Tight Quantum Lower Bound for Approximate Counting with Quantum States

Alexander Belov
University of Latvia

joint with

Ansis Rosmanis
Nagoya University

Joint Latvian-Estonian Theory Days 2022

Rīga, May the 6th, 2022

LATVIJAS UNIVERSITĀTE ANNO 1919
NACIONĀLAIS ATTĪSTĪBAS PLĀNS 2020
EIROPAS SAVIENĪBA
Eiropas Reģionālās attīstības fonds
IEGULDĪJUMS TAVĀ NĀKOTNĒ

History

Variations

variations.pdf

Variations on Quantum Adversary

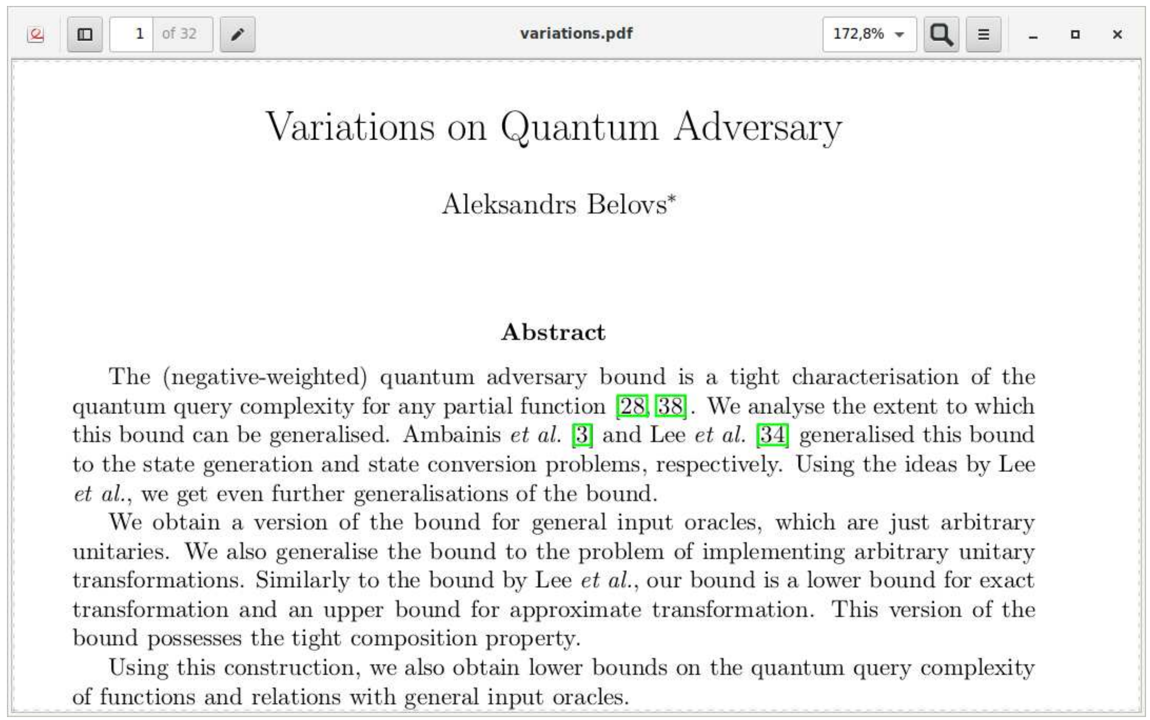
Aleksandrs Belovs*

Abstract

The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.



Variations on Quantum Adversary

Aleksandrs Belovs*

Abstract

The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.

variations.pdf

Variations on Quantum Adversary

Aleksandrs Belovs*

Abstract

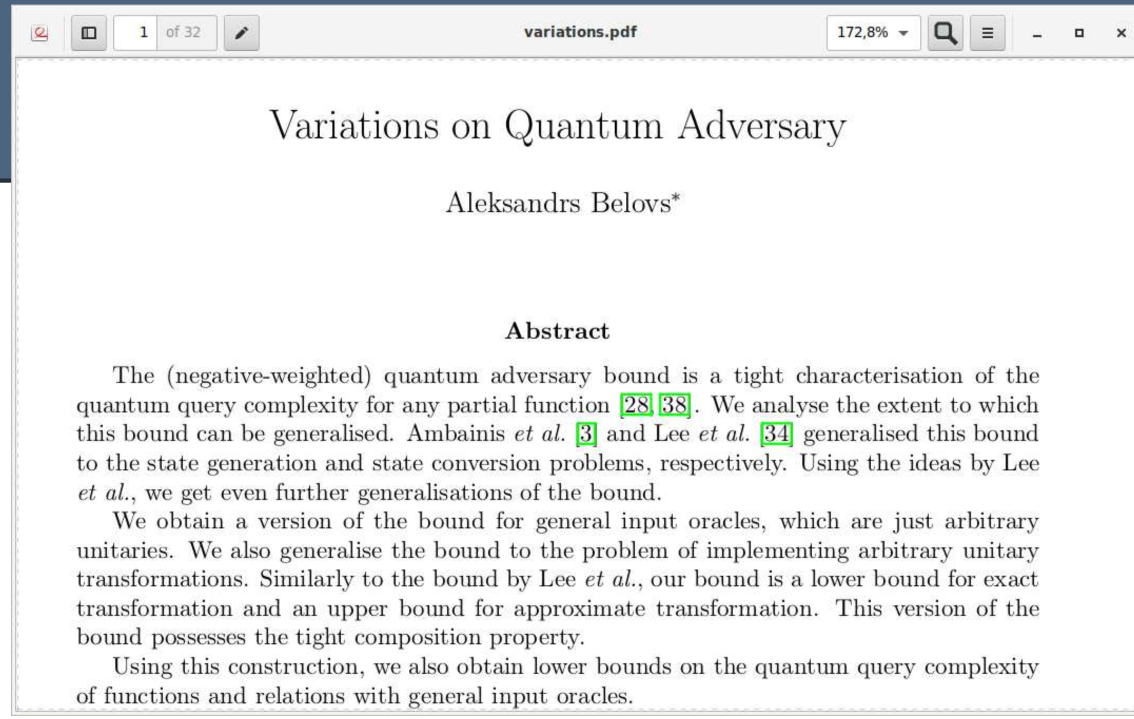
The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.

