$

Read-Once



$Q(f) = \sqrt{S}$

S =# of / edges

New Bounds on Formula Quantum Query Complexity

Upper Bound: We design algorithm to evaluate any Boolean formula with quantum query complexity

$$\boxed{O(\min\{n, S^{1/2}, n^{1/2}G^{1/4}\})}$$

Lower Bound: Given values for n, S, and G, there exists an formula with query complexity

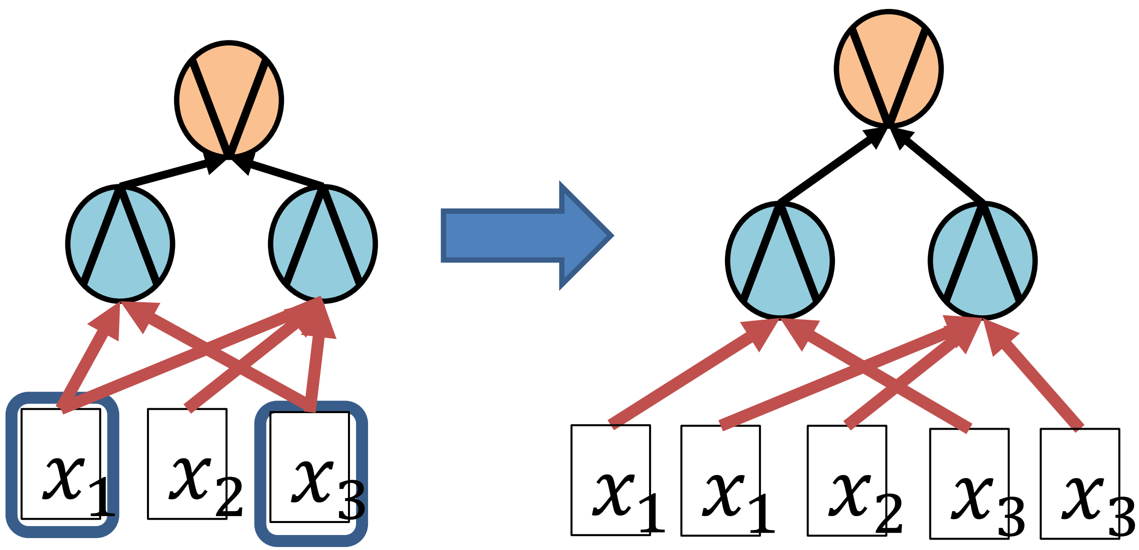
$$\boxed{\Omega(\min\{n, S^{1/2}, n^{1/2}G^{1/4}\})}$$

$n =$# of inputs, $S =$# of input edges, $G =$# of gates

Big Idea: Algorithm

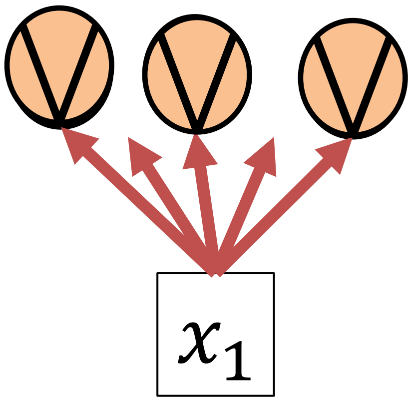
$$\mathrm{Q(f)} = \mathrm{O}(\min\{n, S^{1/2}, n^{1/2}G^{1/4}\})$$

- n – Query all inputs (trivial)
- $S^{1/2}$ – convert to read-once



n =# of inputs, S =# of input edges, G =# of gates

Big Idea: Algorithm



$$\mathrm{Q(f)} = \mathrm{O}(\min\{n, S^{1/2}, \boxed{n^{1/2}G^{1/4}}\})$$

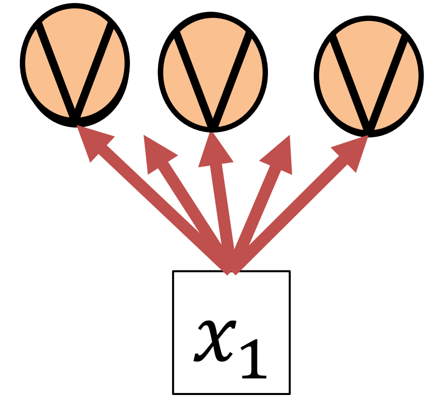
Consider “high degree” (deg $> G^{1/2}$) inputs

