Lifting with XOR

Suhail Sherif, Tata Institute of Fundamental Research

Based on work done with Arkadev Chattopadhyay and Nikhil Mande

Dec 14, 2019

Statements made in these slides are for representational purposes and are not guaranteed to be entirely accurate.

Communication Complexity for Communication Complexity's Sake

Communication Complexity for Communication Complexity's Sake

- This talk is aimed at a better understanding of communication complexity.
- In this talk, we focus on two parties (Alice and Bob) computing a total Boolean function.

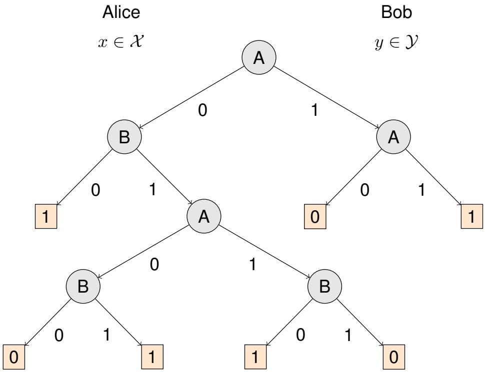
Communication Complexity for Communication Complexity's Sake

- This talk is aimed at a better understanding of communication complexity.
- In this talk, we focus on two parties (Alice and Bob) computing a total Boolean function.
- XOR functions feature in this talk because
  - They are structured enough to reason about.

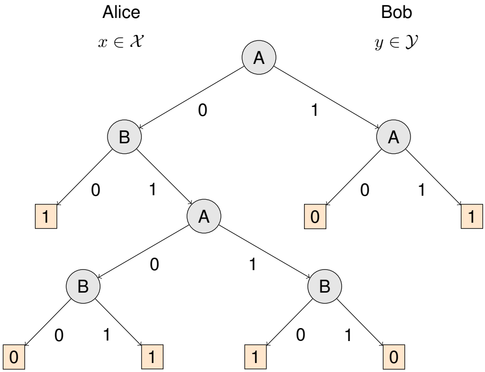
Communication Complexity for Communication Complexity's Sake

- This talk is aimed at a better understanding of communication complexity.
- In this talk, we focus on two parties (Alice and Bob) computing a total Boolean function.
- XOR functions feature in this talk because
  - They are structured enough to reason about.
  - There is enough mystery about them for them to be interesting.

A Communication Protocol



A Communication Protocol

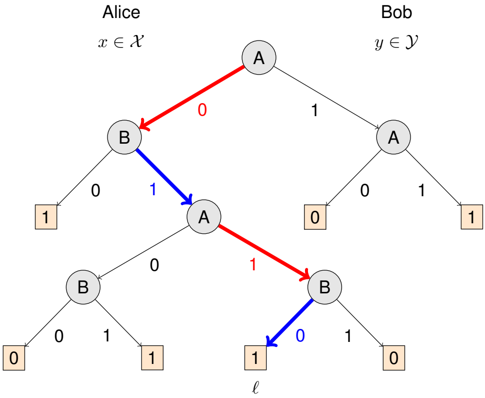


(x, y) is accepted

$\Leftrightarrow$

(x, y) reaches a 1-leaf.

A Communication Protocol



(x, y) is accepted

$\Leftrightarrow$

(x, y) reaches a 1-leaf.

Inputs that reach ℓ

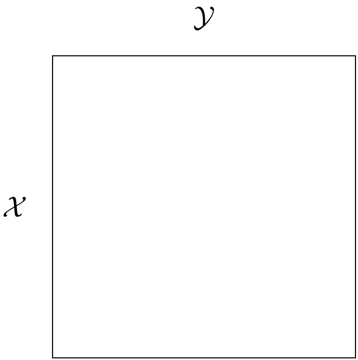
$=$

$\{x : x$ answers red$\}$

$\times$

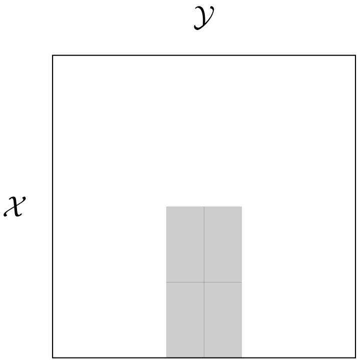
$\{y : y$ answers blue$\}$.

Rank



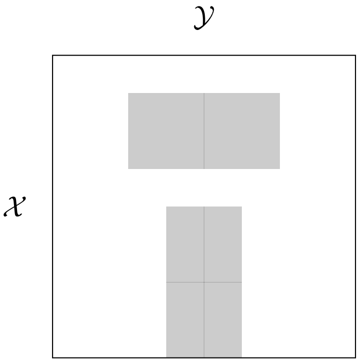
Building the truth table for the function computed by the protocol.

Rank



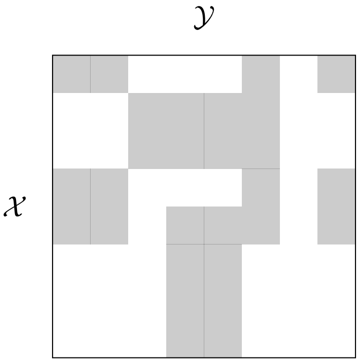
Inputs that reach leaf ℓ contribute a rank 1 matrix.

Rank



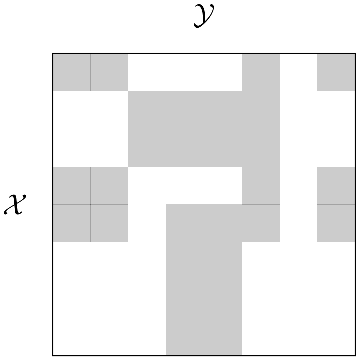
Inputs that reach leaves ℓ_1 or ℓ_2 form a rank ≤ 2 matrix.

Rank



Inputs that reach any 1 leaf form a rank $\leq 2^c$ matrix.

Rank



Cost c protocol for F

$$\Longrightarrow$$

M_F has rank $\leq 2^c$.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks '88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathsf{rank}(F)$$

Protocol-Rank Equivalence?

