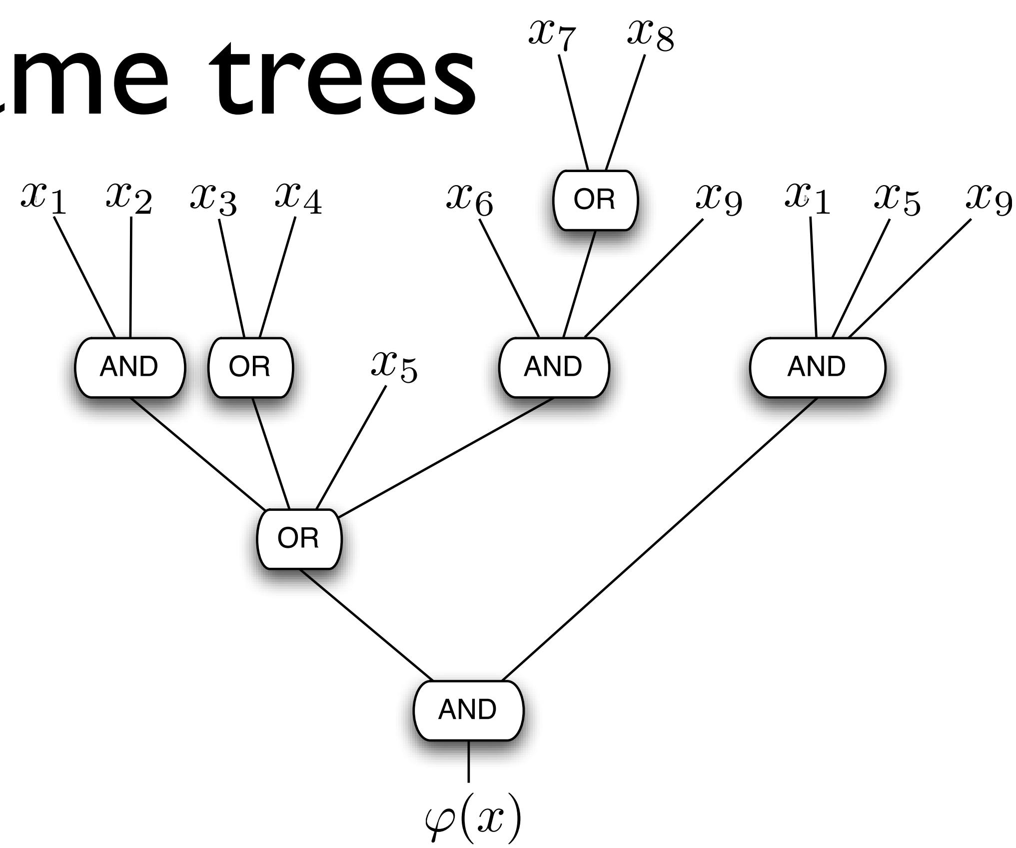
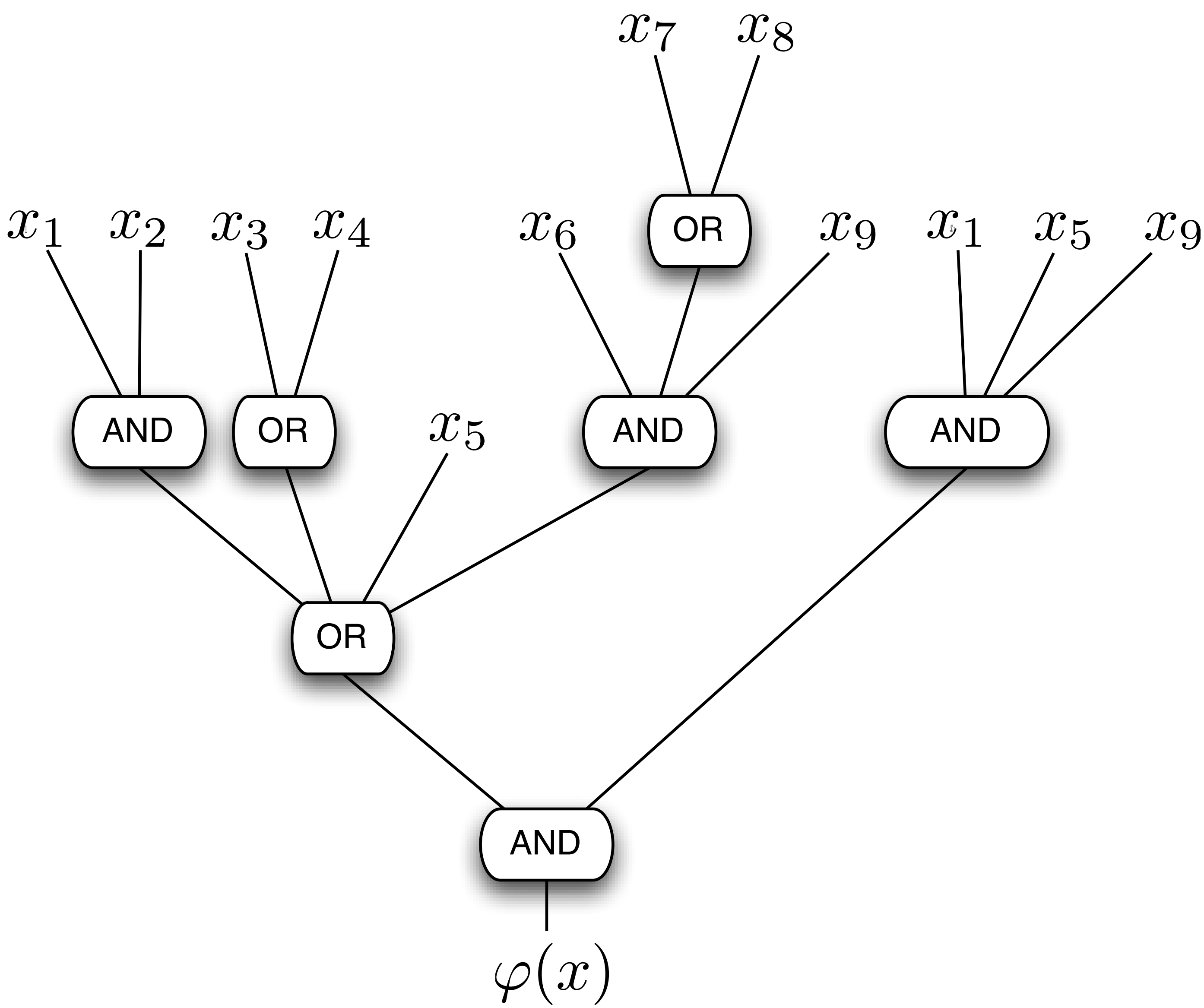
Faster quantum algorithm for evaluating game trees

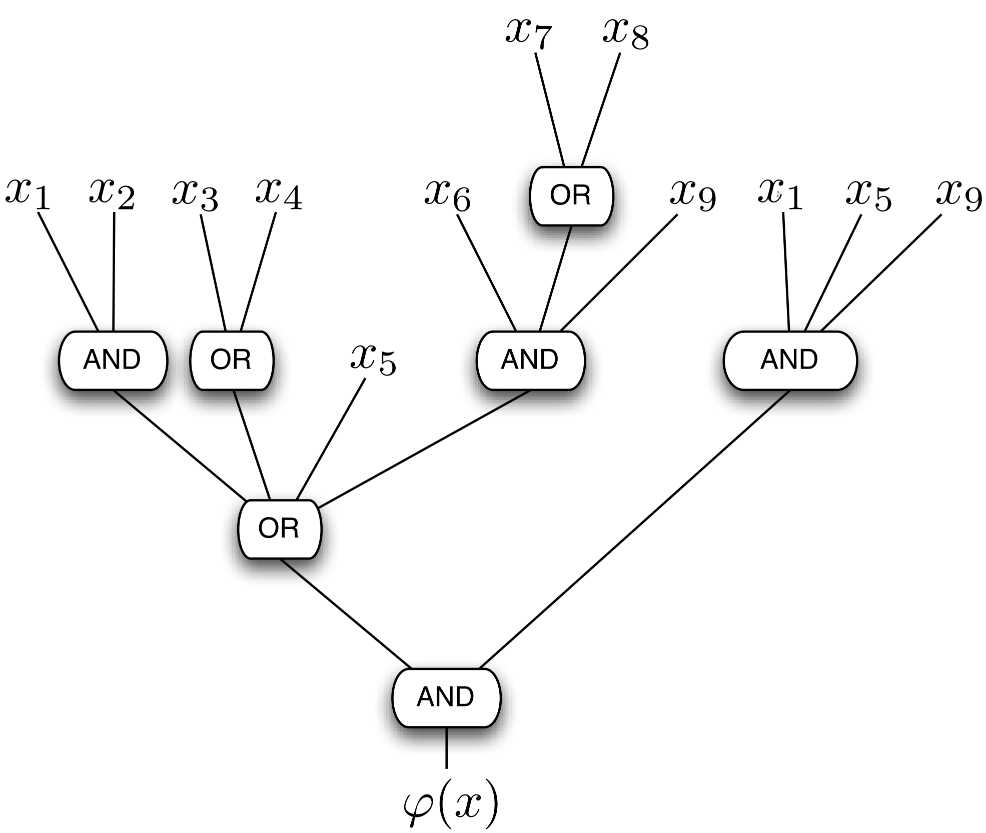
Ben Reichardt

University of Waterloo





x_7
x_8
x_1
x_2
x_3
x_4
x_6
OR
x_9
x_1
x_5
x_9
AND
OR
x_5
AND
AND
OR
AND
$\varphi(x)$



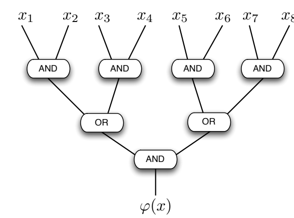
Motivations:

- Two-player games (Chess, Go, …)
  - Nodes ↔ game histories
  - White wins iff ∃ move s.t. ∀ responses, ∃ move s.t. …
- Decision version of min-max tree evaluation
  - inputs are real numbers
  - want to decide if minimax is ≥10 or not
- Model for studying effects of composition on complexity

Deterministic decision-tree complexity = N

Any deterministic algorithm for evaluating a *read-once* AND-OR formula must examine every leaf