If $x_1 = 1$, learn output of $> G^{1/2}$ OR gates

<u>Grover’s Search</u>

- If t out of k inputs have value 1, Grover’s Search finds a 1-valued input in $O\left(\sqrt{\frac{k}{t}}\right)$ quantum queries

Big Idea: Algorithm



$$Q(f) = O(\min\{n, S^{1/2}, \boxed{n^{1/2}G^{1/4}}\})$$

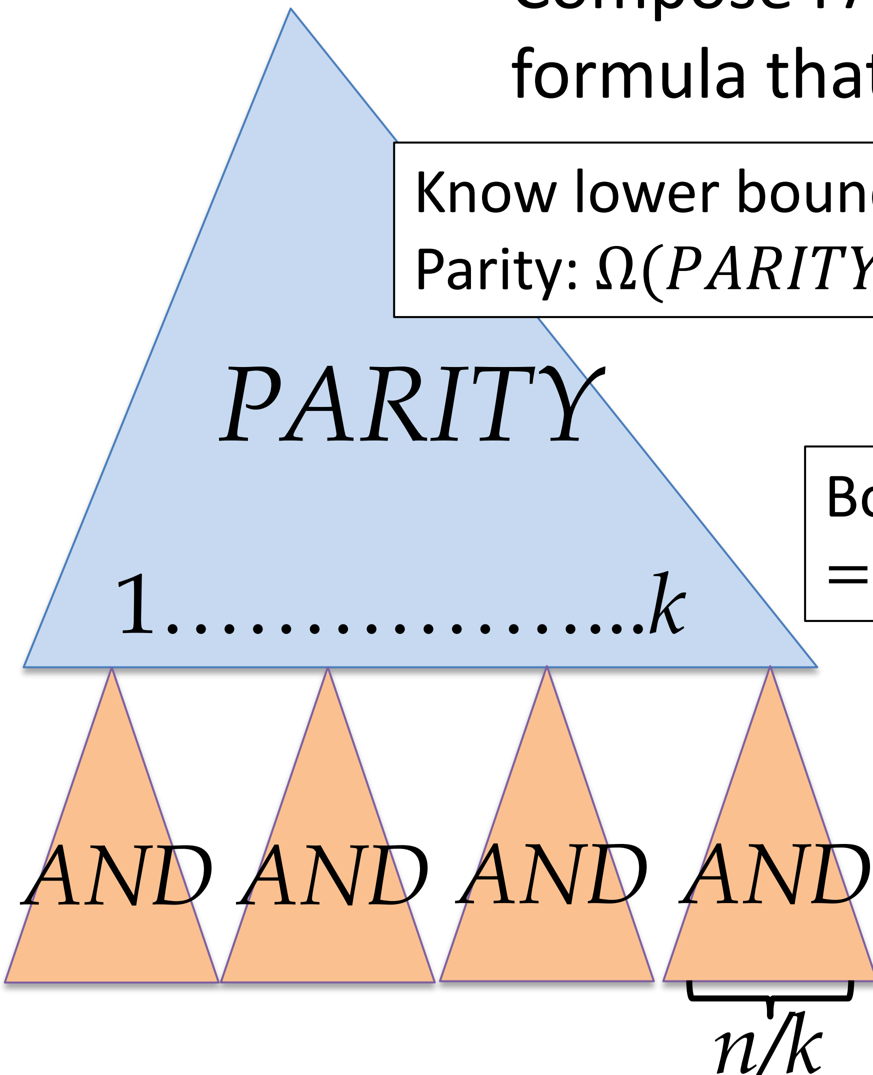
<u>Plan</u>

1. Learn all high deg nodes by Grover search: $O\left(\sqrt{k/t}\right)$
   - Many marked (t large) : many rounds, but rounds use few queries per round
   - Few marked: few rounds, rounds use more queries
2. Now S is small b/c no input is high degree
   - Expand (by repeating inputs) to Read-Once

Parts 1 & 2 each use $O\left(n^{1/2}G^{1/4}\right)$ queries!