- **Goal:**
  Bring the quantum adversary method under common general settings.

tions



Variations on Quantum Adversary

Aleksandrs Belovs*

Abstract

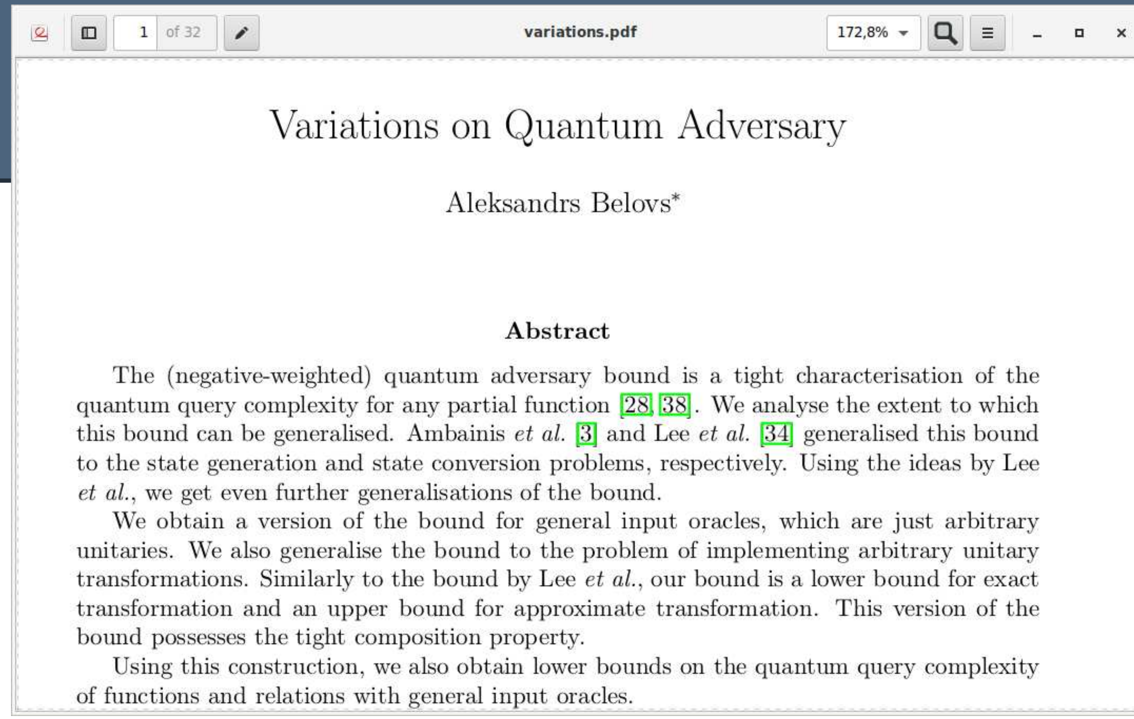
The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.

- **Usual Adversary** Tight characterization of quantum query complexity in normal settings:
  - Input: encodes a string x as a unitary $O_x : |i\rangle|0\rangle \mapsto |i\rangle|x_i\rangle$;
  - Output: evaluates a function $f(x)$ when measured.

...tions



Variations on Quantum Adversary

Aleksandrs Belovs*

Abstract

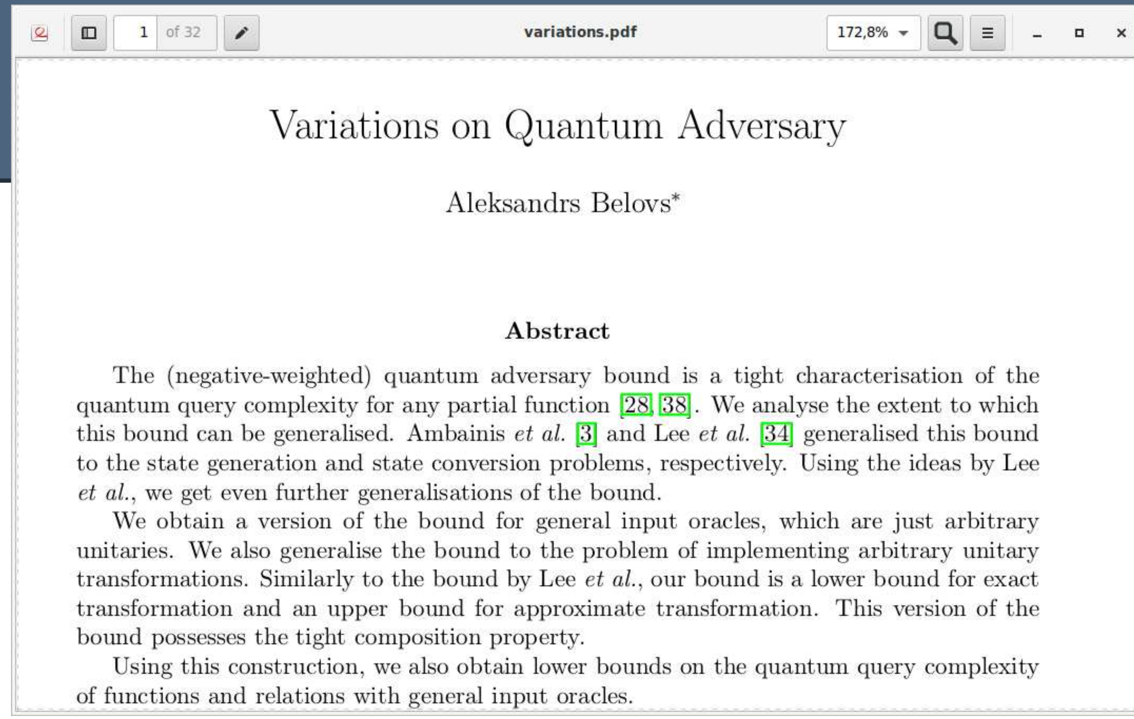
The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.