For balanced, binary formulas



α-β pruning is optimal ⇒ Randomized complexity $N^{0.754}$

[Snir '85, Saks & Wigderson '86, Santha '95]

$N^{0.51}$ ≤ Randomized complexity ≤ N

[Heiman, Wigderson '91]

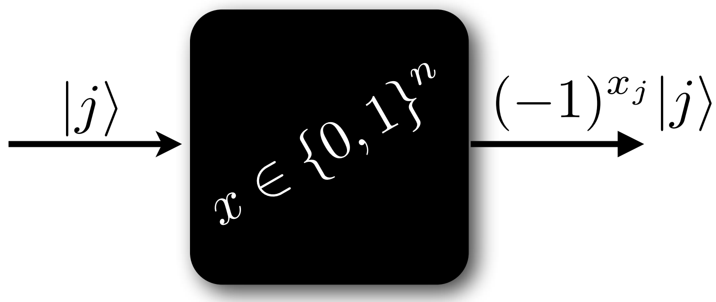
(see also K. Amano, Session 12B Tuesday)

Deterministic decision-tree complexity = N

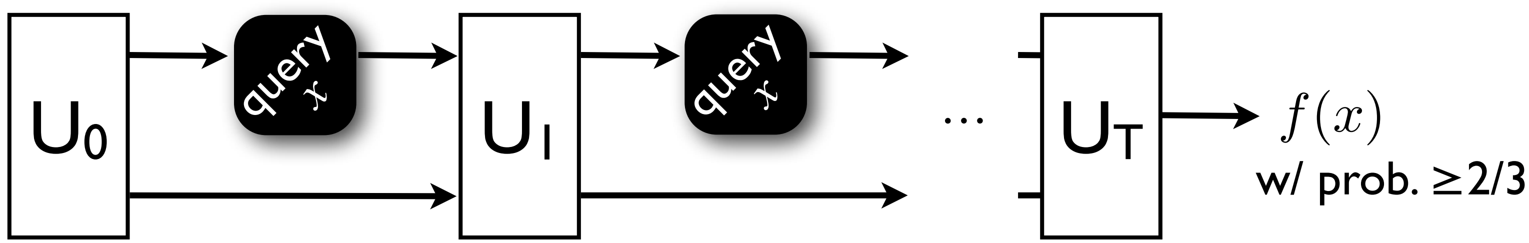
$N^{0.51} \leq$ Randomized complexity $\leq$ N

Quantum query complexity = $\sqrt{N}$

(very special case of the next talk)



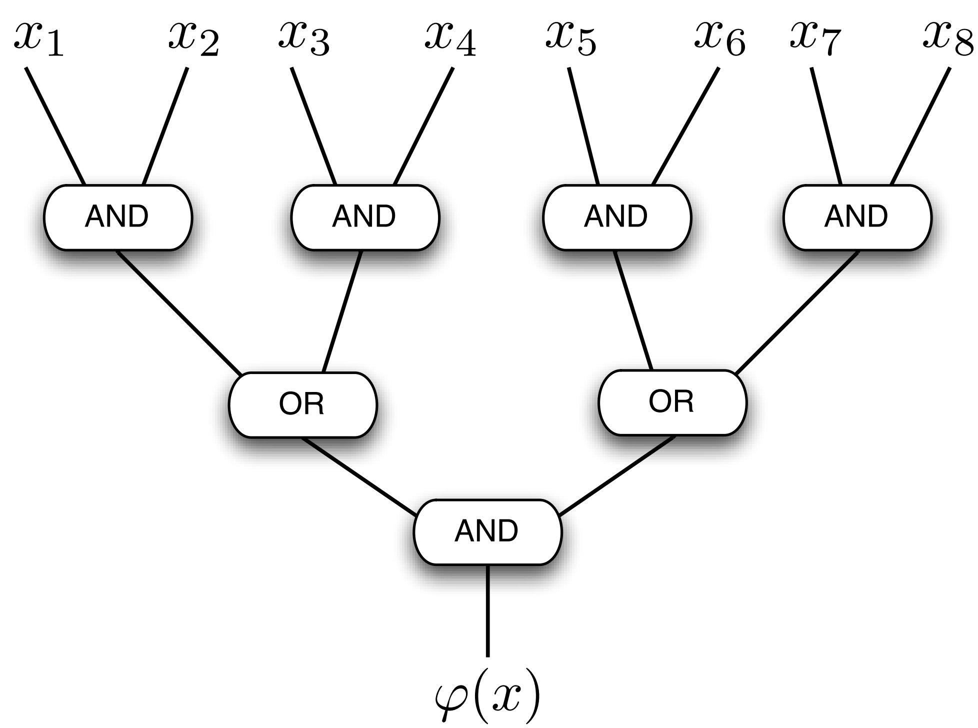
$$|1\rangle + |2\rangle \mapsto (-1)^{x_1}|1\rangle + (-1)^{x_2}|2\rangle$$



This talk: What is the time complexity for quantum algorithms?

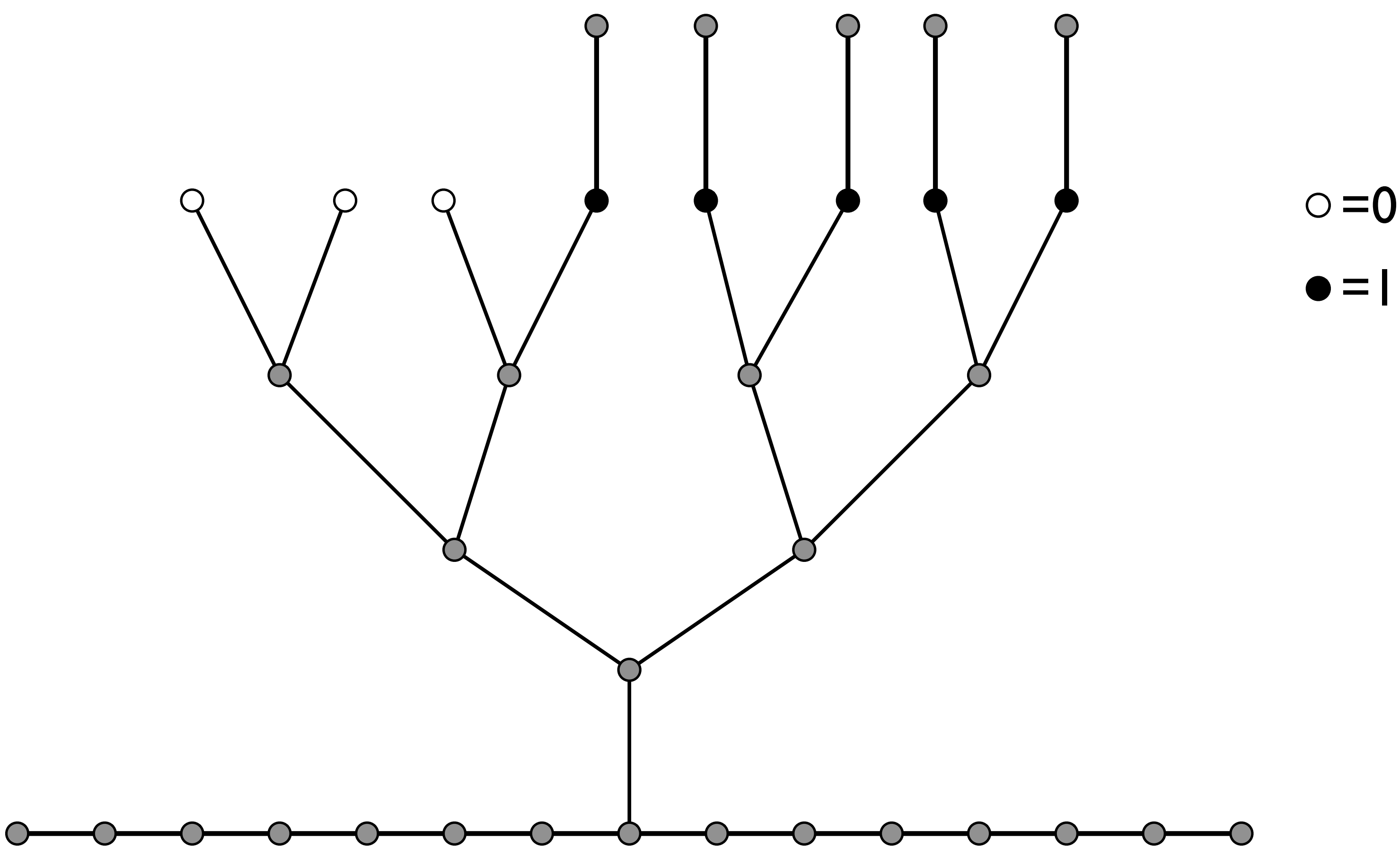
Farhi, Goldstone, Gutmann '07 algorithm

- **Theorem** ([FGG '07]): A balanced binary AND-OR formula can be evaluated in time $N^{\frac{1}{2}+o(1)}$.