Big Idea: Lower Bound

Compose PARITY and AND to get new formula that needs large query complexity



Know lower bound for Parity: $\Omega(PARITY)$

Know lower bound for AND: $\Omega(AND)$

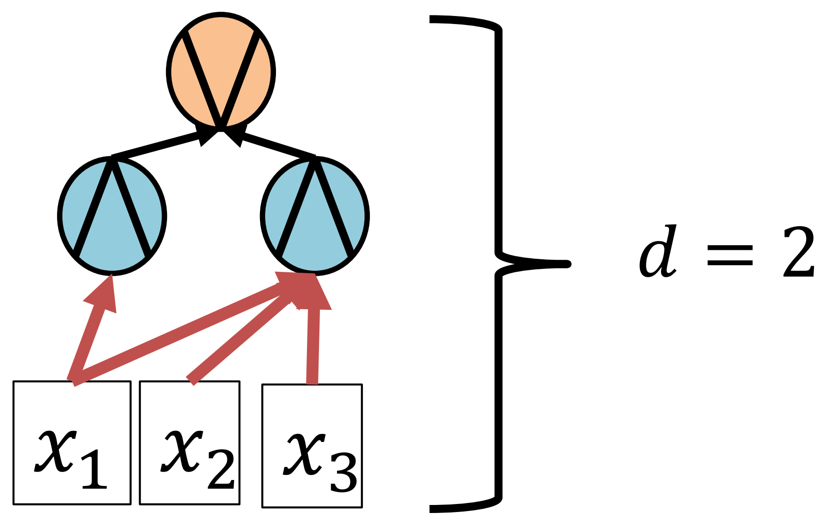
Bound on composed: [Reichardt, 2011]
$= \Omega(PARITY) \times \Omega(AND)$

By adjusting k, can get a formula w/ lower bound that matches $\Omega(\min\{n, S^{1/2}, n^{1/2}G^{1/4}\})$

Extensions: Constant Depth Formulas

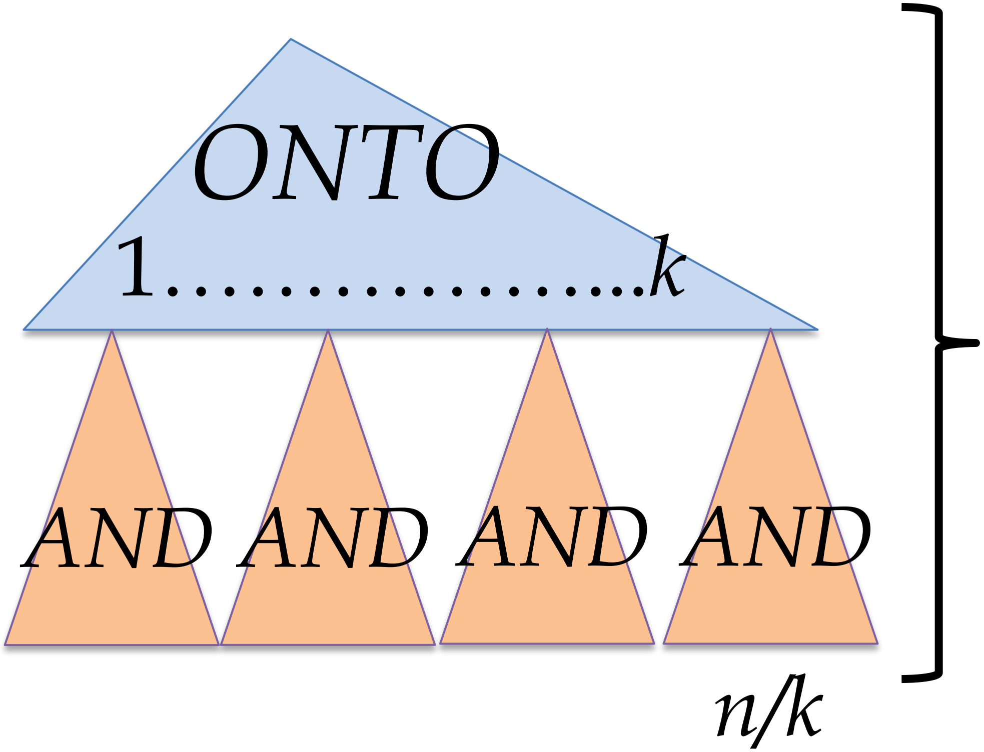
n =# of inputs, S =# of input edges, G =# of gates

What if also know that depth $= d$?