- **Usual Adversary** Tight characterization of quantum query complexity in normal settings:
  - Input: encodes a string x as a unitary $O_x : |i\rangle|0\rangle \mapsto |i\rangle|x_i\rangle$;
  - Output: evaluates a function $f(x)$ when measured.
- **“Variations” Adversary** Semi-tight characterization when
  - Input: arbitrary unitary O_x;
  - Output: arbitrary unitary V_x.

tions



Variations on Quantum Adversary

Aleksandrs Belovs*

Abstract

The (negative-weighted) quantum adversary bound is a tight characterisation of the quantum query complexity for any partial function [28, 38]. We analyse the extent to which this bound can be generalised. Ambainis *et al.* [3] and Lee *et al.* [34] generalised this bound to the state generation and state conversion problems, respectively. Using the ideas by Lee *et al.*, we get even further generalisations of the bound.

We obtain a version of the bound for general input oracles, which are just arbitrary unitaries. We also generalise the bound to the problem of implementing arbitrary unitary transformations. Similarly to the bound by Lee *et al.*, our bound is a lower bound for exact transformation and an upper bound for approximate transformation. This version of the bound possesses the tight composition property.

Using this construction, we also obtain lower bounds on the quantum query complexity of functions and relations with general input oracles.

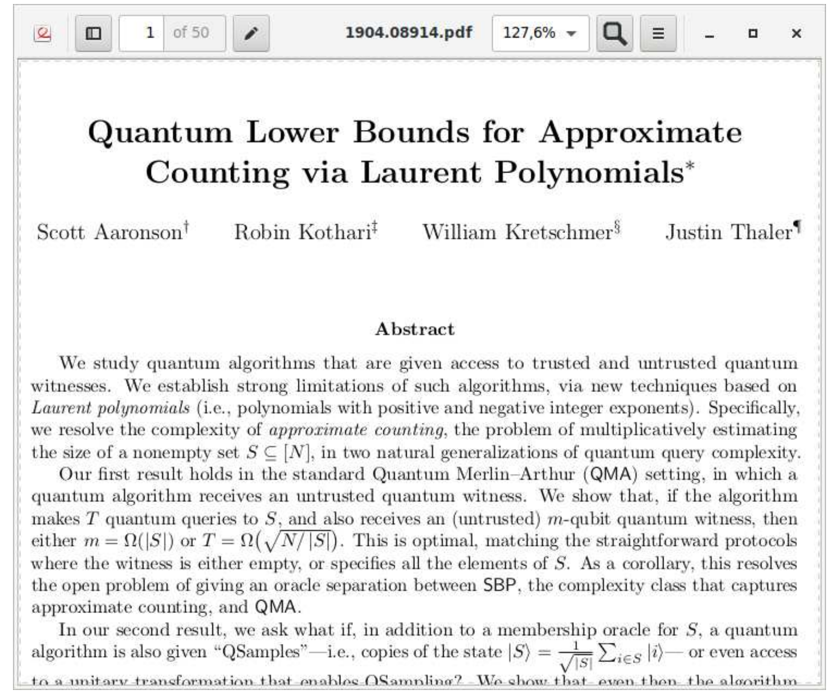
- **"Variations" Adversary** Semi-tight characterization when
  - ☐ Input: arbitrary unitary O_x;
  - ☐ Output: arbitrary unitary V_x.

From this, you can derive other versions:

- State conversion, state preparation, relation evaluation, . . .

Can this new method be used for real problems?

Suggestion



Quantum Lower Bounds for Approximate Counting via Laurent Polynomials*

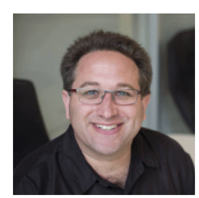
Scott Aaronson† Robin Kothari‡ William Kretschmer§ Justin Thaler¶

Abstract

We study quantum algorithms that are given access to trusted and untrusted quantum witnesses. We establish strong limitations of such algorithms, via new techniques based on *Laurent polynomials* (i.e., polynomials with positive and negative integer exponents). Specifically, we resolve the complexity of *approximate counting*, the problem of multiplicatively estimating the size of a nonempty set $S \subseteq [N]$, in two natural generalizations of quantum query complexity.

Our first result holds in the standard Quantum Merlin–Arthur (QMA) setting, in which a quantum algorithm receives an untrusted quantum witness. We show that, if the algorithm makes T quantum queries to S, and also receives an (untrusted) m-qubit quantum witness, then either $m = \Omega(|S|)$ or $T = \Omega(\sqrt{N/|S|})$. This is optimal, matching the straightforward protocols where the witness is either empty, or specifies all the elements of S. As a corollary, this resolves the open problem of giving an oracle separation between SBP, the complexity class that captures approximate counting, and QMA.

In our second result, we ask what if, in addition to a membership oracle for S, a quantum algorithm is also given "QSamples"—i.e., copies of the state $|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle$— or even access to a unitary transformation that enables QSampling? We show that, even then, the algorithm



Aaronson et al.