Conjecture (Lovász Saks ‘88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathsf{rank}(F)$$

- Connects comm comp measure with algebraic measure.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks '88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathsf{rank}(F)$$

- Connects comm comp measure with algebraic measure. Known analogous connections have been useful.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks '88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathrm{rank}(F)$$

- Connects comm comp measure with algebraic measure. Known analogous connections have been useful.
- Has connections to graph colouring, low degree polynomials.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks ‘88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathsf{rank}(F)$$

- Connects comm comp measure with algebraic measure. Known analogous connections have been useful.
- Has connections to graph colouring, low degree polynomials.

For: [Lovett ‘13] showed that $D(F) \lesssim O\left(\sqrt{\mathsf{rank}(F)}\right)$.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks '88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathrm{rank}(F)$$

- Connects comm comp measure with algebraic measure. Known analogous connections have been useful.
- Has connections to graph colouring, low degree polynomials.

For: [Lovett '13] showed that $D(F) \lesssim O\left(\sqrt{\mathrm{rank}(F)}\right)$.

Against: [Göös Pitassi Watson '15] showed that $\alpha \geq 2$.

Protocol-Rank Equivalence?

Conjecture (Lovász Saks ‘88)

$$\exists \textit{ constant } \alpha \textit{ s.t. } D(F) \leq \log^{\alpha} \mathrm{rank}(F)$$

- Connects comm comp measure with algebraic measure. Known analogous connections have been useful.
- Has connections to graph colouring, low degree polynomials.

For: [Lovett ‘13] showed that $D(F) \lesssim O\left(\sqrt{\mathrm{rank}(F)}\right)$.

Against: [Göös Pitassi Watson ‘15] showed that $\alpha \geq 2$.

Fun fact: LRC is True if you restrict the rank decomposition to be nonnegative.

A Randomized Communication Protocol

Alice

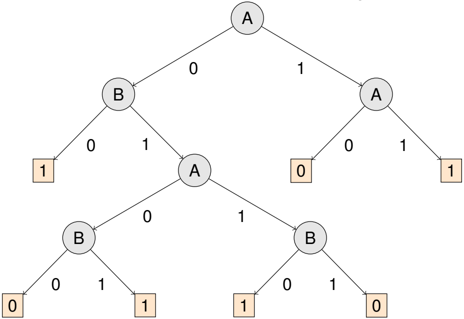
$x \in \mathcal{X}$

r_A, a string of random bits

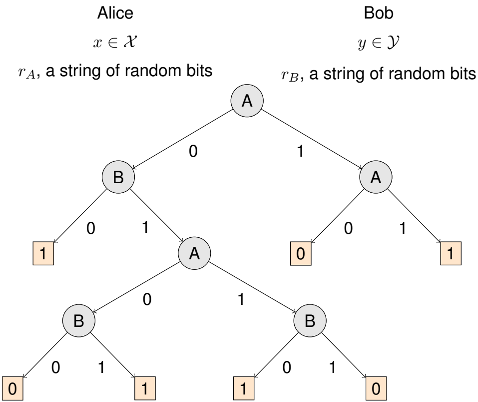
Bob

$y \in \mathcal{Y}$

r_B, a string of random bits

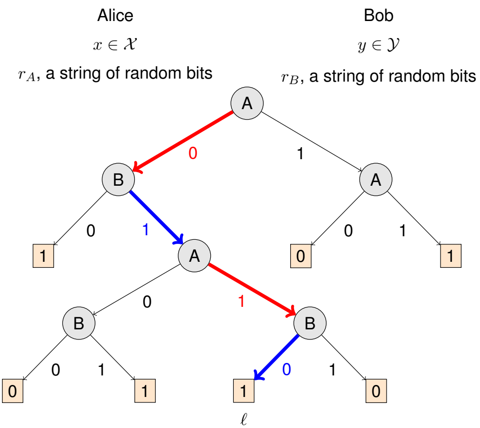


A Randomized Communication Protocol



$$\Pr[(x, y) \text{ is accepted}]$$
$$=$$
$$\Pr[(x, y) \text{ reaches a } 1\text{-leaf}].$$

A Randomized Communication Protocol



$\Pr[(x, y)$ is accepted$]$
$=$
$\Pr[(x, y)$ reaches a 1-leaf$]$.

$\Pr[(x, y)$ reaches $\ell]$
$=$
$\Pr_{r_A}[x$ answers red$]$
$\times$
$\Pr_{r_B}[y$ answers blue$]$.

Small Approximate Rank

$$\Pr_{r_B}[y \text{ answers blue}]$$

$$
\begin{array}{c c}
 & \begin{array}{cccc} 0 & .5 & 0 & .6 \end{array} \\
\Pr_{r_A}[x \text{ answers red}] \quad \begin{array}{c} .5 \\ .8 \\ 0 \\ 0 \end{array} & \left[\begin{array}{cccc} & .25 & & .3 \\ & .4 & & .48 \\ & & & \\ & & & \end{array}\right]
\end{array}
$$

$\Pr[(x, y) \text{ reaches } \ell]$ is a rank 1 matrix.

Small Approximate Rank

$$
\begin{array}{cc}
 & \mathrm{Pr}_{r_B}[y \text{ answers blue}] \\
 & \begin{array}{cccc} 0 & .5 & 0 & .6 \end{array} \\
\mathrm{Pr}_{r_A}[x \text{ answers red}] \; \begin{array}{c} .5 \\ .8 \\ 0 \\ 0 \end{array} & \begin{bmatrix} & .25 & & .3 \\ & .4 & & .48 \\ & & & \\ & & & \end{bmatrix}
\end{array}
$$

$\mathrm{Pr}[(x, y)$ reaches $\ell]$ is a rank 1 matrix.