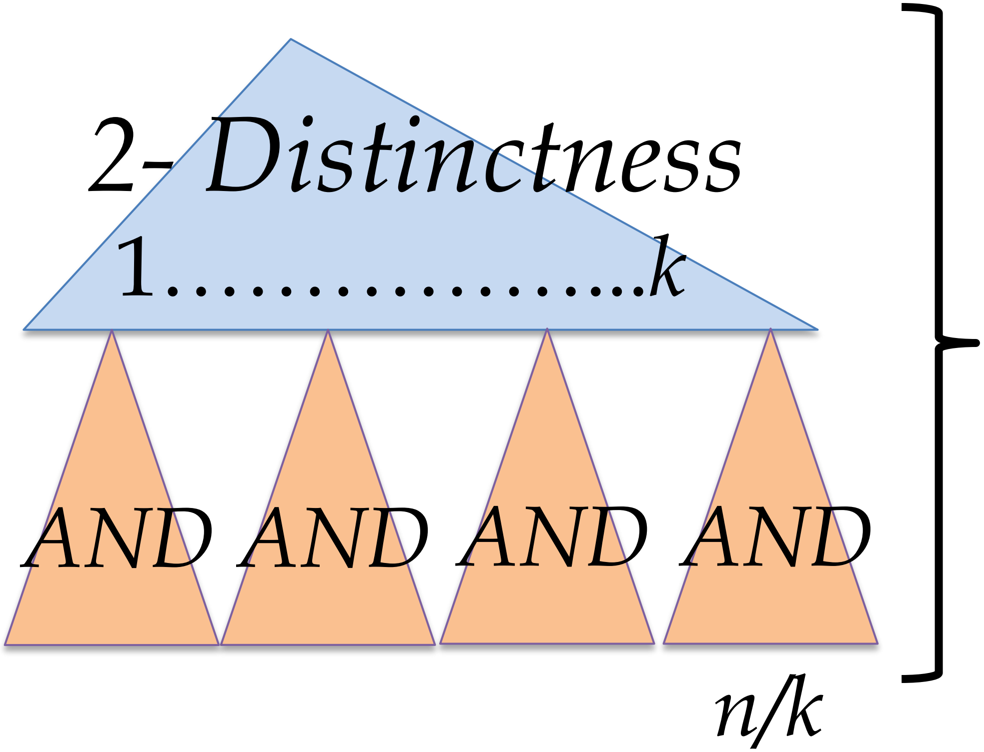
Algorithm (upper bound) holds for any depth, but lower bound uses PARITY, which has linear depth

Extensions: Constant Depth Formulas



Lower bound on query same as upper bound up to logarithmic factors for constant depth > 3!

Extensions: Constant Depth Formulas



$d = 2$

Lower bound = $\Omega(n^{.555})$

For $d = 2, G < n$, so using our algorithm, upper bound is $O(n^{1/2}G^{1/4}) = O(n^{.75})$……Not tight!