1 of 50 1904.08914.pdf 127,6%

Quantum Lower Bounds for Approximate Counting via Laurent Polynomials*

Scott Aaronson† Robin Kothari‡ William Kretschmer§ Justin Thaler¶

Abstract

We study quantum algorithms that are given access to trusted and untrusted quantum witnesses. We establish strong limitations of such algorithms, via new techniques based on *Laurent polynomials* (i.e., polynomials with positive and negative integer exponents). Specifically, we resolve the complexity of *approximate counting*, the problem of multiplicatively estimating the size of a nonempty set $S \subseteq [N]$, in two natural generalizations of quantum query complexity.

Our first result holds in the standard Quantum Merlin–Arthur (QMA) setting, in which a quantum algorithm receives an untrusted quantum witness. We show that, if the algorithm makes T quantum queries to S, and also receives an (untrusted) m-qubit quantum witness, then either $m = \Omega(|S|)$ or $T = \Omega(\sqrt{N/|S|})$. This is optimal, matching the straightforward protocols where the witness is either empty, or specifies all the elements of S. As a corollary, this resolves the open problem of giving an oracle separation between SBP, the complexity class that captures approximate counting, and QMA.

In our second result, we ask what if, in addition to a membership oracle for S, a quantum algorithm is also given "QSamples"—i.e., copies of the state $|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle$— or even access to a unitary transformation that enables QSampling? We show that, even then, the algorithm




NB: Dangling modifier...

- Old and well-known problem:
  - Given a subset $x \subseteq [n]$,
  - find $|x|$ with multiplicative precision ε.

- Can be done in $O\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{|x|}}\right)$ queries assuming membership access to x:

  Brassard, Høyer, and Tapp (1998).

- And this is optimal

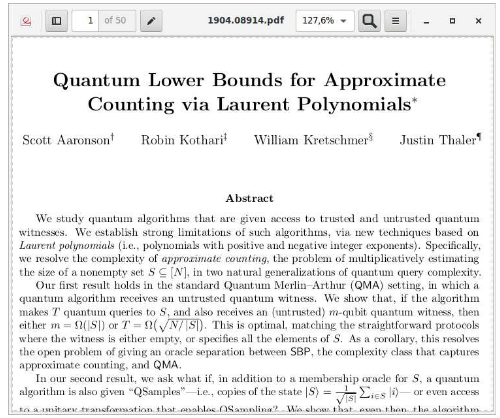
  Bennett, Bernstein, Brassard, and Vazirani (1997).

- Old and well-known problem:
  - Given a subset $x \subseteq [n]$,
  - ~~find $|x|$ with multiplicative precision ε~~.
  - distinguish two cases
    - $|x| = k$, or
    - $|x| = k' = (1+\varepsilon)k$.
- Can be done in $O\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{|x|}}\right)$ queries assuming membership access to x:

  Brassard, Høyer, and Tapp (1998).
- And this is optimal

  Bennett, Bernstein, Brassard, and Vazirani (1997).

New Life



1904.08914.pdf 1 of 50 127,6%

Quantum Lower Bounds for Approximate Counting via Laurent Polynomials*

Scott Aaronson† Robin Kothari‡ William Kretschmer§ Justin Thaler¶

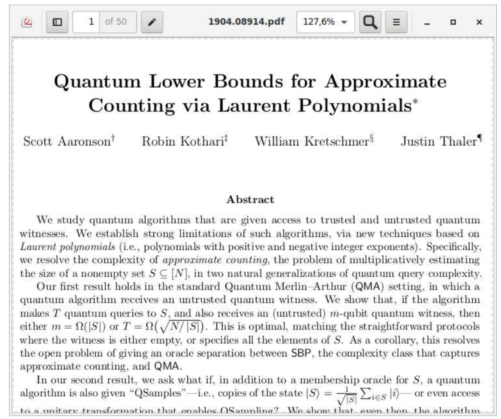
Abstract

We study quantum algorithms that are given access to trusted and untrusted quantum witnesses. We establish strong limitations of such algorithms, via new techniques based on *Laurent polynomials* (i.e., polynomials with positive and negative integer exponents). Specifically, we resolve the complexity of *approximate counting*, the problem of multiplicatively estimating the size of a nonempty set $S \subseteq [N]$, in two natural generalizations of quantum query complexity.

Our first result holds in the standard Quantum Merlin–Arthur (QMA) setting, in which a quantum algorithm receives an untrusted quantum witness. We show that, if the algorithm makes T quantum queries to S, and also receives an (untrusted) m-qubit quantum witness, then either $m = \Omega(|S|)$ or $T = \Omega\left(\sqrt{N/|S|}\right)$. This is optimal, matching the straightforward protocols where the witness is either empty, or specifies all the elements of S. As a corollary, this resolves the open problem of giving an oracle separation between SBP, the complexity class that captures approximate counting, and QMA.

In our second result, we ask what if, in addition to a membership oracle for S, a quantum algorithm is also given "QSamples"—i.e., copies of the state $|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle$— or even access

- Membership oracle is good, but too classique.
- Let us have more quantum resources.



Quantum Lower Bounds for Approximate Counting via Laurent Polynomials*

Scott Aaronson† Robin Kothari‡ William Kretschmer§ Justin Thaler¶

Abstract

We study quantum algorithms that are given access to trusted and untrusted quantum witnesses. We establish strong limitations of such algorithms, via new techniques based on *Laurent polynomials* (i.e., polynomials with positive and negative integer exponents). Specifically, we resolve the complexity of *approximate counting*, the problem of multiplicatively estimating the size of a nonempty set $S \subseteq [N]$, in two natural generalizations of quantum query complexity.