$\mathrm{Pr}[(x, y)$ is accepted$]$ is a rank $\leq 2^c$ matrix.

$$\boxed{\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}$$

M_F

$$\boxed{\begin{matrix} .8 & .9 & .1 & .2 \\ 0 & .9 & .1 & .1 \\ 0 & .1 & .8 & 0 \\ .1 & 0 & 0 & 1 \end{matrix}}$$

$M_{\text{Pr of accepting}}$

$$\boxed{\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}$$

$$M_F$$

$$\boxed{\begin{matrix} .8 & .9 & .1 & .2 \\ 0 & .9 & .1 & .1 \\ 0 & .1 & .8 & 0 \\ .1 & 0 & 0 & 1 \end{matrix}}$$

$$M_{\text{Pr of accepting}}$$

Rank $\leq 2^c$

$$\boxed{\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}$$

$$M_F$$

Approx. Rank $\leq 2^c$

$$\boxed{\begin{matrix} .8 & .9 & .1 & .2 \\ 0 & .9 & .1 & .1 \\ 0 & .1 & .8 & 0 \\ .1 & 0 & 0 & 1 \end{matrix}}$$

$$M_{\text{Pr of accepting}}$$

Rank $\leq 2^c$

$$\boxed{\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}$$

M_F

Approx. Rank $\leq 2^c$

$$\boxed{\begin{matrix} .8 & .9 & .1 & .2 \\ 0 & .9 & .1 & .1 \\ 0 & .1 & .8 & 0 \\ .1 & 0 & 0 & 1 \end{matrix}}$$

$M_{\text{Pr of accepting}}$

Rank $\leq 2^c$

$$\log \mathsf{rank}_{1/3}(F) \leq c.$$

Protocol-Rank Equivalence?

Conjecture (ForgeGod '05, Lee Shraibman '07)

$$\exists \textit{ constant } \beta \textit{ s.t. } R(F) \leq \log^{\beta} \mathsf{rank}_{1/3}(F)$$

Protocol-Rank Equivalence?

Conjecture (ForgeGod '05, Lee Shraibman '07)

$$\exists \textit{ constant } \beta \textit{ s.t. } R(F) \leq \log^{\beta} \mathsf{rank}_{1/3}(F)$$

For a randomized protocol, the number of bits exchanged in the worst case, R(f), is conjectured to be polynomially related to the following absurd formula:

min{rank(M'_f): $M'_f \in \mathbb{R}^{2^n \times 2^n}$, $(M_f - M'_f)_\infty \leq 1/3$}.

Figure: Screenshot from "Communication complexity - Wikipedia" (Dec '05)

Protocol-Rank Equivalence?

Conjecture (ForgeGod '05, Lee Shraibman '07)

$\exists$ *constant* β *s.t.* $R(F) \leq \log^{\beta} \mathsf{rank}_{1/3}(F)$

Implies the LRC! [Gavinsky Lovett '13]

Protocol-Rank Equivalence?

Conjecture (ForgeGod '05, Lee Shraibman '07)

$\exists$ *constant* β *s.t.* $R(F) \leq \log^{\beta} \mathsf{rank}_{1/3}(F)$

Implies the LRC! [Gavinsky Lovett '13]

Set Disjointness shows that $\beta \geq 2$. [Kalyanasundaram Schnitger '92, Razborov '92]

Protocol-Rank Equivalence?

Conjecture (ForgeGod '05, Lee Shraibman '07)

$\exists$ *constant* β *s.t.* $R(F) \leq \log^{\beta} \mathsf{rank}_{1/3}(F)$

Implies the LRC! [Gavinsky Lovett '13]

Set Disjointness shows that $\beta \geq 2$. [Kalyanasundaram Schnitger '92, Razborov '92]

[Göös Jayram Pitassi Watson '17] showed that $\beta \geq 4$.

Nonnegative Ranks

- It is known that $D(F) \leq O(\log^2(\mathsf{rank}^+(F)))$. [Lovász '90]

Nonnegative Ranks

- It is known that $D(F) \leq O(\log^2(\mathsf{rank}^+(F)))$. [Lovász '90]
- One may conjecture that $R(F) \leq \log^{O(1)}(\mathsf{rank}^+_{1/3}(F))$.

Nonnegative Ranks

- It is known that $D(F) \leq O(\log^2(\mathsf{rank}^+(F)))$. [Lovász '90]
- One may conjecture that $R(F) \leq \log^{O(1)}(\mathsf{rank}^+_{1/3}(F))$.
- Or the more reasonable conjecture that

$$R(F) \leq \log^{O(1)}(\max\left\{\mathsf{rank}^+_{1/3}(F), \mathsf{rank}^+_{1/3}(\overline{F}))\right\}).$$

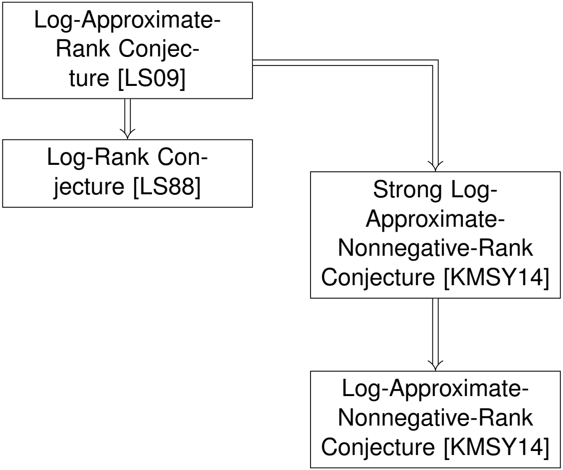
Nonnegative Ranks

- It is known that $D(F) \leq O(\log^2(\mathsf{rank}^+(F)))$. [Lovász '90]
- One may conjecture that $R(F) \leq \log^{O(1)}(\mathsf{rank}^+_{1/3}(F))$.
- Or the more reasonable conjecture that

$$R(F) \leq \log^{O(1)}(\max\left\{\mathsf{rank}^+_{1/3}(F), \mathsf{rank}^+_{1/3}(\overline{F}))\right\}).$$

- [Kol Moran Shpilka Yehudayoff '14] did.

