Classical Applications of Quantum Information Theory

Ronald de Wolf

Centrum Wiskunde & Informatica

Amsterdam

Unexpected proofs: Complex numbers

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

$\cos(x+y)$

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

$\cos(x+y) = \Re(e^{i(x+y)})$

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

$\cos(x+y) = \Re(e^{i(x+y)}) = \Re(e^{ix}e^{iy})$

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

$\cos(x+y) = \Re(e^{i(x+y)}) = \Re(e^{ix}e^{iy})$

$= \Re($

Unexpected proofs: Complex numbers

How to prove

$\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$?

Go to complex numbers!

$e^{ix} = \cos(x) + i\sin(x)$

$$\cos(x+y) = \Re(e^{i(x+y)}) = \Re(e^{ix}e^{iy})$$

$$= \Re(\cos(x)\cos(y) - \sin(x)\sin(y) + i\cos(x)\sin(y) + i\sin(x)\cos(y))$$

Unexpected proofs: Probabilities

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random
2. set $X_{ij} = 1$ if edge (i, j) crosses T (either $i \in T$ or $j \in T$)

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V,E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random
2. set $X_{ij} = 1$ if edge (i,j) crosses T (either $i \in T$ or $j \in T$)
3. Exp $\left[\sum_{(i,j)\in E} X_{ij}\right] = \sum_{(i,j)\in E}$ Exp$[X_{ij}]$

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random
2. set $X_{ij} = 1$ if edge (i, j) crosses T (either $i \in T$ or $j \in T$)
3. $\text{Exp}\left[\sum_{(i,j)\in E} X_{ij}\right] = \sum_{(i,j)\in E} \underbrace{\text{Exp}[X_{ij}]}_{=1/2}$

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random
2. set $X_{ij} = 1$ if edge (i, j) crosses T (either $i \in T$ or $j \in T$)
3. $\text{Exp}\left[\sum_{(i,j)\in E} X_{ij}\right] = \sum_{(i,j)\in E} \underbrace{\text{Exp}[X_{ij}]}_{=1/2} = m/2$

Unexpected proofs: Probabilities

Probabilistic method (Erdős, Alon & Spencer)

Theorem: Every graph (V, E) with m edges contains a bipartite subgraph with $m/2$ edges

Proof:

1. pick vertex-set $T \subseteq V$ at random
2. set $X_{ij} = 1$ if edge (i, j) crosses T (either $i \in T$ or $j \in T$)
3. $$\text{Exp}\left[\sum_{(i,j)\in E} X_{ij}\right] = \sum_{(i,j)\in E} \underbrace{\text{Exp}[X_{ij}]}_{=1/2} = m/2$$
4. but then there is a T with at least $m/2$ crossing edges!

Unexpected proofs: Quantum

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.
- Bonus: no need to implement anything in the lab :-)

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.
- Bonus: no need to implement anything in the lab :-)
- We'll focus on two sets of examples:

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.
- Bonus: no need to implement anything in the lab :-)
- We'll focus on two sets of examples:

  1. Using quantum information theory

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.
- Bonus: no need to implement anything in the lab :-)
- We'll focus on two sets of examples:
  1. Using quantum information theory
  2. Using the connections between quantum algorithms and polynomials

Unexpected proofs: Quantum

- We all know and love quantum information & computation for its algorithms, crypto-schemes, weird communication protocols, non-local effects, etc.
- This talk: using quantum techniques as a proof tool for things in *classical* CS, mathematics, etc.
- Bonus: no need to implement anything in the lab :-)

- We'll focus on two sets of examples:
  1. Using quantum information theory
  2. Using the connections between quantum algorithms and polynomials
- Based on forthcoming survey with Andy Drucker

Part 1:

Using quantum information theory

Example: Locally decodable codes (KdW03)

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$
  decoding: $D(w) = x$ if w is “close” to $C(x)$

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$
  decoding: $D(w) = x$ if w is "close" to $C(x)$
- Inefficient if you only want to decode a small part of x

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$

  decoding: $D(w) = x$ if w is “close” to $C(x)$

- Inefficient if you only want to decode a small part of x
- C is k-query locally decodable if there is a decoder D that only looks at k bits of w, and $D(w, i) = x_i$ (w.h.p.)

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$

  decoding: $D(w) = x$ if w is “close” to $C(x)$

- Inefficient if you only want to decode a small part of x
- C is k-query locally decodable if there is a decoder D that only looks at k bits of w, and $D(w,i) = x_i$ (w.h.p.)