Applications: Boolean Matrix Product Verification

Recall: Boolean Matrix Product Verification

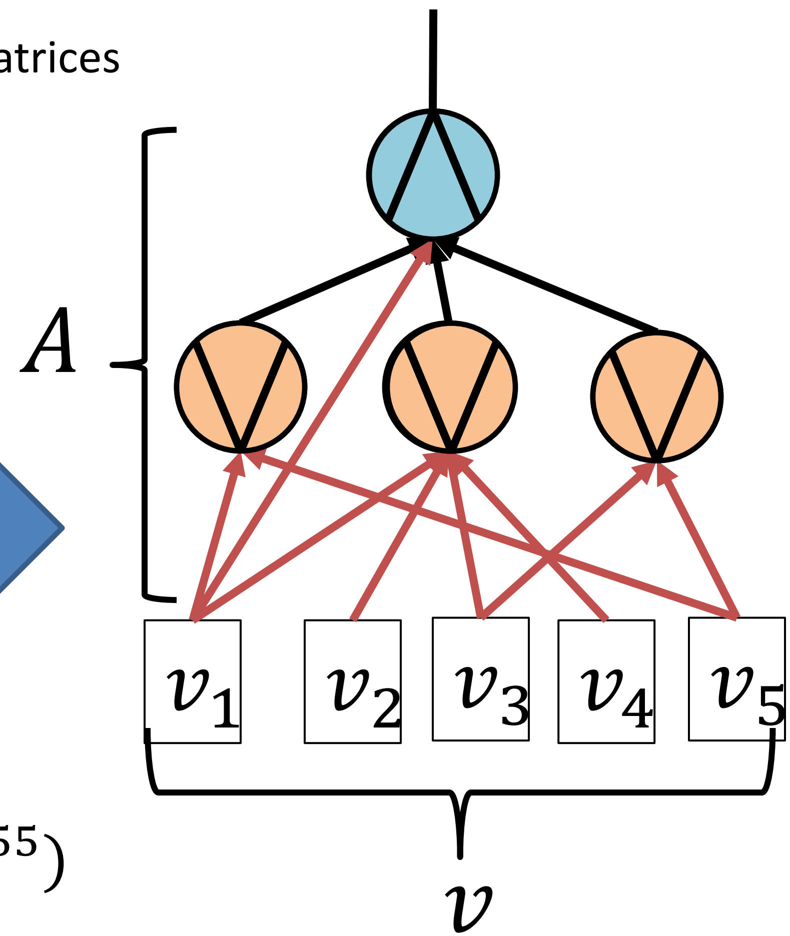
A, B, C are $n \times n$ Boolean matrices

$\rightarrow\ A \times B = C$?

Boolean ***Vector*** Product Verification

$$\begin{bmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & \ddots & \vdots \\ \vdots & \cdots & \ddots \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix} \stackrel{?}{=} \begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}$$

A known v unknown



Lower bound = $\Omega(n^{.555})$

Applications: Boolean Matrix Product Verification

$$\begin{bmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & \ddots & \vdots \\ \vdots & \cdots & \ddots \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \cdots \\ v_{21} & \ddots & \vdots \\ \vdots & \cdots & \ddots \end{bmatrix} \stackrel{?}{=} \mathbb{I} \leq$$

Boolean Matrix Product Verification (all matrices unknown)

A known $\quad$ V unknown

$$f = (AND) \circ (Boolean\,Vector\;Product\;Verification)$$

Lower bound = $\Omega(n^{1.0555})$

Optimal Quantum Algorithm Unknown:

Problem	**Lower Bound**	**Upper Bound**
Triangle Problem n vertex graph → triangle?	$\Omega(n)$	$O(n^{35/27})$
K-distinctness n integers →$\geq k$ of them equal?	$\Omega(n^{2/3})$	$O(n^{k/(k+1)})$
Boolean Matrix Product Verification A, B, C are $n \times n$ Boolean matrices $\rightarrow\ A \times B = C$?	$\Omega(n)$	$O(n^{1.5})$

Application: Classical Formula Complexity

$$Q(f) = O(n^{1/2}G^{1/4})$$

Upper bound on Query Complexity in terms of number of gates in the formula

$$G(f) = \Omega(n^{-2}Q^{4})$$

Lower bound on the number of gates in a formula in terms of the query complexity.