To Cache


Log-Approximate-Rank Conjecture [LS09]
Log-Rank Conjecture [LS88]
Strong Log-Approximate-Nonnegative-Rank Conjecture [KMSY14]
Log-Approximate-Nonnegative-Rank Conjecture [KMSY14]

To Cache

$$
\begin{tikzcd}[arrows=Rightarrow]
\begin{array}{c} R(F) \text{ vs.} \\ \log \mathsf{rank}_{1/3}(F) \end{array} \arrow[d] \arrow[dr] & \\
\begin{array}{c} D(F) \text{ vs.} \\ \log \mathsf{rank}(F) \end{array} & \begin{array}{c} R(F) \text{ vs.} \\ \log \mathsf{rank}^{+}_{1/3}(F) \end{array} \arrow[d] \\
 & \begin{array}{c} R(F) \text{ vs.} \\ \log \left(\mathsf{rank}^{+}_{1/3}(F) + \mathsf{rank}^{+}_{1/3}(\overline{F}) \right) \end{array}
\end{tikzcd}
$$

XOR Compositions

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam '19]

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam '19]
- Patent expired in 1936, so we can feel free to compose with XOR.

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam '19]
- Patent expired in 1936, so we can feel free to compose with XOR.

$$f \circ \text{XOR}(x, y) = f(z) \text{ where } z = x \oplus y.$$

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam ‘19]
- Patent expired in 1936, so we can feel free to compose with XOR.

$$f \circ \mathrm{XOR}(x, y) = f(z) \text{ where } z = x \oplus y.$$

- Neither Alice nor Bob have any idea about any bit of z.

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam '19]
- Patent expired in 1936, so we can feel free to compose with XOR.

$$f \circ \text{XOR}(x, y) = f(z) \text{ where } z = x \oplus y.$$

- Neither Alice nor Bob have any idea about any bit of z.
- But, parity can be computed easily.

XOR Compositions

Encodes an n-bit string into 2 n-bit strings, none of which give any information about the original.

- XOR is good for this purpose. [U.S. Patent 1,310,719, Gilbert Vernam '19]
- Patent expired in 1936, so we can feel free to compose with XOR.

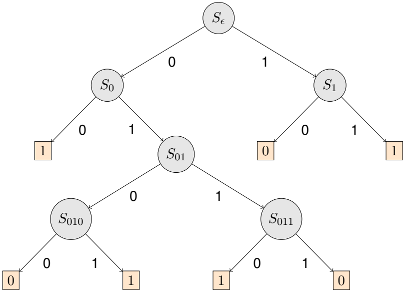
$$f \circ \text{XOR}(x, y) = f(z) \text{ where } z = x \oplus y.$$

- Neither Alice nor Bob have any idea about any bit of z.
- But, parity can be computed easily.
- Expect to lift from Parity Decision Trees (allowed queries are parities, not just bits).

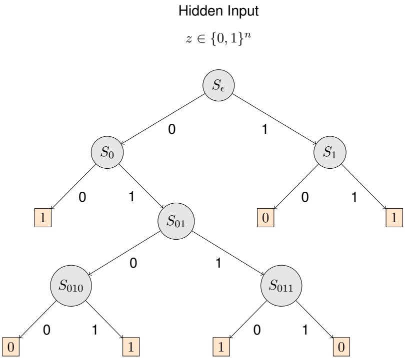
Parity Decision Trees (PDTs)

Hidden Input

$z \in \{0,1\}^n$



Parity Decision Trees (PDTs)

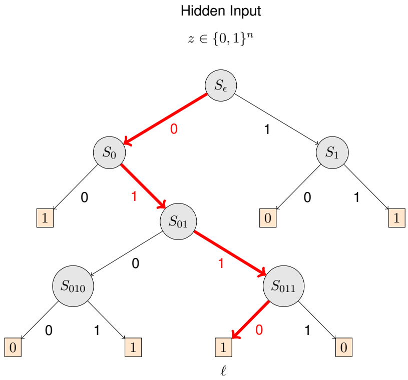


z is accepted

$\Leftrightarrow$

z reaches a 1-leaf.

Parity Decision Trees (PDTs)



z is accepted

$\Leftrightarrow$

z reaches a 1-leaf.

Inputs that reach ℓ

$=$

$\{z : z$ satisfies red constraints$\}$

OR: Hard for deterministic PDTs

- You have to reject just one input.
- Any leaf at depth d has 2^{-d} fraction of inputs.
- $\implies$ there must be a 0-leaf at depth n.

OR: Easy for randomized PDTs

- Randomly sample $S \subseteq_{\mathcal{U}} [n]$.
- Query $\oplus_S z$.

$$\Pr[\text{Query outputs } 0] = \begin{cases} 1 \text{ if } z = 0^n \\ 1/2 \text{ if } z \neq 0^n \end{cases}$$

Deterministic vs Randomized

$\text{OR} \circ \text{XOR}$ is the canonical separation of deterministic and randomized communication complexity.

Deterministic vs Randomized

$\mathrm{OR} \circ \mathrm{XOR}$ is the canonical separation of deterministic and randomized communication complexity.

- There are no such examples known with other small gadgets.

Deterministic vs Randomized

$\mathrm{OR} \circ \mathrm{XOR}$ is the canonical separation of deterministic and randomized communication complexity.

- There are no such examples known with other small gadgets.
- For gadgets that lift query complexity, it is impossible.

Deterministic vs Randomized

$\mathrm{OR} \circ \mathrm{XOR}$ is the canonical separation of deterministic and randomized communication complexity.

- There are no such examples known with other small gadgets.
- For gadgets that lift query complexity, it is impossible.

What other potentially fruitful properties do PDTs have?

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.
  Given a node in a PDT, an RPDT can tell whether the input will reach it.

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.
  Given a node in a PDT, an RPDT can tell whether the input will reach it.
- Hence RPDTs can balance PDTs. So $RPDT(f) \leq \log PDT^{leaf}(f)$.

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.
  Given a node in a PDT, an RPDT can tell whether the input will reach it.
- Hence RPDTs can balance PDTs. So $RPDT(f) \leq \log PDT^{leaf}(f)$.
- Open Problem: Is there anything else that RPDTs can do?