- Hard question: optimal tradeoff between k and m?

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$

  decoding: $D(w) = x$ if w is “close” to $C(x)$

- Inefficient if you only want to decode a small part of x
- C is k-query locally decodable if there is a decoder D that only looks at k bits of w, and $D(w,i) = x_i$ (w.h.p.)

- Hard question: optimal tradeoff between k and m?

- Using quantum, we can show: $k = 2 \Rightarrow m = 2^{\Omega(n)}$

Example: Locally decodable codes (KdW03)

- Error-correcting code: $C : \{0,1\}^n \to \{0,1\}^m$, $m \geq n$
  decoding: $D(w) = x$ if w is "close" to $C(x)$
- Inefficient if you only want to decode a small part of x
- C is k-query locally decodable if there is a decoder D that only looks at k bits of w, and $D(w,i) = x_i$ (w.h.p.)
- Hard question: optimal tradeoff between k and m?
- Using quantum, we can show: $k = 2 \Rightarrow m = 2^{\Omega(n)}$
- Still the only superpolynomial bound known for LDCs

Exponential bound on 2-query LDC

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \rightarrow \{0,1\}^m$, 2-query classical decoder

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \rightarrow \{0,1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!

Exponential bound on 2-query LDC

- Given $C : \{0, 1\}^n \to \{0, 1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!
- Some massaging: make the quantum query uniform

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \to \{0,1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!
- Some massaging: make the quantum query uniform
- Consider query-result $|\phi_x\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} (-1)^{C(x)_j} |j\rangle$

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \to \{0,1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!
- Some massaging: make the quantum query uniform
- Consider query-result $|\phi_x\rangle = \frac{1}{\sqrt{m}}\sum_{j=1}^{m}(-1)^{C(x)_j}|j\rangle$
- $|\phi_x\rangle$ has $\log m$ qubits, but allows us to predict each of the encoded bits $x_1, \ldots, x_n$

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \to \{0,1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!
- Some massaging: make the quantum query uniform
- Consider query-result $|\phi_x\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} (-1)^{C(x)_j} |j\rangle$
- $|\phi_x\rangle$ has $\log m$ qubits, but allows us to predict each of the encoded bits $x_1, \ldots, x_n$
- Nayak's random access code bound: $\log m \geq \Omega(n)$

Exponential bound on 2-query LDC

- Given $C : \{0,1\}^n \to \{0,1\}^m$, 2-query classical decoder
- Can replace 2 classical queries by 1 quantum query!
- Some massaging: make the quantum query uniform
- Consider query-result $|\phi_x\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} (-1)^{C(x)_j} |j\rangle$
- $|\phi_x\rangle$ has $\log m$ qubits, but allows us to predict each of the encoded bits $x_1, \ldots, x_n$
- Nayak's random access code bound: $\log m \geq \Omega(n)$

  $\Rightarrow$ 2-query LDCs need exponential length

Other examples using q info theory

Other examples using q info theory

- Lower bound for communication complexity of inner product (CDNT'98)

Other examples using q info theory

- Lower bound for communication complexity of inner product (CDNT'98)
  - This uses Holevo's theorem

Other examples using q info theory

- Lower bound for communication complexity of inner product (CDNT'98)
  - This uses Holevo's theorem
- Lower bounds on rigidity of Hadamard matrix (dW'06)

Other examples using q info theory

- Lower bound for communication complexity of inner product (CDNT'98)
  - This uses Holevo's theorem
- Lower bounds on rigidity of Hadamard matrix (dW'06)
  - This uses the fact (due to Nayak) that encoding of n objects in a d-dimensional quantum system has average recovery probability $\leq d/n$

Part 2:

Using connections with polynomials

Quantum query algorithms

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0, 1\}^n$

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i,b\rangle \mapsto |i, b \oplus x_i\rangle$$

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output
- Most known quantum algorithms function in this setting:

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output
- Most known quantum algorithms function in this setting:

  Deutsch-Jozsa, Simon, Shor, Grover, random walks

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output
- Most known quantum algorithms function in this setting:

  Deutsch-Jozsa, Simon, Shor, Grover, random walks

- Connection with polynomials (BBCMW 98):

Quantum query algorithms

- T-query quantum algorithm interleaves fixed unitaries with queries to its input $x \in \{0,1\}^n$

$$O : |i, b\rangle \mapsto |i, b \oplus x_i\rangle$$

- A final measurement determines output
- Most known quantum algorithms function in this setting:

  Deutsch-Jozsa, Simon, Shor, Grover, random walks

- Connection with polynomials (BBCMW 98):

  $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree $\leq 2T$

Quantum lower bounds by polynomials

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$\alpha|i,0\rangle + \beta|i,1\rangle$

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$$\alpha|i,0\rangle + \beta|i,1\rangle \mapsto (\alpha(1-x_i) + \beta x_i)|i,0\rangle + (\alpha x_i + \beta(1-x_i))|i,1\rangle$$

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$$\alpha|i,0\rangle + \beta|i,1\rangle \mapsto (\alpha(1-x_i)+\beta x_i)|i,0\rangle + (\alpha x_i + \beta(1-x_i))|i,1\rangle$$

  3. Fixed unitaries don't change degree

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$$\alpha|i,0\rangle + \beta|i,1\rangle \mapsto (\alpha(1-x_i) + \beta x_i)|i,0\rangle + (\alpha x_i + \beta(1-x_i))|i,1\rangle$$

  3. Fixed unitaries don't change degree
- If the algorithm computes $f : \{0,1\}^n \to \{0,1\}$, then $P(x) \approx f(x)$ for all $x \in \{0,1\}^n$

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$$\alpha|i,0\rangle + \beta|i,1\rangle \mapsto (\alpha(1-x_i)+\beta x_i)|i,0\rangle + (\alpha x_i + \beta(1-x_i))|i,1\rangle$$

  3. Fixed unitaries don't change degree
- If the algorithm computes $f : \{0,1\}^n \to \{0,1\}$, then $P(x) \approx f(x)$ for all $x \in \{0,1\}^n$
- *Lower bounds* on degrees of approximating polynomials give *lower bounds* on quantum query complexity

Quantum lower bounds by polynomials

- $\Pr[\text{algo outputs } 1]$ is polynomial $P(x)$ of degree$\leq 2T$
- Because amplitudes of final state have degree$\leq T$:
  1. At the start: amplitudes are constants (degree 0)
  2. Query increases degree by 1:

$$\alpha|i,0\rangle + \beta|i,1\rangle \mapsto (\alpha(1-x_i) + \beta x_i)|i,0\rangle + (\alpha x_i + \beta(1-x_i))|i,1\rangle$$

  3. Fixed unitaries don't change degree
- If the algorithm computes $f : \{0,1\}^n \to \{0,1\}$, then $P(x) \approx f(x)$ for all $x \in \{0,1\}^n$
- *Lower bounds* on degrees of approximating polynomials give *lower bounds* on quantum query complexity
- Instead of a lower bound method, we can also view this as a method for constructing polynomials!

Example: ε-approximations for symmetric f

Example: ε-approximations for symmetric f

- Sherstov (08) solved a problem in probability theory using the minimal degree of ε-approximating polynomials for symmetric Boolean functions

Example: ε-approximations for symmetric f

- Sherstov (08) solved a problem in probability theory using the minimal degree of ε-approximating polynomials for symmetric Boolean functions
- Symmetric $f : \{0,1\}^n \rightarrow \{0,1\}$ only depends on Hamming weight $|x|$.

Example: ε-approximations for symmetric f

- Sherstov (08) solved a problem in probability theory using the minimal degree of ε-approximating polynomials for symmetric Boolean functions
- Symmetric $f : \{0,1\}^n \to \{0,1\}$ only depends on Hamming weight $|x|$. Examples: OR, Parity, Majority

Example: ε-approximations for symmetric f

- Sherstov (08) solved a problem in probability theory using the minimal degree of ε-approximating polynomials for symmetric Boolean functions
- Symmetric $f : \{0,1\}^n \to \{0,1\}$ only depends on Hamming weight $|x|$. Examples: OR, Parity, Majority
- W.l.o.g.: Assume $f(x) = 1$ if x has weight $|x| \geq t$