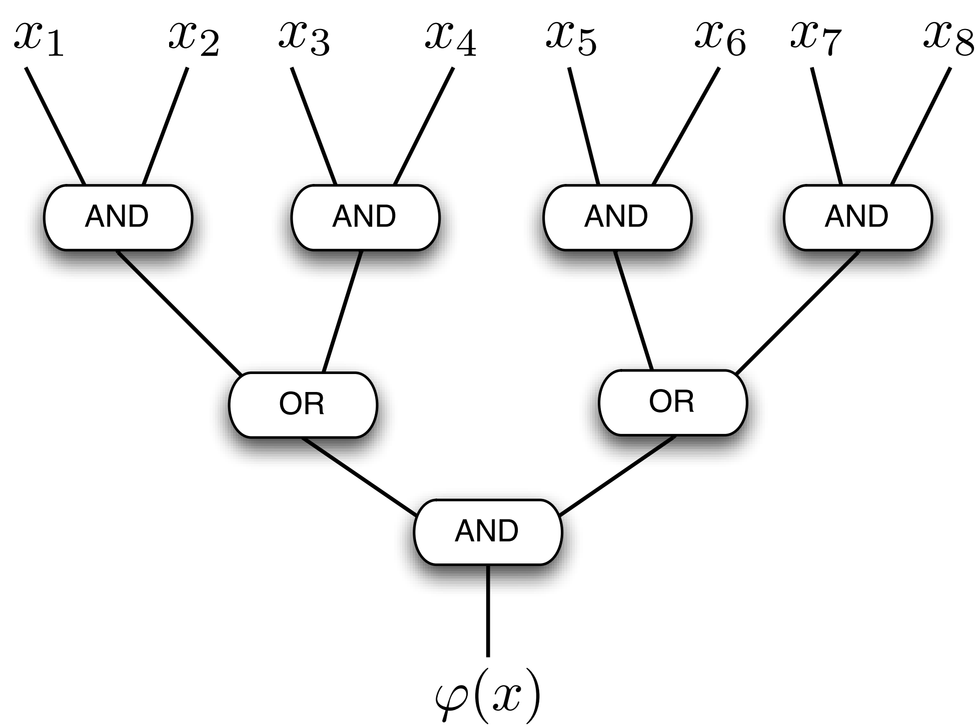
Farhi, Goldstone, Gutmann '07 algorithm

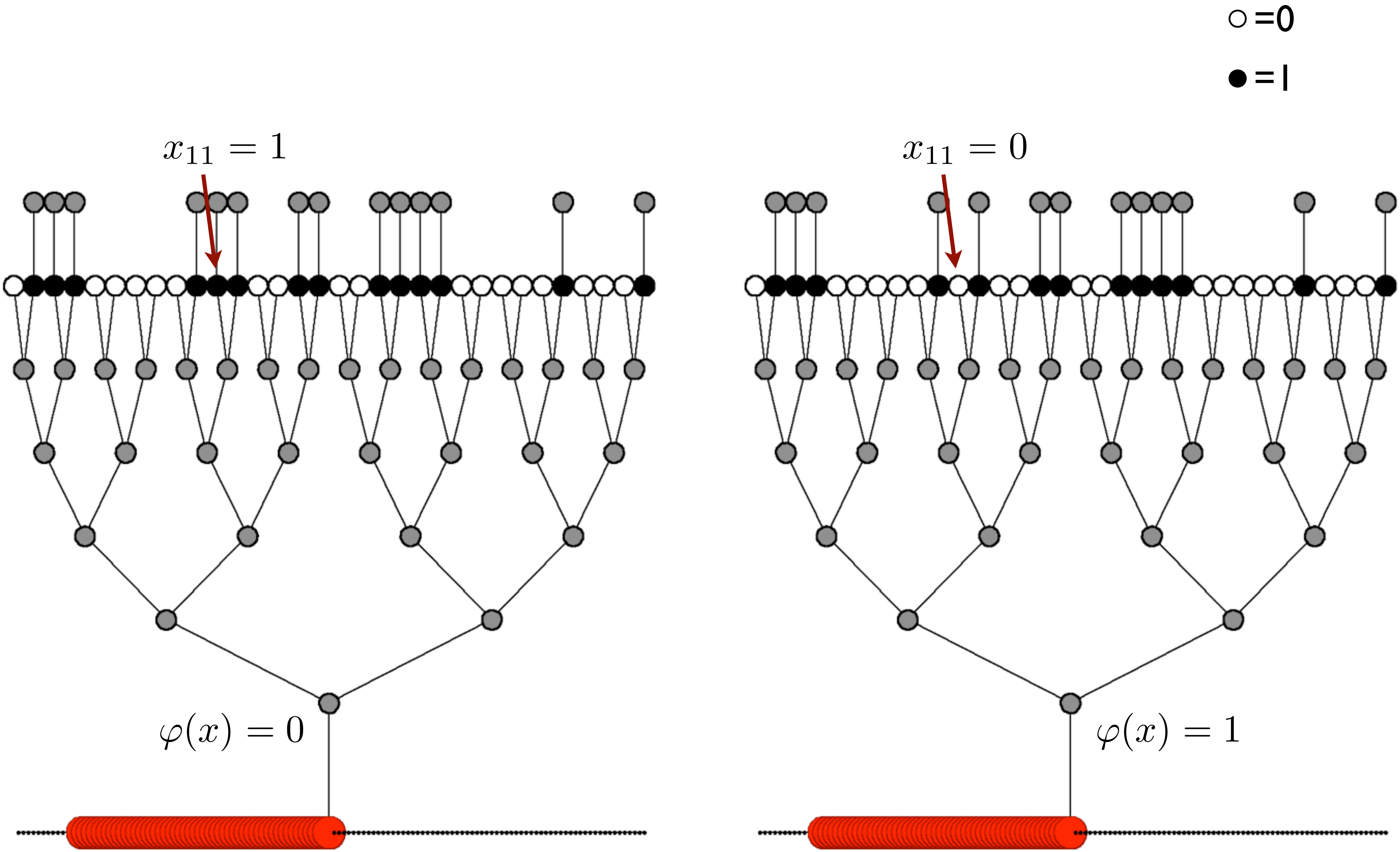
- **Theorem** ([FGG '07]): A balanced binary AND-OR formula can be evaluated in time $N^{1/2+o(1)}$.
  - Convert formula to a tree, and attach a line to the root
  - Add edges above leaf nodes evaluating to one



Farhi, Goldstone, Gutmann '07 algorithm

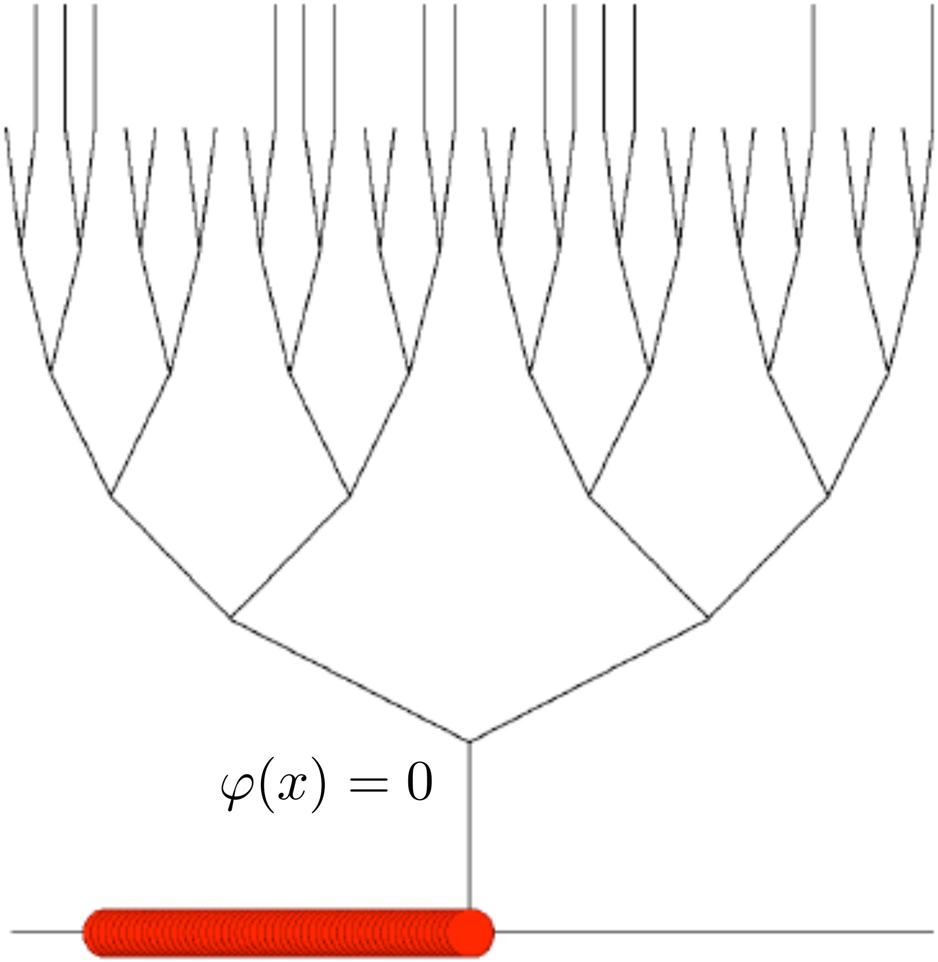
- **Theorem** ([FGG '07]): A balanced binary AND-OR formula can be evaluated in time $N^{\frac{1}{2}+o(1)}$.
  - Convert formula to a tree, and attach a line to the root
  - Add edges above leaf nodes evaluating to one