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.
  Given a node in a PDT, an RPDT can tell whether the input will reach it.
- Hence RPDTs can balance PDTs. So $RPDT(f) \leq \log PDT^{leaf}(f)$.
- Open Problem: Is there anything else that RPDTs can do?
- Lifted Open Problem: Is there an XOR function easy for randomized communication but hard for P^{EQ} protocols?

When else is randomness powerful?

- RPDTs can compute affine subspaces the same way it does AND.
  Given a node in a PDT, an RPDT can tell whether the input will reach it.
- Hence RPDTs can balance PDTs. So $RPDT(f) \leq \log PDT^{leaf}(f)$.
- Open Problem: Is there anything else that RPDTs can do?
- Lifted Open Problem: Is there an XOR function easy for randomized communication but hard for P^{EQ} protocols?
  For general functions, this question was answered very recently. [Chattopadhyay Lovett Vinyals '19] exhibited a function with the separation.

Meet the PDT measures: Fourier Analysis

$$AND : \{\pm 1\}^n \to \{0, 1\}$$

Meet the PDT measures: Fourier Analysis

$$AND : \{\pm 1\}^n \to \{0,1\}$$

$$AND(z_1, z_2, z_3) = \frac{1}{8} - \frac{1}{8}z_1 - \frac{1}{8}z_2 - \frac{1}{8}z_3 + \frac{1}{8}z_1z_2 + \frac{1}{8}z_1z_3 + \frac{1}{8}z_2z_3 - \frac{1}{8}z_1z_2z_3$$

Meet the PDT measures: Fourier Analysis

$$AND : \{\pm 1\}^n \to \{0,1\}$$

$$AND(z_1, z_2, z_3) = \frac{1}{8} - \frac{1}{8}z_1 - \frac{1}{8}z_2 - \frac{1}{8}z_3 + \frac{1}{8}z_1z_2 + \frac{1}{8}z_1z_3 + \frac{1}{8}z_2z_3 - \frac{1}{8}z_1z_2z_3$$

Sparsity: $\mathsf{sp}(f)$ is the number of non-zero coefficients.
ℓ_1: $||\widehat{f}||_1$ is the sum of the absolute values of the coefficients.

Meet the PDT measures: Fourier Analysis

$$AND : \{\pm 1\}^n \to \{0,1\}$$

$$AND(z_1,z_2,z_3) = \frac{1}{8} - \frac{1}{8}z_1 - \frac{1}{8}z_2 - \frac{1}{8}z_3 + \frac{1}{8}z_1z_2 + \frac{1}{8}z_1z_3 + \frac{1}{8}z_2z_3 - \frac{1}{8}z_1z_2z_3$$

Sparsity: $\mathsf{sp}(f)$ is the number of non-zero coefficients.
ℓ_1: $\left|\left|\widehat{f}\right|\right|_1$ is the sum of the absolute values of the coefficients.

- Every leaf is an affine subspace in $\mathbb{F}_2$.

Meet the PDT measures: Fourier Analysis

$$AND : \{\pm 1\}^n \to \{0,1\}$$

$$AND(z_1, z_2, z_3) = \frac{1}{8} - \frac{1}{8}z_1 - \frac{1}{8}z_2 - \frac{1}{8}z_3 + \frac{1}{8}z_1 z_2 + \frac{1}{8}z_1 z_3 + \frac{1}{8}z_2 z_3 - \frac{1}{8}z_1 z_2 z_3$$

Sparsity: $\mathsf{sp}(f)$ is the number of non-zero coefficients.
ℓ_1: $||\widehat{f}||_1$ is the sum of the absolute values of the coefficients.

- Every leaf is an affine subspace in $\mathbb{F}_2$.
- For a function f computable by a depth-k PDT, $\mathsf{sp}(f) \leq 2^{2k}$ and $||\widehat{f}||_1 \leq 2^k$.
- For a function f computable by a depth-k RPDT, $||\widehat{f}||_{1,1/3} \leq 2^k$.

Lifting with XOR

Measure for f	Measure for $F = f \circ \text{XOR}$

Lifting with XOR

Measure for f	Measure for $F = f \circ \text{XOR}$
$\text{sp}(f)$	$\text{rank}(F) = \text{sp}(f)$

If $PDT(f) \leq \log^{O(1)} \text{sp}(f)$, then the LRC is true for XOR functions!

Lifting with XOR

Measure for f	Measure for $F = f \circ \text{XOR}$
$\mathsf{sp}(f)$	$\mathsf{rank}(F) = \mathsf{sp}(f)$
$\mathsf{sp}_{1/3}(f)$	$\mathsf{sp}_{1/3'}(f)/n \leq \mathsf{rank}_{1/3}(F) \leq \mathsf{sp}_{1/3}(f)$

If $RPDT(f) \leq \log^{O(1)} \mathsf{sp}_{1/3}(f)$, then the LARC is true for XOR functions!

Lifting with XOR

Measure for f	Measure for $F = f \circ \mathsf{XOR}$
$\mathsf{sp}(f)$	$\mathsf{rank}(F) = \mathsf{sp}(f)$
$\mathsf{sp}_{1/3}(f)$	$\mathsf{sp}_{1/3'}(f)/n \leq \mathsf{rank}_{1/3}(F) \leq \mathsf{sp}_{1/3}(f)$
$\lvert\lvert\widehat{f}\rvert\rvert_1$	$\lvert\lvert\widehat{F}\rvert\rvert_1 = \lvert\lvert\widehat{f}\rvert\rvert_1$

Why are we looking at $||\widehat{F}||_1$?

Lifting with XOR

Measure for f	Measure for $F = f \circ \text{XOR}$
$\mathsf{sp}(f)$	$\mathsf{rank}(F) = \mathsf{sp}(f)$
$\mathsf{sp}_{1/3}(f)$	$\mathsf{sp}_{1/3'}(f)/n \leq \mathsf{rank}_{1/3}(F) \leq \mathsf{sp}_{1/3}(f)$
$\lvert\lvert\widehat{f}\rvert\rvert_1$	$\lvert\lvert\widehat{F}\rvert\rvert_1 = \lvert\lvert\widehat{f}\rvert\rvert_1$

Why are we looking at $||\widehat{F}||_1$?