Our first result holds in the standard Quantum Merlin–Arthur (QMA) setting, in which a quantum algorithm receives an untrusted quantum witness. We show that, if the algorithm makes T quantum queries to S, and also receives an (untrusted) m-qubit quantum witness, then either $m = \Omega(|S|)$ or $T = \Omega\left(\sqrt{N/|S|}\right)$. This is optimal, matching the straightforward protocols where the witness is either empty, or specifies all the elements of S. As a corollary, this resolves the open problem of giving an oracle separation between SBP, the complexity class that captures approximate counting, and QMA.

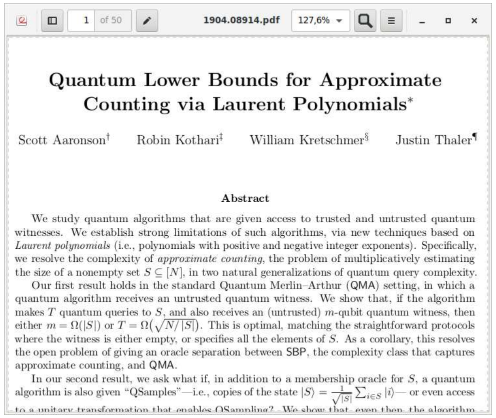
In our second result, we ask what if, in addition to a membership oracle for S, a quantum algorithm is also given "QSamples"—i.e., copies of the state $|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle$— or even access




- Membership oracle is good, but too classique.
- Let us have more quantum resources.

What can be more quantum than this

$$|\psi_x\rangle = \frac{1}{\sqrt{|x|}} \sum_{i \in x} |i\rangle \qquad ?$$



1 of 50 1904.08914.pdf 127,6%

Quantum Lower Bounds for Approximate Counting via Laurent Polynomials*

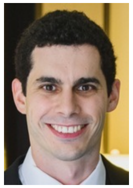
Scott Aaronson† Robin Kothari‡ William Kretschmer§ Justin Thaler¶

Abstract

We study quantum algorithms that are given access to trusted and untrusted quantum witnesses. We establish strong limitations of such algorithms, via new techniques based on *Laurent polynomials* (i.e., polynomials with positive and negative integer exponents). Specifically, we resolve the complexity of *approximate counting*, the problem of multiplicatively estimating the size of a nonempty set $S \subseteq [N]$, in two natural generalizations of quantum query complexity.

Our first result holds in the standard Quantum Merlin–Arthur (QMA) setting, in which a quantum algorithm receives an untrusted quantum witness. We show that, if the algorithm makes T quantum queries to S, and also receives an (untrusted) m-qubit quantum witness, then either $m = \Omega(|S|)$ or $T = \Omega\left(\sqrt{N/|S|}\right)$. This is optimal, matching the straightforward protocols where the witness is either empty, or specifies all the elements of S. As a corollary, this resolves the open problem of giving an oracle separation between SBP, the complexity class that captures approximate counting, and QMA.

In our second result, we ask what if, in addition to a membership oracle for S, a quantum algorithm is also given "QSamples"—i.e., copies of the state $|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle$— or even access



- Can have copies of
- Can reflect about

$$|\psi_x\rangle = \frac{1}{\sqrt{|x|}} \sum_{i \in x} |i\rangle .$$

Theorem. *Assume $\varepsilon = 1$. Then the algorithm either has to*

- *invoke the membership oracle* $\Omega\left(\sqrt{n/k}\right)$ *times, or*
- *access the state* $|\psi_x\rangle$ *at least* $\Omega\left(\min\{k^{1/3}, \sqrt{n/k}\}\right)$ *times.*

Our Result

Our Settings

Assume $\varepsilon = 1$...

We consider all $1/k \leq \varepsilon \leq 1$.

Our Settings

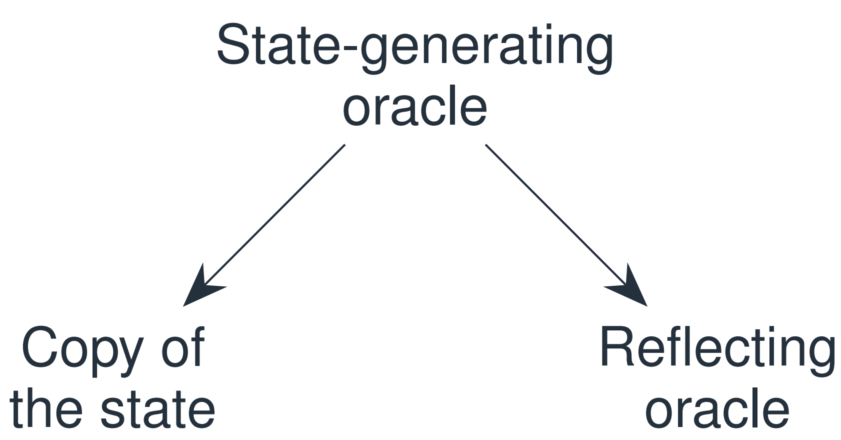
Assume $\varepsilon = 1$...

We consider all $1/k \leq \varepsilon \leq 1$.

- Can have copies of
- Can reflect about

$$|\psi_x\rangle = \frac{1}{\sqrt{|x|}} \sum_{i \in x} |i\rangle.$$

We add state-generating oracle: $|0\rangle \mapsto |\psi_x\rangle$.