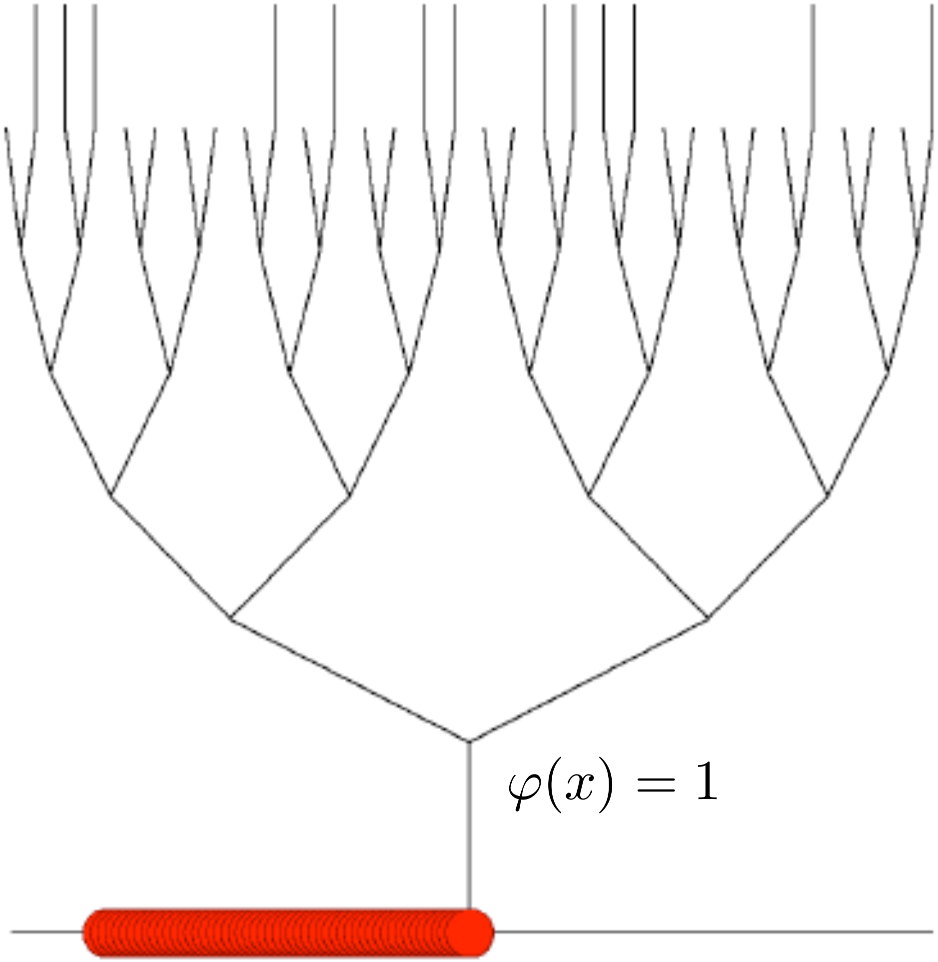




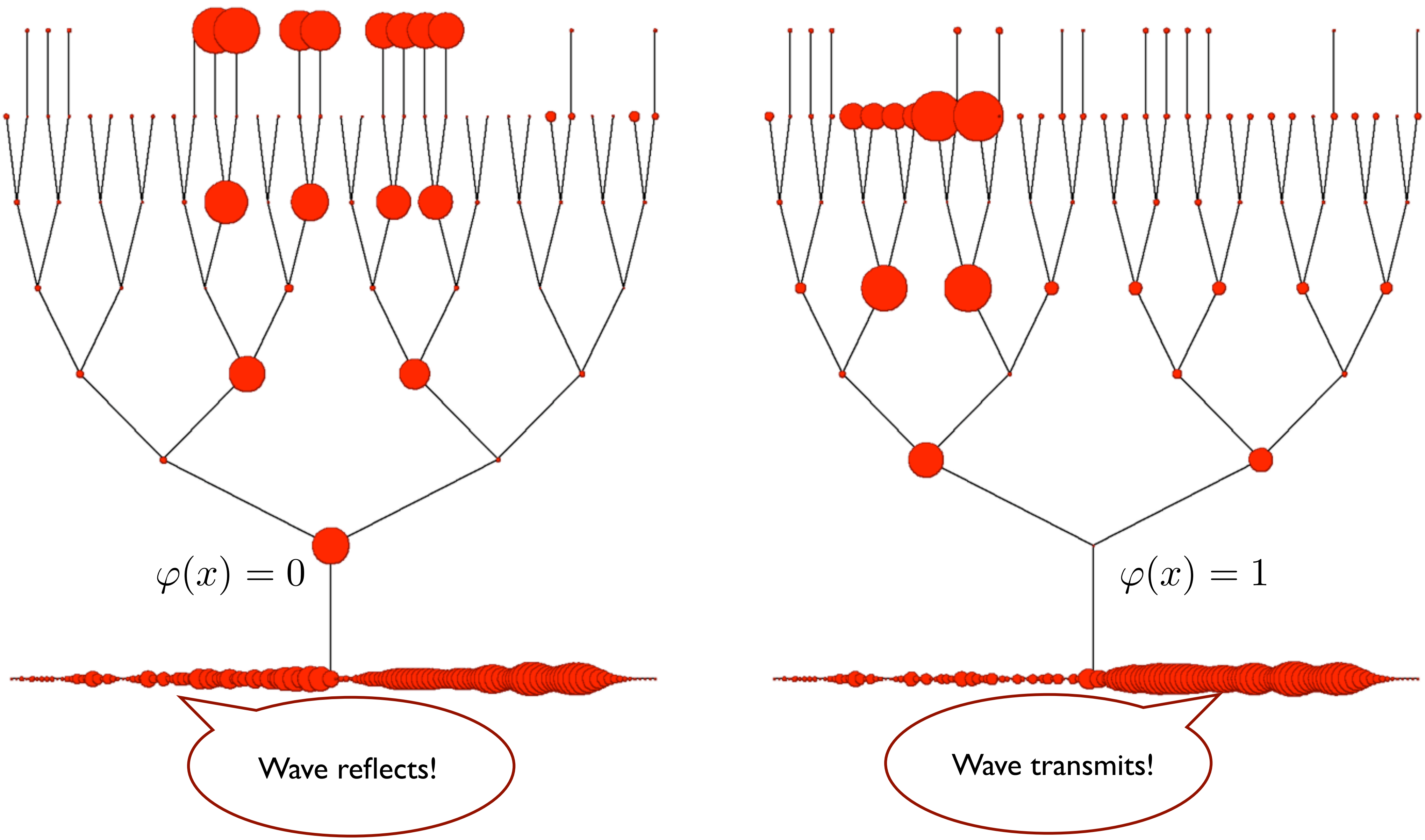
○ =0
● =1
$x_{11} = 1$
$x_{11} = 0$
$\varphi(x) = 0$
$\varphi(x) = 1$

$$|\psi_t\rangle = e^{iA_G t}|\psi_0\rangle$$

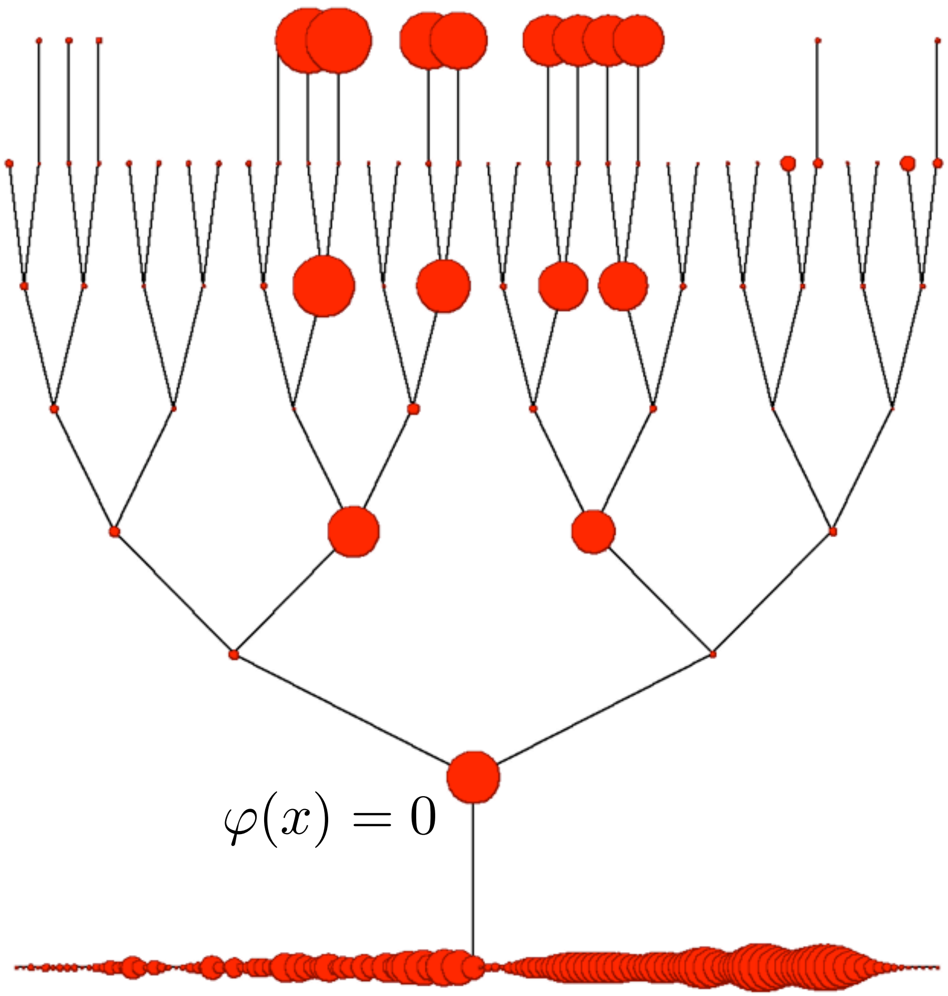




$$|\psi_t\rangle = e^{iA_G t}|\psi_0\rangle$$

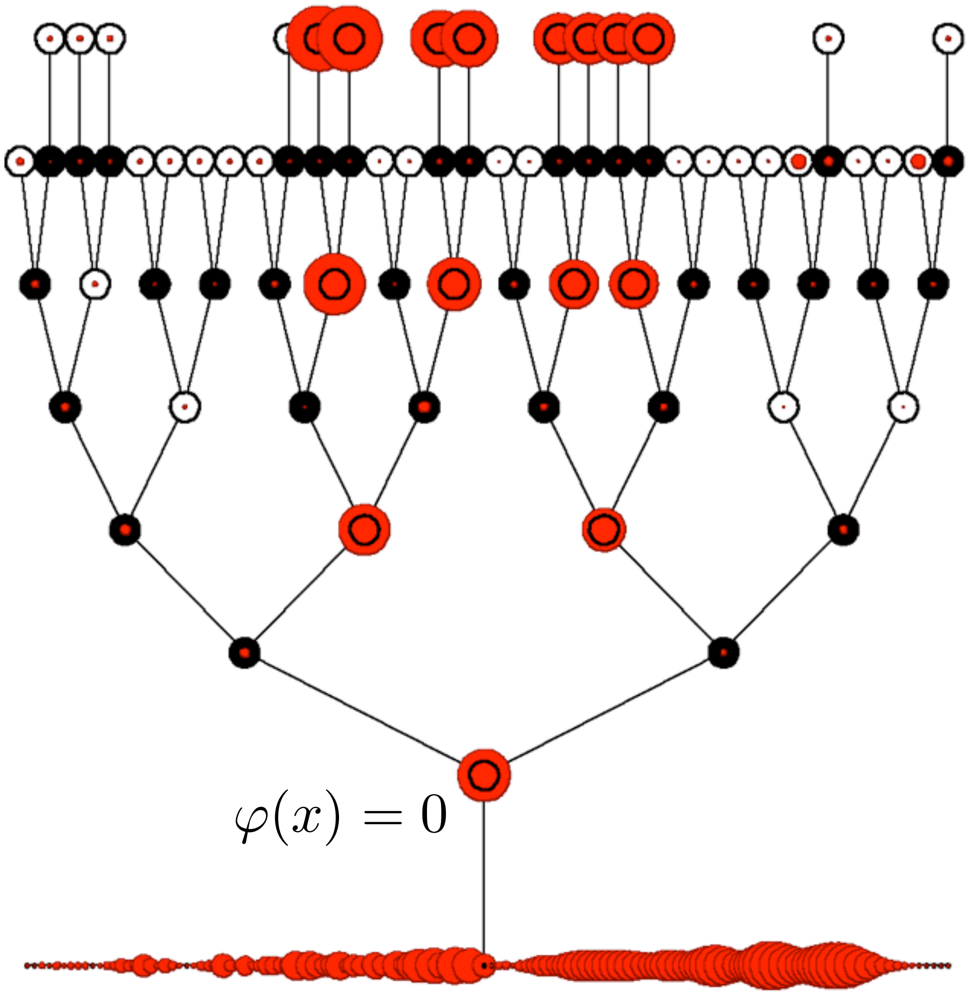


What's going on?