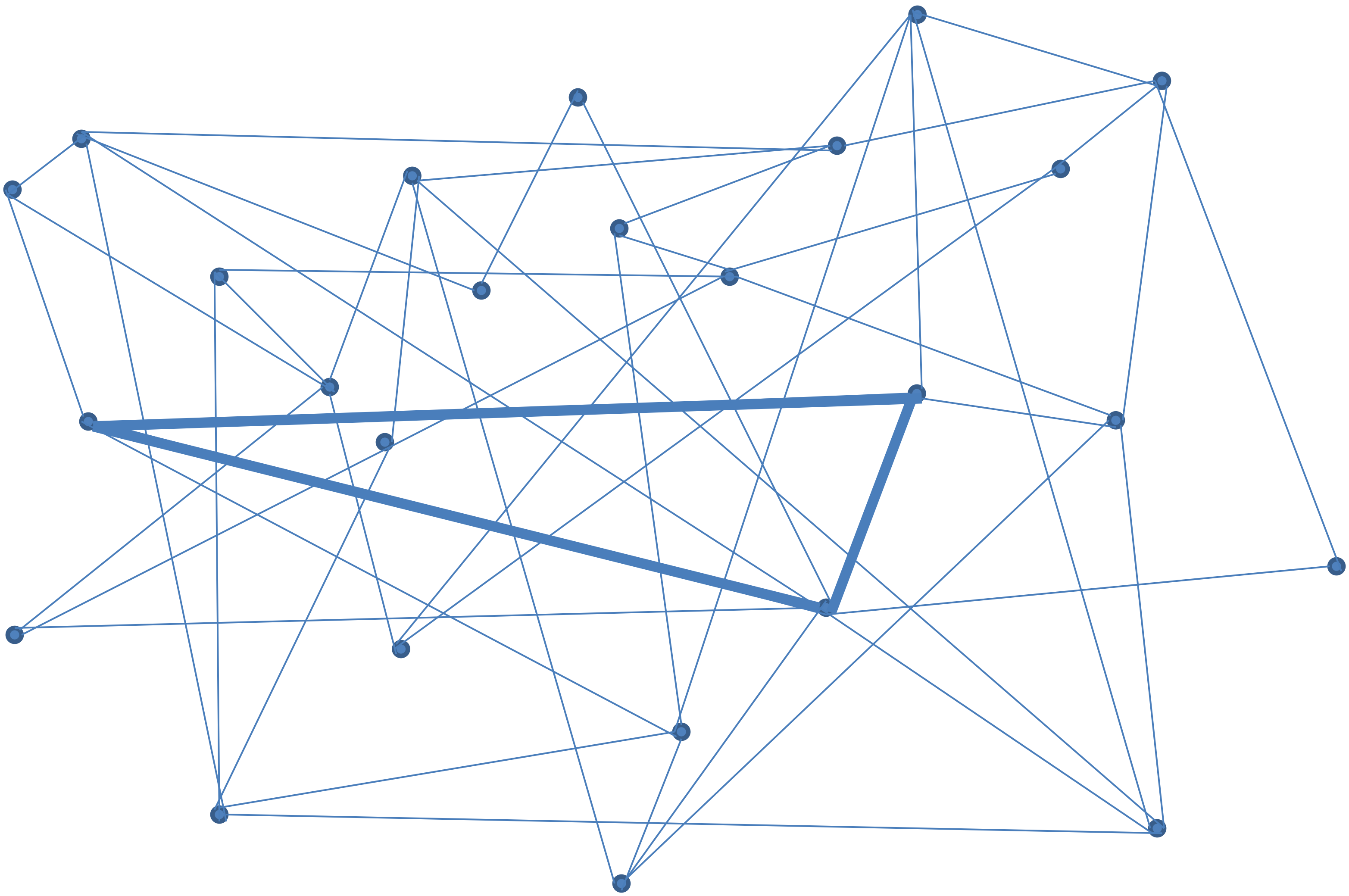
Results:

- PARITY requires n^2gates (previous best result: $S = O(n^2)$)
- Graph Planarity requires n gates (nothing known)

Recap

1. Described Boolean Formulas
2. Gave an optimal quantum algorithm for Boolean formulas
3. Improved lower bound for Boolean Matrix Product Verification
4. Gave new lower bounds on number of gates needed for classical formulas

Is there a Triangle?



The Quantum Query Complexity of Read-Many Formulas

Andrew Childs, Shelby Kimmel, Robin Kothari

arXiv:1112.0548