Copies of the state	$\ell = \Omega\left(\min\left\{k,\ \frac{\sqrt{k}}{\varepsilon},\ \frac{n}{k\varepsilon^2}\right\}\right)$
Membership oracle	$q_{\mathsf{M}} = \Omega\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{k}}\right)$
State-generating oracle	$q_{\mathsf{G}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\quad \frac{k^{1/3}}{\varepsilon^{2/3}}\right\}\right)$
• with copies of the state	$q_{\mathsf{G}}\sqrt{\ell} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
Reflecting oracle	$q_{\mathsf{R}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \sqrt{\frac{k}{\varepsilon}} + \sqrt{\frac{n}{k}}\right\}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}}\sqrt{\ell + q_{\mathsf{G}}} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $\ell + q_{\mathsf{G}} \geq 1$
• with membership oracle:	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $q_{\mathsf{M}} = \Omega\left(\sqrt{\frac{n}{k}}\right)$

Copies of the state	$\ell = \Omega\left(\min\left\{k,\ \frac{\sqrt{k}}{\varepsilon},\ \frac{n}{k\varepsilon^2}\right\}\right)$
Membership oracle	$q_{\mathsf{M}} = \Omega\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{k}}\right)$
State-generating oracle	$q_{\mathsf{G}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \frac{k^{1/3}}{\varepsilon^{2/3}}\right\}\right)$
• with copies of the state	$\ldots\sqrt{\frac{k}{\varepsilon}}\right)$
Reflecting oracle	$\ldots\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \sqrt{\frac{k}{\varepsilon}} + \sqrt{\frac{n}{k}}\right\}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}}\sqrt{\ell} + q_{\mathsf{G}} \ldots \sqrt{\frac{k}{\varepsilon}}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $\ell + q_{\mathsf{G}} \geq 1$
• with membership oracle:	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $q_{\mathsf{M}} = \Omega\left(\sqrt{\frac{n}{k}}\right)$

This is tight!

Copies of the state	$\ell = \Omega\left(\min\left\{k,\ \frac{\sqrt{k}}{\varepsilon},\ \frac{n}{k\varepsilon^2}\right\}\right)$
Membership oracle	$q_{\mathsf{M}} = \Omega\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{k}}\right)$
State-generating oracle	$q_{\mathsf{G}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \frac{k^{1/3}}{\varepsilon^{2/3}}\right\}\right)$
• with copies of the state	$q_{\mathsf{G}}\sqrt{\ell} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
Reflecting oracle	$q_{\mathsf{R}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \sqrt{\frac{k}{\varepsilon}} + \sqrt{\frac{n}{k}}\right\}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}}\sqrt{\ell + q_{\mathsf{G}}} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $\ell + q_{\mathsf{G}} \geq 1$
• with membership oracle:	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $q_{\mathsf{M}} = \Omega\left(\sqrt{\frac{n}{k}}\right)$

Proposition. It is possible to solve the approximate counting problem using $O(k \log k)$ classical samples from x.

Proof. Sample the elements out of x sufficiently many times.
Output that $|x| = k$ if the number of distinct elements observed is at most k, otherwise output that $|x| = k'$.
The algorithm has 1-sided error.
The analysis follows from the standard coupon collecting problem.

- This is the only place where we are tight only up to a log.

Copies of the state	$\ell = \Omega\left(\min\left\{k,\ \frac{\sqrt{k}}{\varepsilon},\ \frac{n}{k\varepsilon^2}\right\}\right)$
Membership oracle	$q_{\mathsf{M}} = \Omega\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{k}}\right)$
State-generating oracle	$q_{\mathsf{G}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \frac{k^{1/3}}{\varepsilon^{2/3}}\right\}\right)$
• with copies of the state	$q_{\mathsf{G}}\sqrt{\ell} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
Reflecting oracle	$q_{\mathsf{R}} = \Omega\left(\min\left\{\frac{1}{\varepsilon}\sqrt{\frac{n}{k}},\ \sqrt{\frac{k}{\varepsilon}} + \sqrt{\frac{n}{k}}\right\}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}}\sqrt{\ell + q_{\mathsf{G}}} = \Omega\left(\frac{\sqrt{k}}{\varepsilon}\right)$
• with copies of the state or state-generating oracle	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $\ell + q_{\mathsf{G}} \geq 1$
• with membership oracle:	$q_{\mathsf{R}} = \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ and $q_{\mathsf{M}} = \Omega\left(\sqrt{\frac{n}{k}}\right)$

Proposition. It is possible to solve the approximate counting problem using any of the following input oracles $O\left(\frac{1}{\varepsilon}\sqrt{\frac{n}{k}}\right)$ times: the state-generating, the reflecting, or the membership one.

Proof. For the membership oracle, this is just quantum counting. Otherwise, we reflect about the state ψ_x, which does not change the way the algorithm works.

Copies of the state	$\ell = \Omega\left(\min\left\{k,\ \dfrac{\sqrt{k}}{\varepsilon},\ \dfrac{n}{k\varepsilon^2}\right\}\right)$
Membership oracle	$q_{\mathsf{M}} = \Omega\left(\dfrac{1}{\varepsilon}\sqrt{\dfrac{n}{k}}\right)$
State-generating oracle • with copies of the state	$q_{\mathsf{G}} = \Omega\left(\min\left\{\dfrac{1}{\varepsilon}\sqrt{\dfrac{n}{k}},\ \dfrac{k^{1/3}}{\varepsilon^{2/3}}\right\}\right)$ $q_{\mathsf{G}}\sqrt{\ell} = \Omega\left(\dfrac{\sqrt{k}}{\varepsilon}\right)$
Reflecting oracle • with copies of the state or state-generating oracle • with copies of the state or state-generating oracle • with membership oracle:	$q_{\mathsf{R}} = \Omega\left(\min\left\{\dfrac{1}{\varepsilon}\sqrt{\dfrac{n}{k}},\ \sqrt{\dfrac{k}{\varepsilon}} + \sqrt{\dfrac{n}{k}}\right\}\right)$ $q_{\mathsf{R}}\sqrt{\ell + q_{\mathsf{G}}} = \Omega\left(\dfrac{\sqrt{k}}{\varepsilon}\right)$ $q_{\mathsf{R}} = \Omega\left(\sqrt{\dfrac{k}{\varepsilon}}\right)$ and $\ell + q_{\mathsf{G}} \geq 1$ $q_{\mathsf{R}} = \Omega\left(\sqrt{\dfrac{k}{\varepsilon}}\right)$ and $q_{\mathsf{M}} = \Omega\left(\sqrt{\dfrac{n}{k}}\right)$

Proposition. Assume t distinct elements of x are given to the algorithm. Then, it is possible to solve the approximate counting problem using any of the following input oracles $O\left(\frac{1}{\varepsilon}\sqrt{\frac{k}{t}}\right)$ times: the state-generating, or the reflecting one.