Example: ε-approximations for symmetric f

- Sherstov (08) solved a problem in probability theory using the minimal degree of ε-approximating polynomials for symmetric Boolean functions
- Symmetric $f : \{0,1\}^n \to \{0,1\}$ only depends on Hamming weight $|x|$. Examples: OR, Parity, Majority
- W.l.o.g.: Assume $f(x) = 1$ if x has weight $|x| \geq t$
- Sherstov used Chebyshev polynomials to construct ε-error polynomials of degree

$$O\left(\sqrt{tn} + \sqrt{n \log(1/\varepsilon)}\right) \log n$$

Example: ε-approximations for symmetric f

Example: ε-approximations for symmetric f

- We can do better using quantum algorithms

Example: ε-approximations for symmetric f

- We can do better using quantum algorithms
- Simple proof for optimal degree bound (dW 08)

$$O\left(\sqrt{tn} + \sqrt{n\log(1/\varepsilon)}\right)$$

Example: ε-approximations for symmetric f

- We can do better using quantum algorithms
- Simple proof for optimal degree bound (dW 08)

$$O\left(\sqrt{tn} + \sqrt{n\log(1/\varepsilon)}\right)$$

- Ingredients:

Example: ε-approximations for symmetric f

- We can do better using quantum algorithms
- Simple proof for optimal degree bound (dW 08)

$$O\left(\sqrt{tn} + \sqrt{n\log(1/\varepsilon)}\right)$$

- Ingredients:
  - "exact Grover": if there are exactly i 1s, we can find one with certainty using $\frac{\pi}{4}\sqrt{n/i}$ queries

Example: ε-approximations for symmetric f

- We can do better using quantum algorithms
- Simple proof for optimal degree bound (dW 08)

$$O\left(\sqrt{tn} + \sqrt{n\log(1/\varepsilon)}\right)$$

- Ingredients:
  - "exact Grover": if there are exactly i 1s, we can find one with certainty using $\frac{\pi}{4}\sqrt{n/i}$ queries
  - "ε-error Grover": we can find one with error ε using $O(\sqrt{n\log(1/\varepsilon)})$ queries (BCWZ 99)

ε-approximations for symmetric f (cntd)

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \rightarrow \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for $|x| = t-1, t-2, \ldots, 3, 2, 1$

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:

1. Run exact Grover $t-1$ times, for
   $|x| = t-1, t-2, \ldots, 3, 2, 1$
   Note: if $|x| < t$, then this finds all 1s with certainty

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for
     $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:

1. Run exact Grover $t-1$ times, for
   $|x| = t-1, t-2, \ldots, 3, 2, 1$
   Note: if $|x| < t$, then this finds all 1s with certainty
   Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
2. Run ε-error Grover to try to find another 1

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for
     $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
  2. Run ε-error Grover to try to find another 1
     Queries: $O(\sqrt{n \log(1/\varepsilon)})$

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
  2. Run ε-error Grover to try to find another 1
     Queries: $O(\sqrt{n \log(1/\varepsilon)})$
  3. If step 2 found a 1, conclude $|x| \geq t$ and output 1;

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
  2. Run ε-error Grover to try to find another 1
     Queries: $O(\sqrt{n \log(1/\varepsilon)})$
  3. If step 2 found a 1, conclude $|x| \geq t$ and output 1; else assume all 1s have been found and output $f(x)$

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \rightarrow \{0,1\}$, error $\leq \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
  2. Run ε-error Grover to try to find another 1
     Queries: $O(\sqrt{n \log(1/\varepsilon)})$
  3. If step 2 found a 1, conclude $|x| \geq t$ and output 1; else assume all 1s have been found and output $f(x)$
- ε-error algorithm using $O(\sqrt{tn} + \sqrt{n \log(1/\varepsilon)})$ queries

ε-approximations for symmetric f (cntd)

- Goal: compute symmetric $f : \{0,1\}^n \to \{0,1\}$, error $\le \varepsilon$
- Quantum algorithm:
  1. Run exact Grover $t-1$ times, for
     $|x| = t-1, t-2, \ldots, 3, 2, 1$
     Note: if $|x| < t$, then this finds all 1s with certainty
     Queries: $\sum_{i=1}^{t-1} \frac{\pi}{4}\sqrt{n/i} = O(\sqrt{tn})$
  2. Run ε-error Grover to try to find another 1
     Queries: $O(\sqrt{n \log(1/\varepsilon)})$
  3. If step 2 found a 1, conclude $|x| \ge t$ and output 1;
     else assume all 1s have been found and output $f(x)$
- ε-error algorithm using $O(\sqrt{tn} + \sqrt{n \log(1/\varepsilon)})$ queries
  $\Rightarrow$ ε-error polynomial of degree $O(\sqrt{tn} + \sqrt{n \log(1/\varepsilon)})$

Connections polynomials $\longleftrightarrow$ q algos

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

- Approximate polynomials $\leftrightarrow$ bounded-error q algos

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

- Approximate polynomials $\leftrightarrow$ bounded-error q algos
- Sign-representing polynmls $\leftrightarrow$ unbounded-error q algos

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

- Approximate polynomials $\leftrightarrow$ bounded-error q algos
- Sign-representing polynmls $\leftrightarrow$ unbounded-error q algos
- Robust polynomials $\leftrightarrow$ robust quantum algorithms

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, . . .