Grolmusz [Grolmusz '97] conjectured: $R(F) \leq \log^{O(1)} ||\widehat{F}||_1$

Lifting with XOR

Measure for f	Measure for $F = f \circ \text{XOR}$
$\mathsf{sp}(f)$	$\mathsf{rank}(F) = \mathsf{sp}(f)$
$\mathsf{sp}_{1/3}(f)$	$\mathsf{sp}_{1/3'}(f)/n \leq \mathsf{rank}_{1/3}(F) \leq \mathsf{sp}_{1/3}(f)$
$\lvert\lvert\widehat{f}\rvert\rvert_1$	$\lvert\lvert\widehat{F}\rvert\rvert_1 = \lvert\lvert\widehat{f}\rvert\rvert_1$
$\lvert\lvert\widehat{f}\rvert\rvert_{1,1/3}$	$\lvert\lvert\widehat{F}\rvert\rvert_{1,1/3} = \lvert\lvert\widehat{f}\rvert\rvert_{1,1/3}$

Lifting with XOR

Measure for f	Measure for $F = f \circ \mathrm{XOR}$
$\mathrm{sp}(f)$	$\mathrm{rank}(F) = \mathrm{sp}(f)$
$\mathrm{sp}_{1/3}(f)$	$\mathrm{sp}_{1/3'}(f)/n \leq \mathrm{rank}_{1/3}(F) \leq \mathrm{sp}_{1/3}(f)$
$\lvert\lvert\widehat{f}\rvert\rvert_1$	$\lvert\lvert\widehat{F}\rvert\rvert_1 = \lvert\lvert\widehat{f}\rvert\rvert_1$
$\lvert\lvert\widehat{f}\rvert\rvert_{1,1/3}$	$\lvert\lvert\widehat{F}\rvert\rvert_{1,1/3} = \lvert\lvert\widehat{f}\rvert\rvert_{1,1/3}$
$PDT(f)$	$PDT(f)^{1/6} \leq D(F) \leq 2PDT(f)$

If the LRC is true for XOR functions, then
$PDT(f) \leq \log^{O(1)} \mathrm{sp}(f)$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \left|\left|\widehat{g}\right|\right|_1 (sgn(\hat{g}(S_1)) z_{S_1})$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \left|\left|\widehat{g}\right|\right|_1 (sgn(\hat{g}(S_1))z_{S_1})$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.

$$\mathbb{E}[g_{sample}(z)] = \sum_{S \subseteq [n]} \frac{|\hat{g}(S_1)|}{\left|\left|\widehat{g}\right|\right|_1} sgn(\hat{g}(S))z_S = \sum_S \hat{g}(S) z_S = g(z)$$

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{2} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2}\right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots \right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.
- Since each term in the addition is bounded, we can use Hoeffding's Lemma.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.
- Since each term in the addition is bounded, we can use Hoeffding's Lemma.
- whp, if $T = O\left(\left|\left|\widehat{g}\right|\right|_1^2\right)$, g_{sample} approximates g on a fixed z.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.
- Since each term in the addition is bounded, we can use Hoeffding's Lemma.
- whp, if $T = O\left(\left|\left|\widehat{g}\right|\right|_1^2\right)$, g_{sample} approximates g on a fixed z.
- whp, if $T = O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$, g_{sample} approximates g on all z.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto$+\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- For any z, $g(z) = \mathbb{E}[g_{sample}(z)]$.
- Since each term in the addition is bounded, we can use Hoeffding's Lemma.
- whp, if $T = O\left(\left|\left|\widehat{g}\right|\right|_1^2\right)$, g_{sample} approximates g on a fixed z.
- whp, if $T = O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$, g_{sample} approximates g on all z.
- Approximate sparsity of g is less than $O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- Approximate sparsity of g is less than $O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$.

$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto$+\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots\right)$$

- Approximate sparsity of g is less than $O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$.
- If $\left|\left|\widehat{f}\right|\right|_{1,1/3} \leq k$, then $\mathsf{sp}_{1/3+\epsilon}(f) \leq O(k^2 n/\epsilon^2)$.

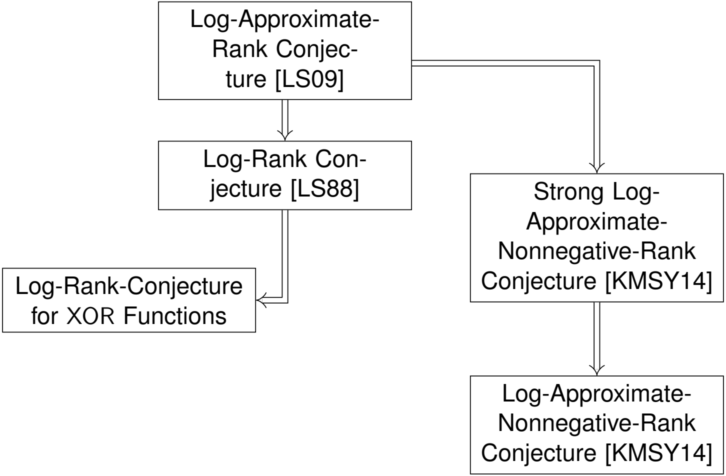
$\log \mathsf{sp}_{1/3'}(f)$, $\log \left|\left|\widehat{f}\right|\right|_{1,1/3}$: Potayto, Potahto+$\log n$

$$g(z) = \sum_S \hat{g}(S) z_S$$

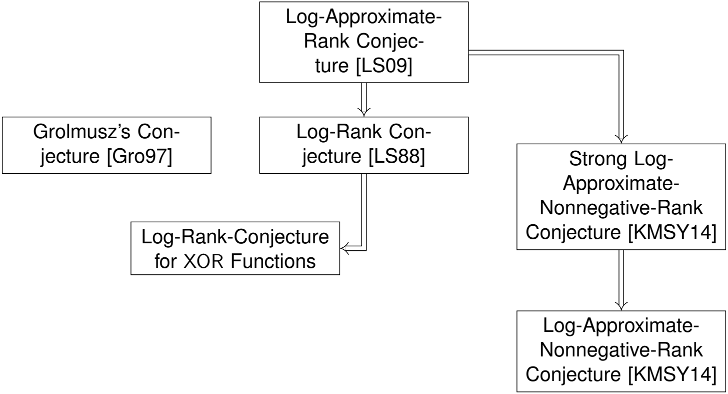
$$g_{sample}(z) = \frac{\left|\left|\widehat{g}\right|\right|_1}{T} \left(sgn(\hat{g}(S_1)) z_{S_1} + sgn(\hat{g}(S_2)) z_{S_2} + \cdots \right)$$