Proof. Let S be the subset given to us.
Perform amplitude estimation on ψ_x, where the marked elements are the ones in S.
The amplitude is either $\sqrt{t/k}$ or $\sqrt{t/k'}$.
Since $\sqrt{t/k} = (1+\Omega(\varepsilon))\sqrt{t/k'}$, it takes $O\left(\frac{1}{\varepsilon}\sqrt{\frac{k}{t}}\right)$ queries to the reflecting or the state-generating oracles to distinguish the two cases.

Techniques

Standard Adversary Bound

Spectral formulation of the negative-weighted adversary bound

Høyer, Lee, Špalek, 2007

$$\frac{\|\Gamma\|}{\max_{j\in[n]} \|\Gamma \circ \Delta_j\|}$$

- X, Y: sets of positive and negative inputs;
- Γ: real $X \times Y$-matrix;
- Δ_j: also $X \times Y$-matrix given by

$$\Delta_j[\![x, y]\!] = 1_{x_j \neq y_j};$$

- $\Gamma \circ \Delta_j$ is Hadamard product:

$$(A \circ B)[\![x, y]\!] = A[\![x, y]\!]B[\![x, y]\!];$$

- $\|\Gamma\|$ is spectral norm.

Group $\Delta^{\mathrm{mem}} = (\Delta^{\mathrm{mem}}_{x,y})_{x\in X, y\in Y}$, where

$$\Delta^{\mathrm{mem}}_{x,y} = \bigoplus_{j\in[n]} 1_{x_j\neq y_j} = \begin{pmatrix} 1_{x_1\neq y_1} & 0 & \cdots & 0 \\ 0 & 1_{x_2\neq y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1_{x_n\neq y_n} \end{pmatrix}$$

Extend Hadamard product notation:

$$(\Gamma \circ \Delta^{\mathrm{mem}})[\![x,y]\!] = \Gamma[\![x,y]\!]\Delta^{\mathrm{mem}}_{x,y}.$$

We have

$$\Gamma \circ \Delta^{\mathrm{mem}} = \bigoplus_{j\in[n]} \Gamma \circ \Delta_j \quad \Longrightarrow \quad \|\Gamma \circ \Delta^{\mathrm{mem}}\| = \max_{j\in[n]} \|\Gamma \circ \Delta_j\|.$$

Standard Input Oracle

Spectral formulation of the negative-weighted adversary bound
Høyer, Lee, Špalek, 2007

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta^{\mathrm{mem}}\|}$$

We say that $\Delta^{\mathrm{mem}} = (\Delta^{\mathrm{mem}}_{x,y})_{x\in X, y\in Y}$, where

$$\Delta^{\mathrm{mem}}_{x,y} = \bigoplus_{j\in[n]} 1_{x_j \neq y_j} = \begin{pmatrix} 1_{x_1\neq y_1} & 0 & \cdots & 0 \\ 0 & 1_{x_2\neq y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1_{x_n \neq y_n} \end{pmatrix}$$

represents the standard input oracle.

General Input Oracle

General input oracle: arbitrary unitary O_x with $x \in X \cup Y$.

Belovs, 2019

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

General input oracle is represented by

$$\Delta_{x,y} = O_x - O_y.$$

General Input Oracle

General input oracle: arbitrary unitary O_x with $x \in X \cup Y$.
Belovs, 2019

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

General input oracle is represented by

$$\Delta_{x,y} = O_x - O_y.$$

We can use for the oracle reflecting about ψ_x:

$$\Delta^{\text{refl}}_{x,y} = (2\psi_x\psi_x^* - I) - (2\psi_y\psi_y^* - I) = 2(\psi_x\psi_x^* - \psi_y\psi_y^*).$$

State-Generating Input Oracle

State-generating input oracle: maps $|0\rangle \mapsto |\psi_x\rangle$.

Belovs, 2019

Is represented by

$$\Delta^{\text{gen}}_{x,y} = (\psi_x \oplus \psi_x^*) - (\psi_y \oplus \psi_y^*).$$

$$\psi_x \oplus \psi_x^* = \begin{pmatrix} | & 0 & 0 & 0 \\ \psi_x & 0 & 0 & 0 \\ | & 0 & 0 & 0 \\ 0 & - & \psi_x^* & - \end{pmatrix}.$$

Multiple Input Oracles

Adversary bound with general input oracles:

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

How can we account for several input oracles?

Multiple Input Oracles

Adversary bound with general input oracles:

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

How can we account for several input oracles?

If $\|\Gamma\| = 1$, then the algorithm needs $\Omega\left(\frac{1}{\|\Gamma \circ \Delta\|}\right)$ queries to solve the problem.

Multiple Input Oracles

Adversary bound with general input oracles:

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

How can we account for several input oracles?

If $\|\Gamma\| = 1$, then the algorithm needs $\Omega\left(\frac{1}{\|\Gamma \circ \Delta\|}\right)$ queries to solve the problem.

Now assume we have $\Delta^{(1)}$ and $\Delta^{(2)}$.