Observe: State inside tree converges to energy-zero eigenstate of the graph

What's going on?



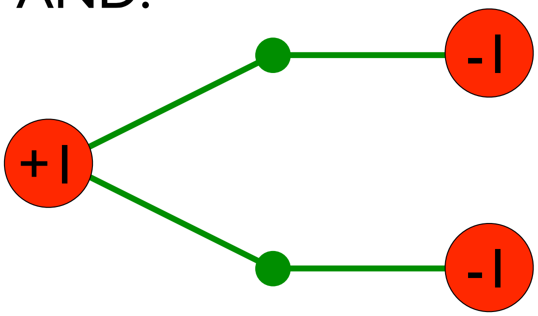
○ =0

● =1

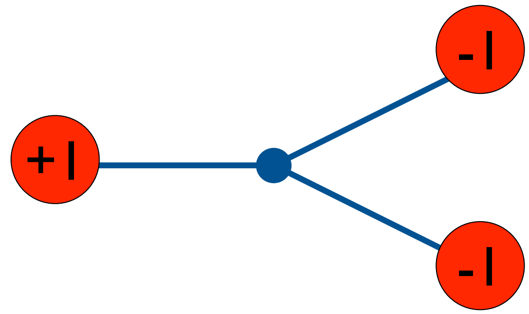
Observe: State inside tree converges to energy-zero eigenstate of the graph (supported on vertices that witness the formula's value)

Energy-zero eigenvectors for AND & OR gadgets

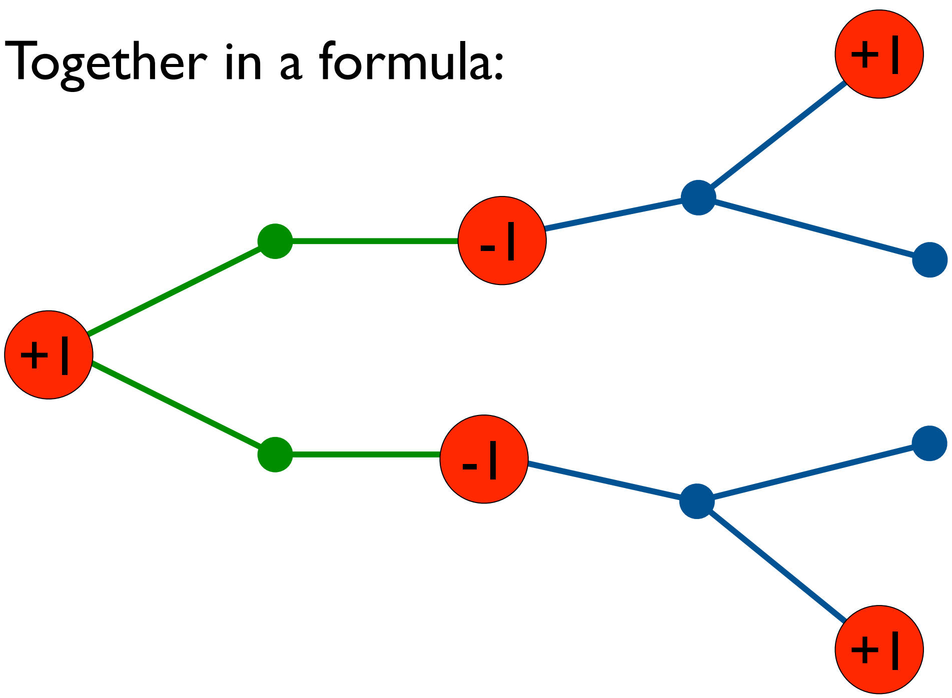
AND:



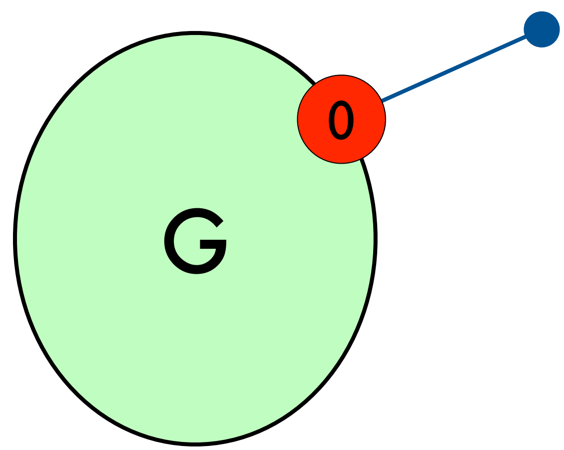
OR:



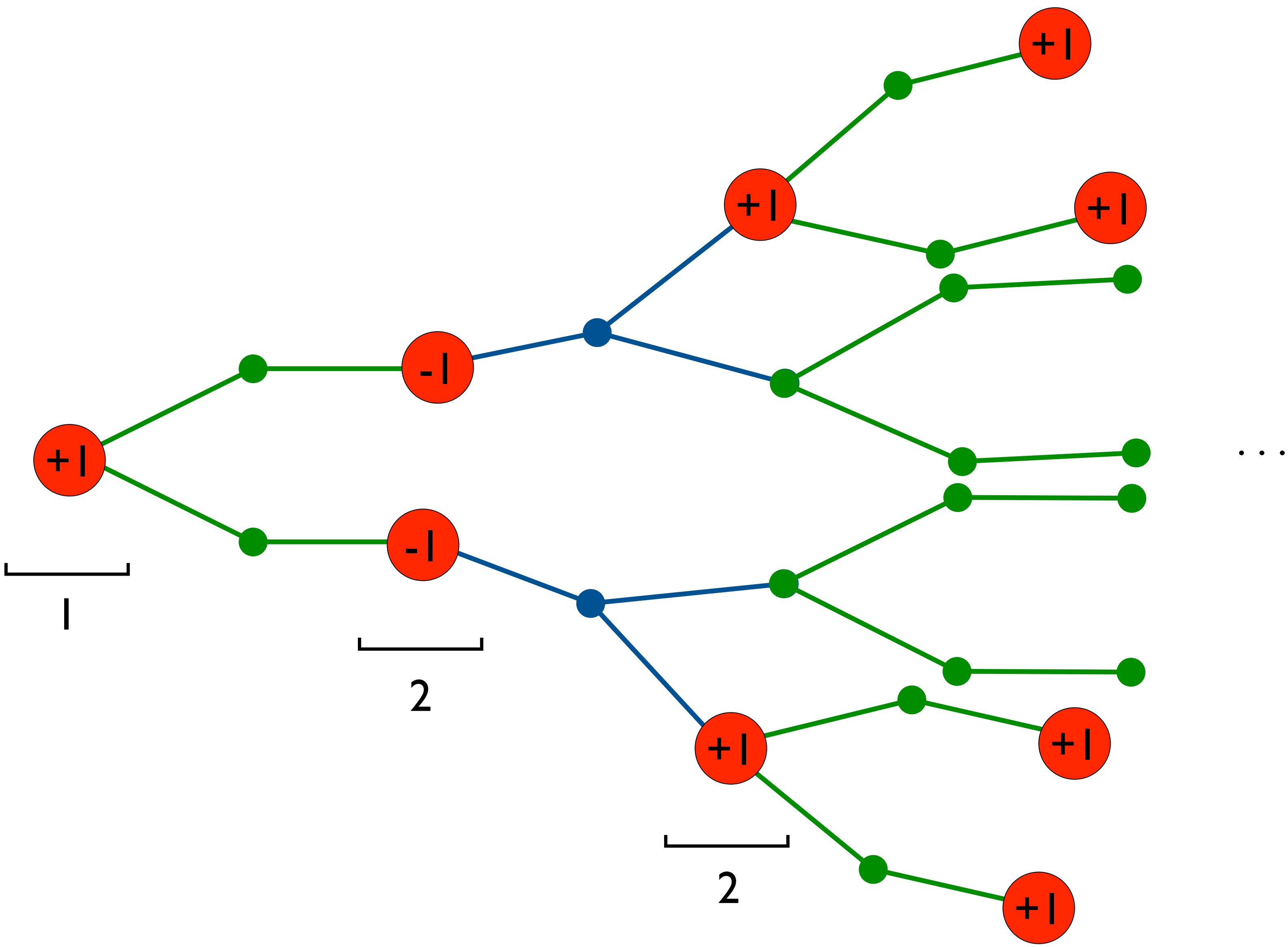
Together in a formula:



Input adds constraints
via dangling edges:



Balanced AND-OR formula evaluation in O(√n) time



Squared norm = $1+2+2+4+4+8+8+\cdots+2^{\frac{1}{2}\log_2 n} = O(\sqrt{n})$

Effective spectral gap lemma

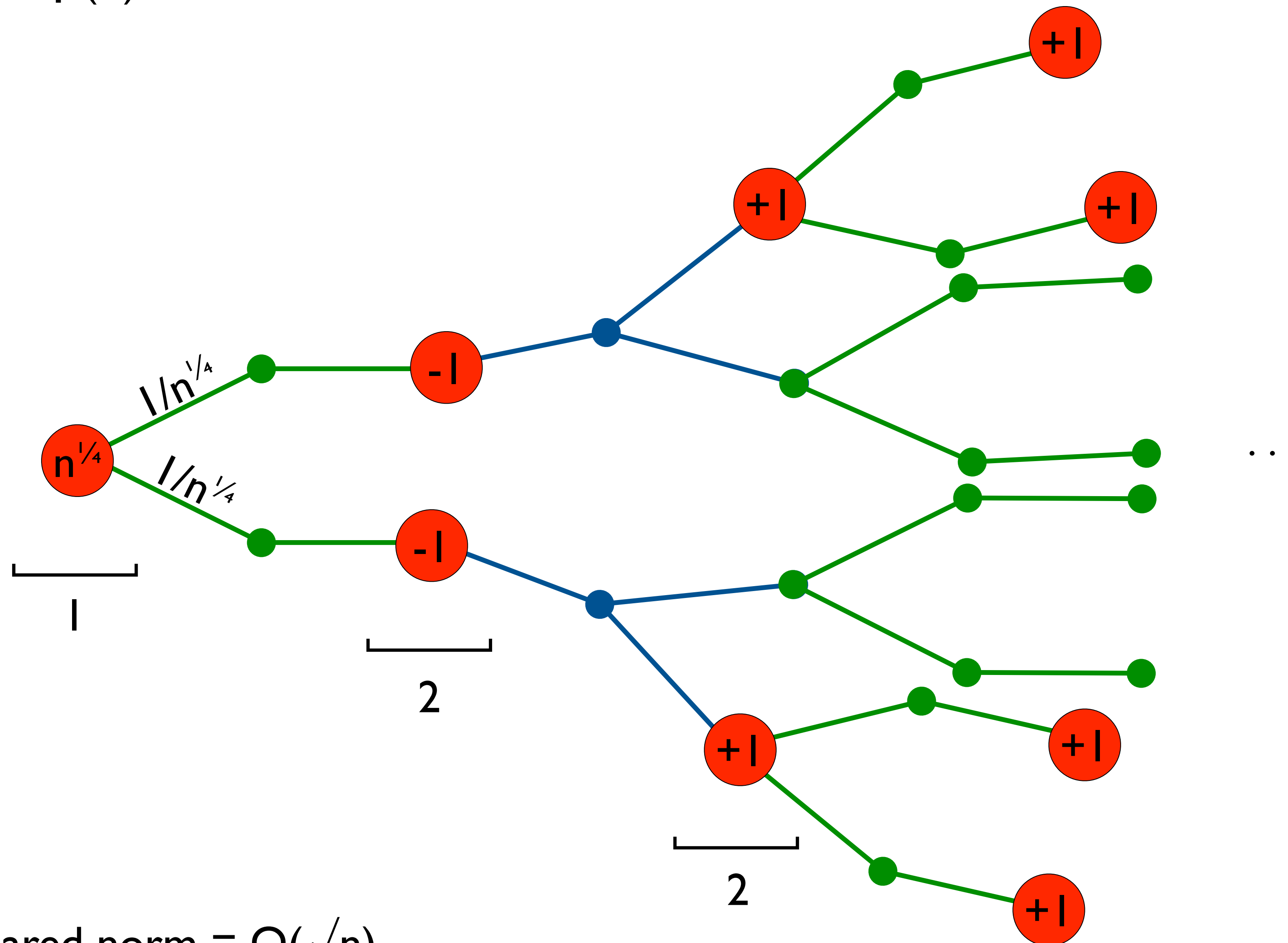
If M $\vec{u}$ ≠ 0, then M $\vec{u}$ ⊥ Kernel($M^\dagger$)

$$\left(\text{by the SVD } M = \sum_\rho \rho \, |v_\rho\rangle\langle u_\rho| \right)$$

$$\left\| \begin{array}{c} \text{projection of } M\,\vec{u} \text{ onto the} \\ \text{span of the left singular vectors} \\ \text{of } M \text{ with singular values } \leq \lambda \end{array} \right\| \leq \lambda \|\vec{u}\|$$

$$\left(\text{since } \|\Pi M \vec{u}\|^2 = \sum_{\rho \leq \lambda} \rho^2 |\langle u_\rho | u \rangle|^2 \right)$$

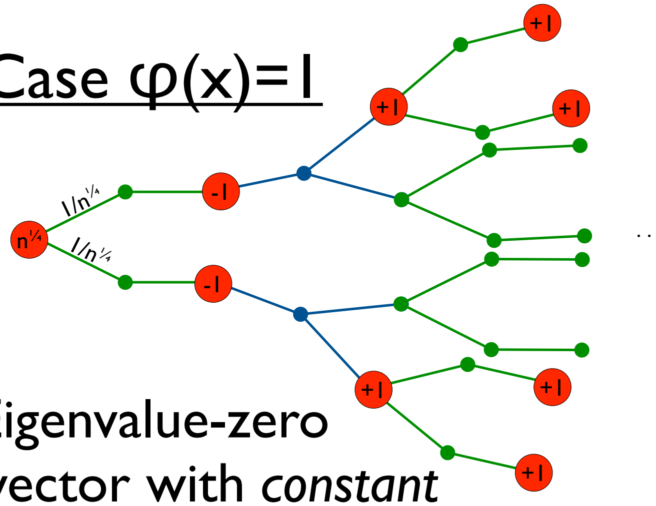
Case φ(x)=1



Squared norm = $O(\sqrt{n})$

Constant overlap on root vertex

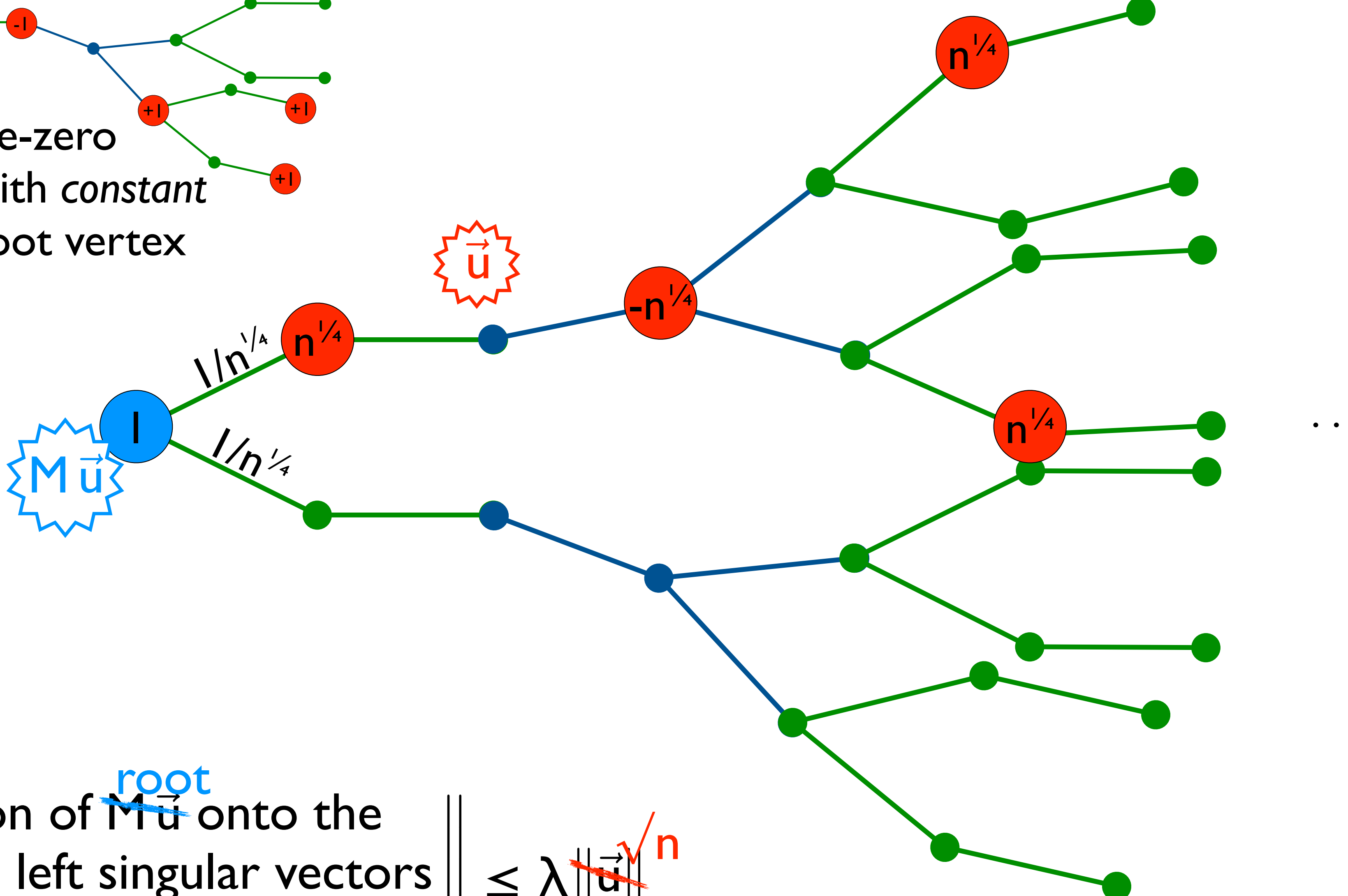
Case φ(x)=1



Eigenvalue-zero eigenvector with *constant* overlap on root vertex

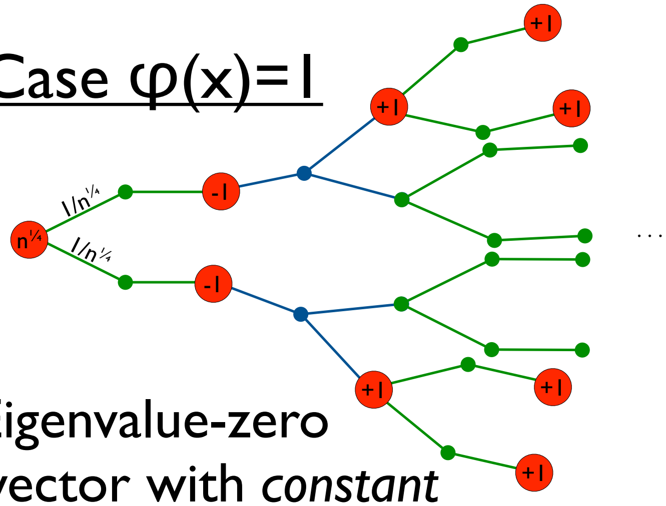
Case φ(x)=0

Root vertex has $\Omega(1/\sqrt{n})$ *effective spectral gap*



$$\left\| \begin{array}{c} \text{projection of } \cancel{M\vec{u}}\ \text{root onto the} \\ \text{span of the left singular vectors} \\ \text{of } M \text{ with singular values} \leq \lambda \end{array} \right\| \leq \lambda \cancel{\|\vec{u}\|}\sqrt{n}$$