- Approximate sparsity of g is less than $O\left(\left|\left|\widehat{g}\right|\right|_1^2 n\right)$.
- If $\left|\left|\widehat{f}\right|\right|_{1,1/3} \leq k$, then $\mathsf{sp}_{1/3+\epsilon}(f) \leq O(k^2 n/\epsilon^2)$.
- LARC for XOR functions is "equivalent" to the corresponding ℓ_1-based conjecture for XOR functions. Implies Grolmusz' conjecture.

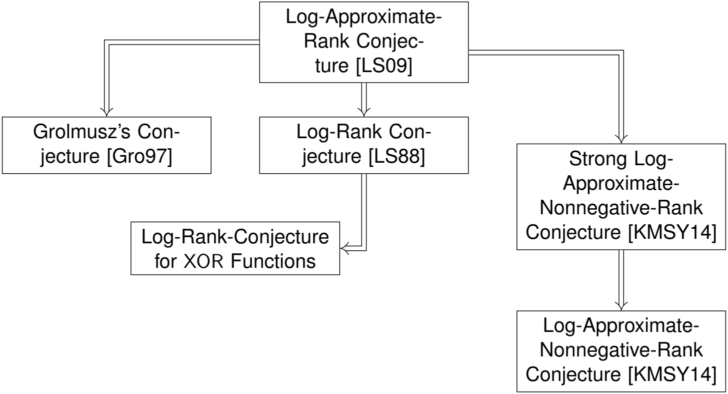
Take a Breather



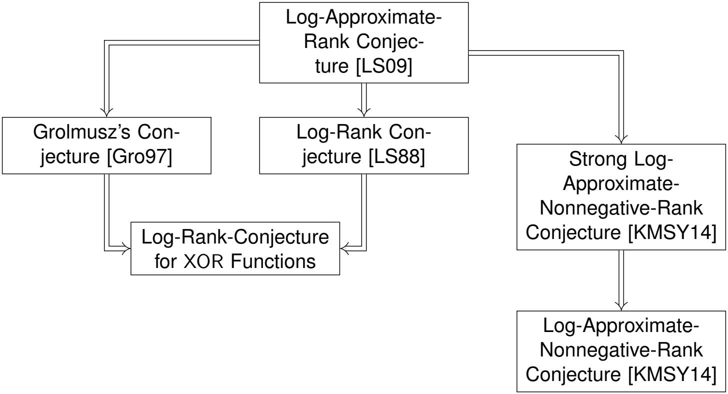
Take a Breather



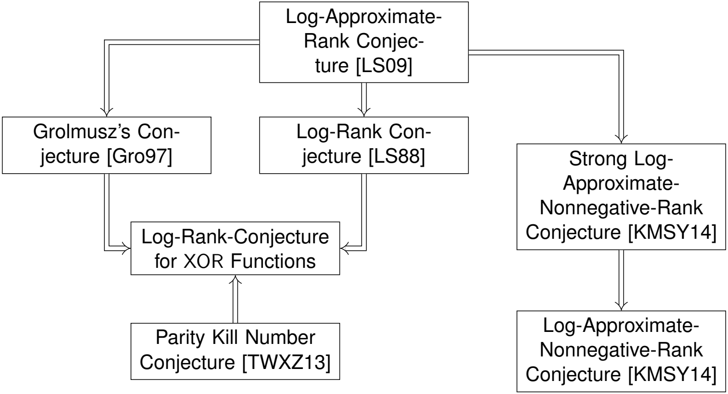
Take a Breather



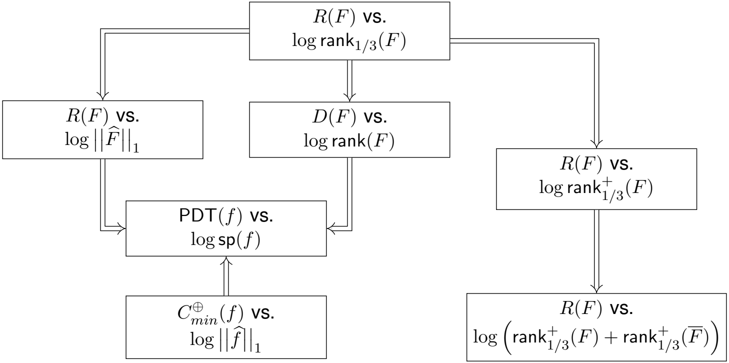
Take a Breather



Take a Breather



Take a Breather



Is $RPDT(f) \leq \log \left|\left|\widehat{f}\right|\right|_1$?

Functions with small Fourier ℓ_1 norm:

- ANDs/affine subspaces.

Is $RPDT(f) \leq \log \left|\left|\widehat{f}\right|\right|_1$?

Functions with small Fourier ℓ_1 norm:

- ANDs/affine subspaces.
- Similar to the case of leaves in a protocol: sum of few disjoint ANDs.

Is $RPDT(f) \leq \log \left|\left|\widehat{f}\right|\right|_1$?