Multiple Input Oracles

Adversary bound with general input oracles:

$$\frac{\|\Gamma\|}{\|\Gamma \circ \Delta\|}$$

How can we account for several input oracles?

If $\|\Gamma\| = 1$, then the algorithm has to make

- $\Omega\left(\frac{1}{\|\Gamma \circ \Delta^{(1)}\|}\right)$ queries to the first oracle; or
- $\Omega\left(\frac{1}{\|\Gamma \circ \Delta^{(2)}\|}\right)$ queries to the second oracle

to solve the problem.

Let Γ be an $X \times Y$-matrix with $\|\Gamma\| = 1$.
Then, the algorithm has to make

- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{mem}}\|}\right)$ queries to the membership oracle; or
- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{refl}}\|}\right)$ queries to the reflecting oracle; or
- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{gen}}\|}\right)$ queries to the state-generating oracle

to solve the problem.

Copies of the state

With copies of the state ψ_x, the problem is a state-conversion problem.

Ambainis, Magnin, Rötteler, Roland, 2011

Define the Gram matrix Ψ with

$$\Psi[\![x, y]\!] = \langle \psi_x, \psi_y \rangle.$$

The Gram matrix of ℓ copies is $\Psi^{\circ \ell}$.

Formulation of the Optimisation Problem

Let Γ be an $X \times Y$-matrix with $\|\Gamma\| = 1$ and $\|\Gamma \circ \Psi^{\circ \ell}\| = \Omega(1)$. Then, having ℓ copies of the state ψ_x, the algorithm has to make

- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{mem}}\|}\right)$ queries to the membership oracle; or
- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{refl}}\|}\right)$ queries to the reflecting oracle; or
- $\Omega\left(\dfrac{1}{\|\Gamma \circ \Delta^{\mathrm{gen}}\|}\right)$ queries to the state-generating oracle

to solve the problem.

Representation Theory

The matrix Γ is symmetric with respect to the permutations of $[n]$.

$$\Gamma = \sum_{j=0}^{k} \gamma_j \Phi_j,$$

where Φ_j are the isomorphisms between the copies of the irreps of the symmetric group in $\mathbb{R}^Y$ and $\mathbb{R}^X$.

$$\|\Gamma\| = \max_j |\gamma_j|.$$

Some calculations. . .

Some Quantities

Let $k' = (1+\varepsilon)k$.

$$\phi_{j,0} := \sqrt{\frac{j(k-j+1)(n-k-j+1)}{(n-2j+2)(n-2j+1)k}}, \qquad \phi'_{j,0} := \sqrt{\frac{j(k'-j+}{(n-2j+}}$$

$$\phi_{j,1} := \sqrt{\frac{k}{n}}, \qquad \phi'_{j,1} := \sqrt{\frac{k'}{n}},$$

$$\phi_{j,2} := \frac{n-2k}{\sqrt{nk}}\sqrt{\frac{j(n-j+1)}{(n-2j+2)(n-2j)}}, \qquad \phi'_{j,2} := \frac{n-2k'}{\sqrt{nk'}}\sqrt{\frac{}{(n}}$$

$$\phi_{j,3} := \sqrt{\frac{(n-j+1)(k-j)(n-k-j)}{(n-2j+1)(n-2j)k}}, \qquad \phi'_{j,3} := \sqrt{\frac{(n-j+1}{(n-2}}$$

$$\phi_j = \begin{pmatrix} \phi_{j,0} \\ \phi_{j,1} \\ \phi_{j,2} \\ \phi_{j,3} \end{pmatrix} \qquad \phi'_j = \begin{pmatrix} \phi'_{j,0} \\ \phi'_{j,1} \\ \phi'_{j,2} \\ \phi'_{j,3} \end{pmatrix}$$

$$\widetilde{\phi}_j = \begin{pmatrix} \gamma_{j-1}\phi_{j,0} \\ \gamma_j\phi_{j,1} \\ \gamma_j\phi_{j,2} \\ \gamma_{j+1}\phi_{j,3} \end{pmatrix} \qquad \widetilde{\phi}'_j = \begin{pmatrix} \gamma_{j-1}\phi'_{j,0} \\ \gamma_j\phi'_{j,1} \\ \gamma_j\phi'_{j,2} \\ \gamma_{j+1}\phi'_{j,3} \end{pmatrix}.$$

$$\Gamma \circ \Psi = \sum_{j=0}^{k} \langle \phi_j, \widetilde{\phi}'_j \rangle \Phi_j$$

$$\|\Gamma \circ \Delta^{\text{gen}}\| = \max_j \max \left\{ \left\| \widetilde{\phi}'_j - \gamma_j \phi_j \right\|, \left\| \gamma_j \phi'_j - \widetilde{\phi}_j \right\| \right\}$$

$$\|\Gamma \circ \Delta^{\text{refl}}\| = \max_j \left\| \phi'_j \widetilde{\phi}'_j{}^* - \widetilde{\phi}_j \phi_j^* \right\|$$

$$\|\Gamma \circ \Delta^{\text{mem}}\| = \max_j \max \left\{ \left| \frac{\sqrt{(k-j)(n-k'-j)}}{n-2j} \gamma_j - \frac{\sqrt{(k'-j)(n-k-j)}}{n-2j} \gamma_{j+1} \right|, \left| \frac{\sqrt{(k'-j)(n-k-j)}}{n-2j} \gamma_j - \frac{\sqrt{(k-j)(n-k'-j)}}{n-2j} \gamma_{j+1} \right| \right\}.$$

Summary

Summary

- Demonstrated how to use the new version of the adversary bound
  - for various input oracles;
  - to prove trade-offs between them.
- Developed ancillary lemmas for subsets of a uniform set.

Open Problems

- Do we need representation theory?

Thank you!