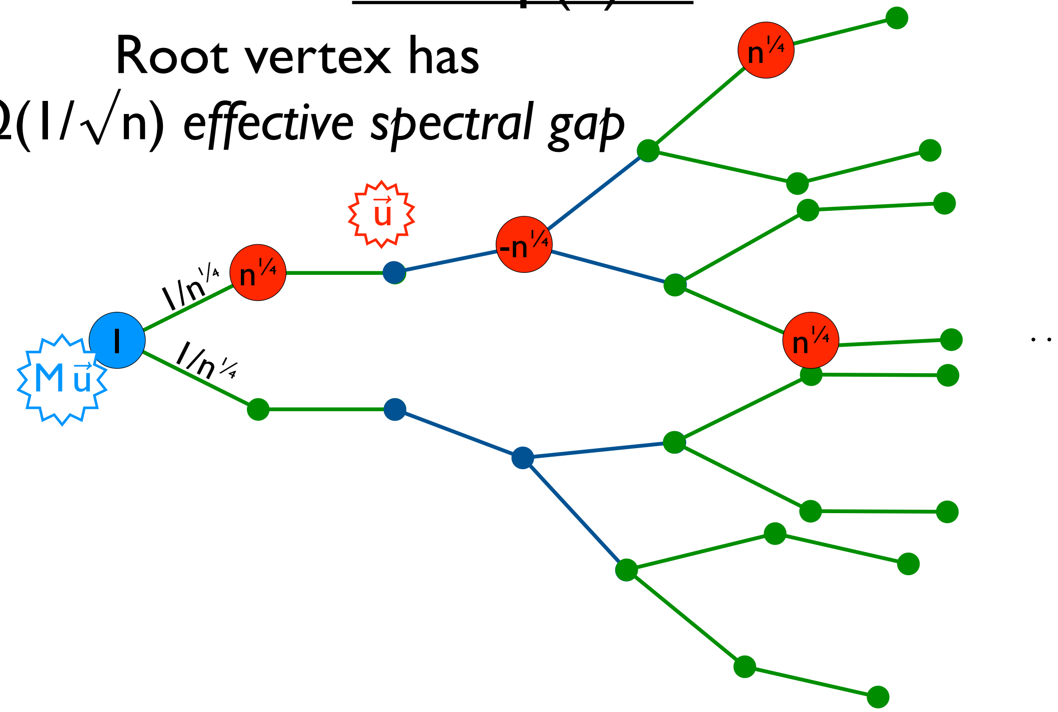
Case $\varphi(x)=1$



Eigenvalue-zero eigenvector with *constant* overlap on root vertex

Case $\varphi(x)=0$

Root vertex has $\Omega(1/\sqrt{n})$ *effective spectral gap*



Quantum algorithm:

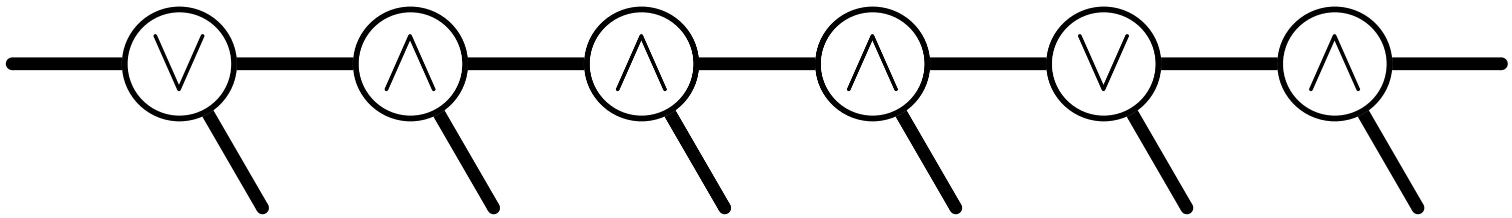
Run a quantum walk on the graph, for $\sqrt{n}$ steps from the root.

- $\varphi(x)=1 \Rightarrow$ walk is stationary
- $\varphi(x)=0 \Rightarrow$ walk mixes

Evaluating unbalanced formulas

[Ambainis, Childs, Reichardt, Špalek, Zhang '10]

Proper edge weights on an unbalanced formula give $\sqrt{(n \cdot \text{depth})}$ queries



depth n, spectral gap 1/n

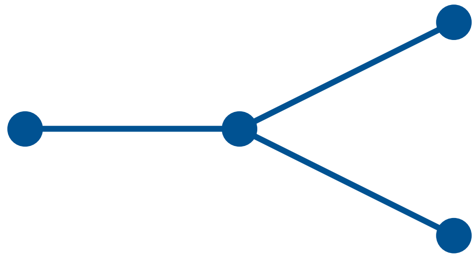
"Rebalancing" Theorem:
For any AND-OR formula with n leaves, there is an equivalent formula with $n\, e^{\sqrt{\log n}}$ leaves, and depth $e^{\sqrt{\log n}}$

[Bshouty, Cleve, Eberly '91, Bonet, Buss '94]

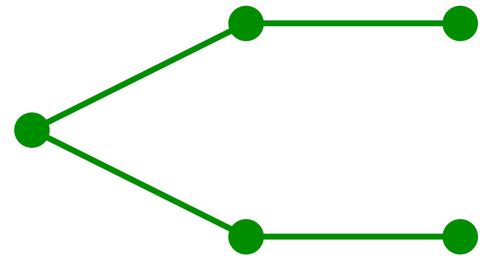
$\Rightarrow$ $O(\sqrt{n}\, e^{\sqrt{\log n}})$ query algorithm

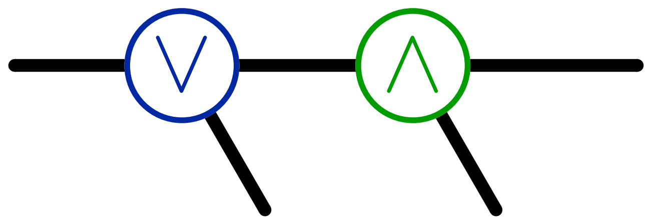
Today: $O(\sqrt{n} \log n)$

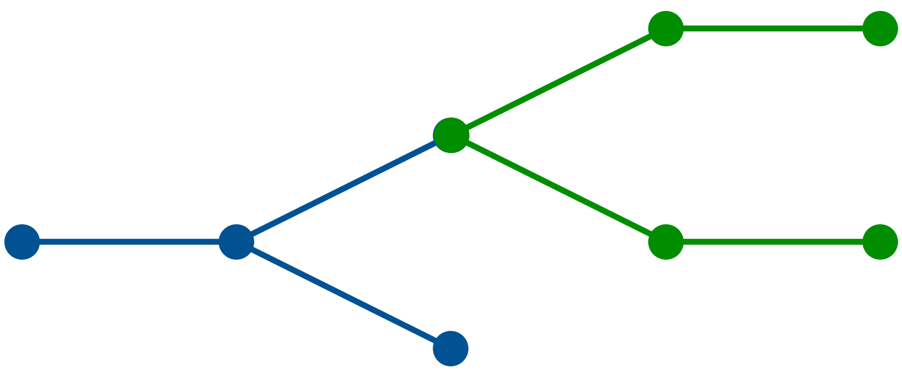
OR:

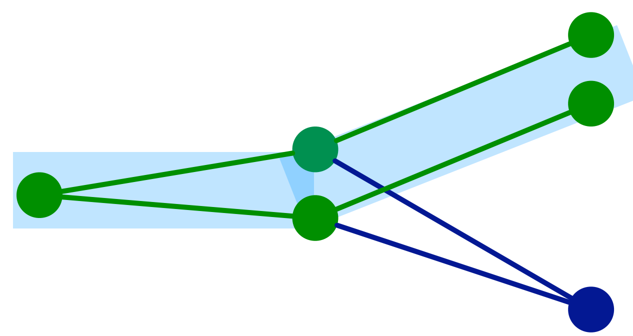


AND:







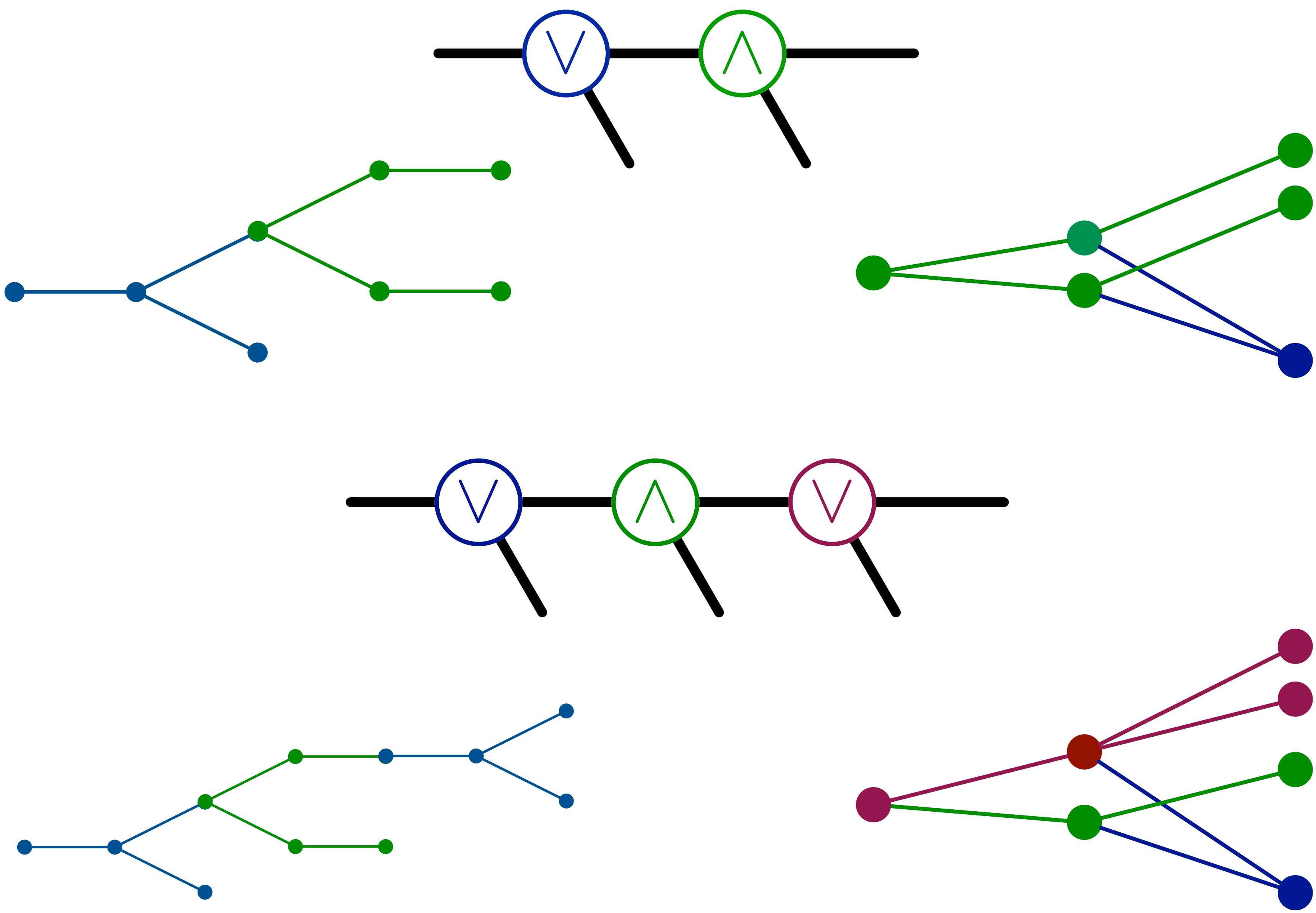


Direct-sum composition

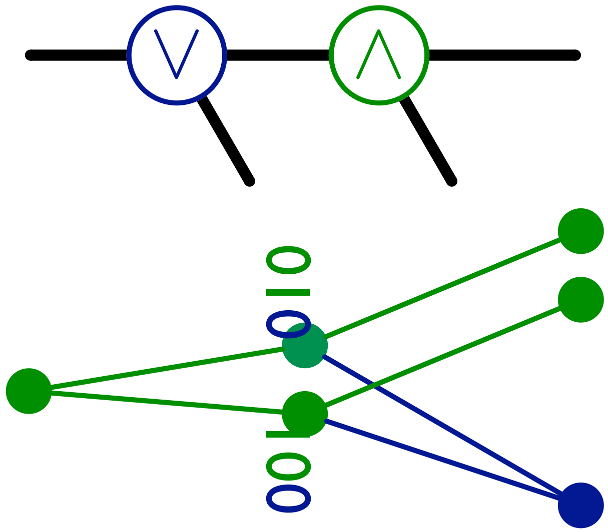
Tensor-product composition

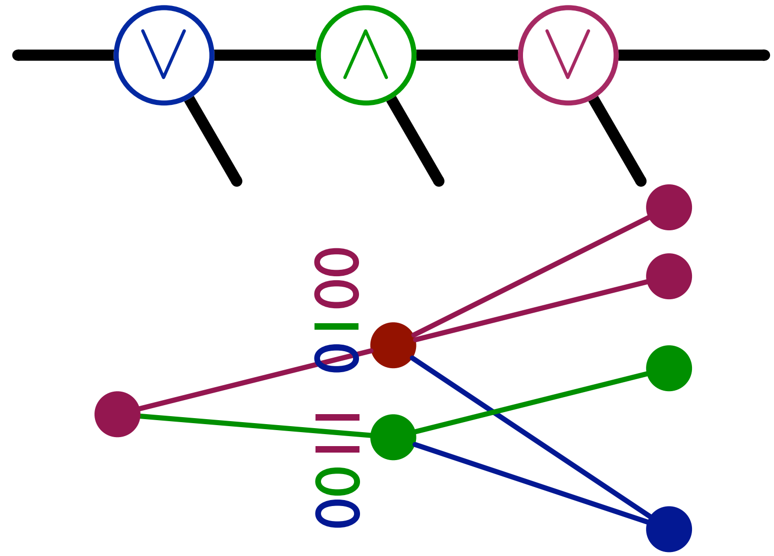
Direct-sum composition

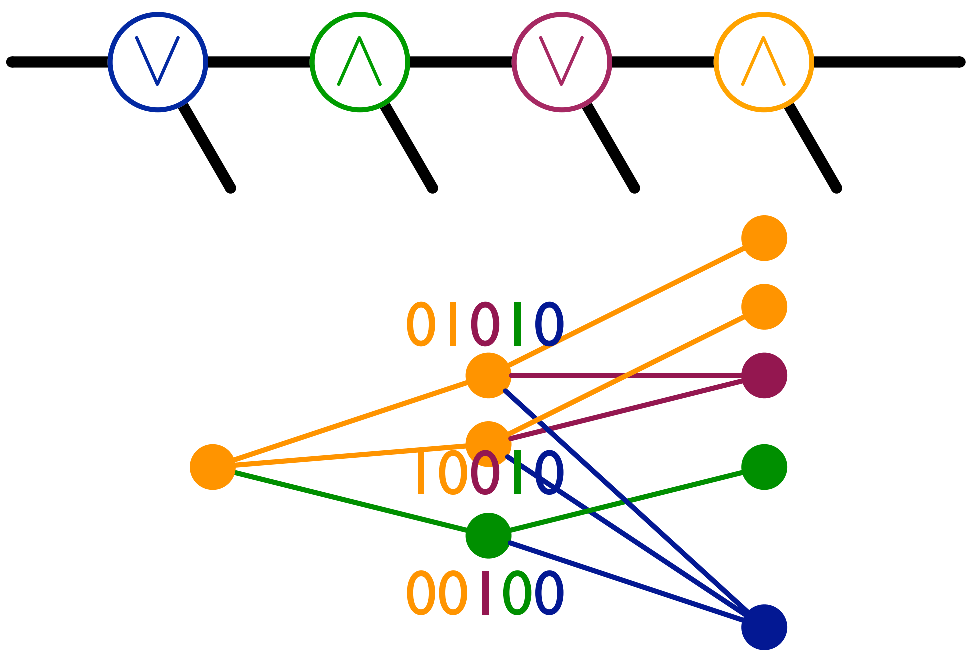
Tensor-product composition



Tensor-product composition





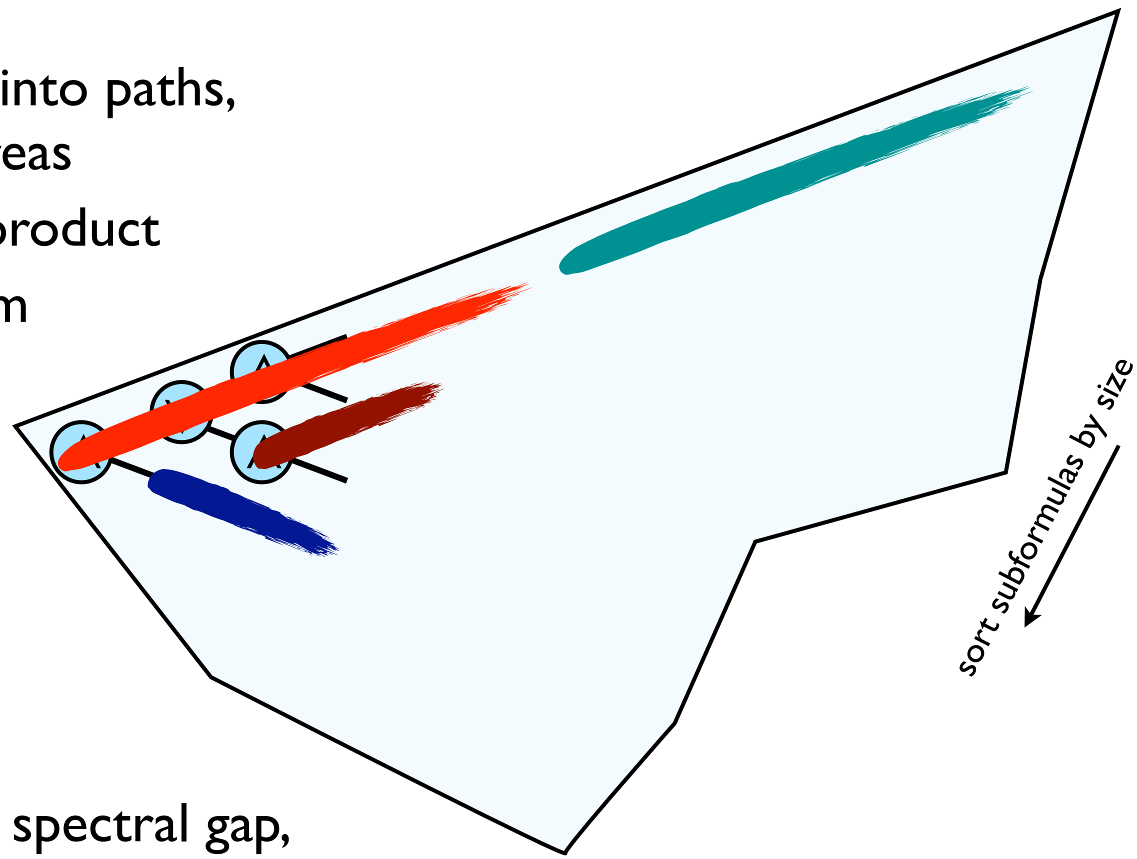


Properties

- Depth from root stays ≤2 — $1/\sqrt{n}$ spectral gap
- Graph stays sparse—provided composition is along the maximally unbalanced formula
- Middle vertices ↔ Maximal false inputs

Final algorithm

- With direct-sum composition, large depth implies small spectral gap
- Tensor-product composition gives $\sqrt{n}$-query algorithm (optimal), but graph is dense and norm too large for efficient implementation of quantum walk
- Hybrid approach:
  - Decompose the formula into paths, longer in less balanced areas
  - Along each path, tensor-product
  - Between paths, direct-sum



- Tradeoff gives $1/(\sqrt{n} \log n)$ spectral gap, while maintaining sparsity and small norm
  $\Rightarrow$ Quantum walk has efficient implementation (poly-log n after preprocessing)

ACRŠZ '10	today
$\sqrt{n}\, e^{\sqrt{\log n}}$	$\sqrt{n} \log n$