- Approximate polynomials $\leftrightarrow$ bounded-error q algos
- Sign-representing polynmls $\leftrightarrow$ unbounded-error q algos
- Robust polynomials $\leftrightarrow$ robust quantum algorithms
- Rational polynomials $\leftrightarrow$ q algos with postselection

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

- Approximate polynomials $\leftrightarrow$ bounded-error q algos
- Sign-representing polynmls $\leftrightarrow$ unbounded-error q algos
- Robust polynomials $\leftrightarrow$ robust quantum algorithms
- Rational polynomials $\leftrightarrow$ q algos with postselection

  Pair of polynomials p, q such that $\dfrac{p(x)}{q(x)} \approx f(x)$ for all x

Connections polynomials $\leftrightarrow$ q algos

- Polynomials of different types are prominent in complexity theory, communication, learning theory, ...

- Approximate polynomials $\leftrightarrow$ bounded-error q algos
- Sign-representing polynmls $\leftrightarrow$ unbounded-error q algos
- Robust polynomials $\leftrightarrow$ robust quantum algorithms
- Rational polynomials $\leftrightarrow$ q algos with postselection

  Pair of polynomials p, q such that $\dfrac{p(x)}{q(x)} \approx f(x)$ for all x

  Related to Aaronson's PostBQP $=$ PP

Applications of these connections

Applications of these connections

- PP is closed under intersection (Aaronson'04)

Applications of these connections

- PP is closed under intersection (Aaronson'04)
- Tight upper bounds on sign-degree of read-once formulas (ACRSZ'07, Lee'09)

Applications of these connections

- PP is closed under intersection (Aaronson'04)
- Tight upper bounds on sign-degree of read-once formulas (ACRSZ'07, Lee'09)
- The only way we know how to construct robust polynomials for functions such as Parity (BNRW'05)

Applications of these connections

- PP is closed under intersection (Aaronson'04)
- Tight upper bounds on sign-degree of read-once formulas (ACRSZ'07, Lee'09)
- The only way we know how to construct robust polynomials for functions such as Parity (BNRW'05)
- Jackson's Theorem in approximation theory (Drucker& dW'09)

Applications of these connections

- PP is closed under intersection (Aaronson'04)
- Tight upper bounds on sign-degree of read-once formulas (ACRSZ'07, Lee'09)
- The only way we know how to construct robust polynomials for functions such as Parity (BNRW'05)
- Jackson's Theorem in approximation theory (Drucker& dW'09)
- Separating communication complexity classes PP and UPP (BVW'07)

Applications of these connections

- PP is closed under intersection (Aaronson'04)
- Tight upper bounds on sign-degree of read-once formulas (ACRSZ'07, Lee'09)
- The only way we know how to construct robust polynomials for functions such as Parity (BNRW'05)
- Jackson's Theorem in approximation theory (Drucker& dW'09)
- Separating communication complexity classes PP and UPP (BVW'07), using Razborov's conversion from quantum communication protocols to polynomials

Summary

Summary

- Quantum proofs for classical theorems

Summary

- Quantum proofs for classical theorems
- Two sets of examples:

Summary

- Quantum proofs for classical theorems
- Two sets of examples:

1. using quantum information theory

Summary

- Quantum proofs for classical theorems
- Two sets of examples:
  1. using quantum information theory
  2. using connections with polynomials

Summary

- Quantum proofs for classical theorems
- Two sets of examples:
  1. using quantum information theory
  2. using connections with polynomials
- There are other examples

Summary

- Quantum proofs for classical theorems
- Two sets of examples:
  1. using quantum information theory
  2. using connections with polynomials
- There are other examples (see our survey)

Summary

- Quantum proofs for classical theorems
- Two sets of examples:
  1. using quantum information theory
  2. using connections with polynomials
- There are other examples (see our survey)
- Not yet the probabilistic method on steroids, but

Summary

- Quantum proofs for classical theorems
- Two sets of examples:
    1. using quantum information theory
    2. using connections with polynomials
- There are other examples (see our survey)
- Not yet the probabilistic method on steroids, but
  this could be the beginning of a beautiful proof method