Functions with small Fourier ℓ_1 norm:

- ANDs/affine subspaces.
- Similar to the case of leaves in a protocol: sum of few disjoint ANDs.

$$\begin{array}{cccc} & z_1 & z_2 & z_3 \\ S_1 & 0 & 0 & * \\ S_2 & 1 & * & 0 \\ S_3 & * & 1 & 1 \end{array}$$

Larger example

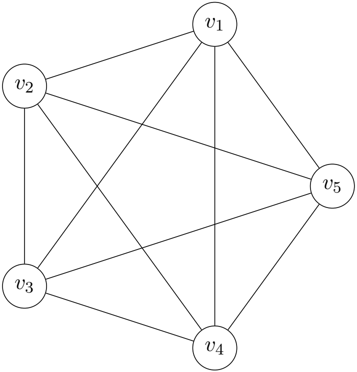
$$
\begin{array}{cccccc}
 & z_{1,2} & z_{1,3} & \cdots & z_{m-2,m} & z_{m-1,m} \\
S_1 & 0 & 0 & \cdots & * & * \\
S_2 & 1 & * & \cdots & * & * \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
S_{m-1} & * & * & \cdots & * & 0 \\
S_m & * & * & \cdots & 1 & 1
\end{array}
$$

Rephrased

$$\mathsf{SINK} : \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$

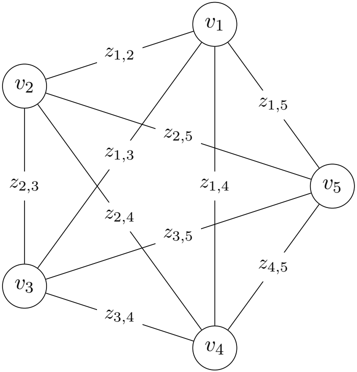
Rephrased

$$\mathsf{SINK} : \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



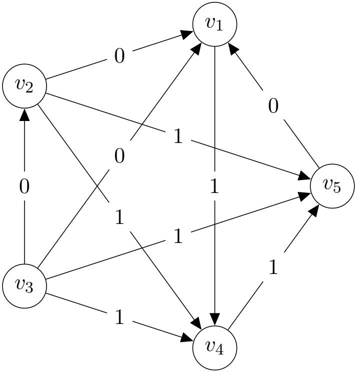
Rephrased

$$\mathsf{SINK} : \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



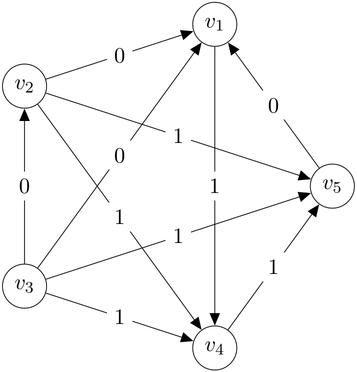
Rephrased

$$\mathsf{SINK} : \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



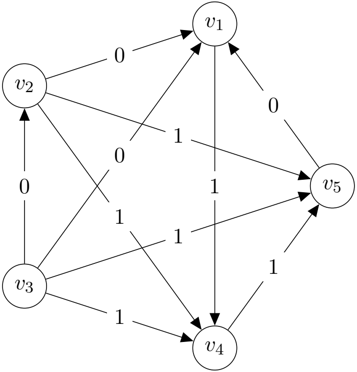
Rephrased

$$\mathsf{SINK} : \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



$\mathsf{SINK}(z) = 1$ iff there is a sink in the graph G_z.

Rephrased

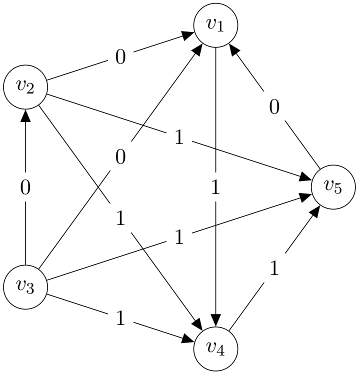


$\mathsf{SINK}(z) = 1$ iff there is a sink in the graph G_z.

- $\left|\left|\widehat{\mathsf{SINK}}\right|\right|_1 \leq m$.
- $\mathsf{sp}_{1/3}(\mathsf{SINK}) \leq m^4$.

Rephrased

$$F := \mathsf{SINK} \circ \mathsf{XOR} : \{0,1\}^{\binom{m}{2}} \times \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



$\mathsf{SINK}(z) = 1$ iff there is a sink in the graph G_z.

- $\left|\left|\widehat{\mathsf{SINK}}\right|\right|_1 \le m$.
- $\mathsf{sp}_{1/3}(\mathsf{SINK}) \le m^4$.
- $\left|\left|\widehat{F}\right|\right|_1 \le m$.
- $\mathsf{rank}_{1/3}(F) \le m^4$.

Alice

$x \in \{0,1\}^{\binom{m}{2}}$

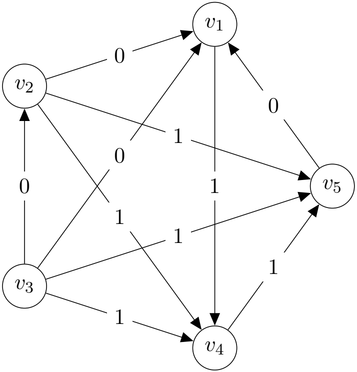
$z = x \oplus y$

Bob

$y \in \{0,1\}^{\binom{m}{2}}$

Rephrased

$$F := \mathsf{SINK} \circ \mathsf{XOR} : \{0,1\}^{\binom{m}{2}} \times \{0,1\}^{\binom{m}{2}} \to \{0,1\}$$



Alice

$x \in \{0,1\}^{\binom{m}{2}}$

$z = x \oplus y$

Bob

$y \in \{0,1\}^{\binom{m}{2}}$

$\mathsf{SINK}(z) = 1$ iff there is a sink in the graph G_z.

- $\left|\left|\widehat{\mathsf{SINK}}\right|\right|_1 \leq m$.
- $\mathsf{sp}_{1/3}(\mathsf{SINK}) \leq m^4$.
- $\left|\left|\widehat{F}\right|\right|_1 \leq m$.
- $\mathsf{rank}_{1/3}(F) \leq m^4$.

Viewing it as a sum of equalities, $\mathsf{rank}^+_{1/3}(F) \leq m^{O(1)}$.

SINK is not easy

Theorem (Chattopadhyay Mande S '19)

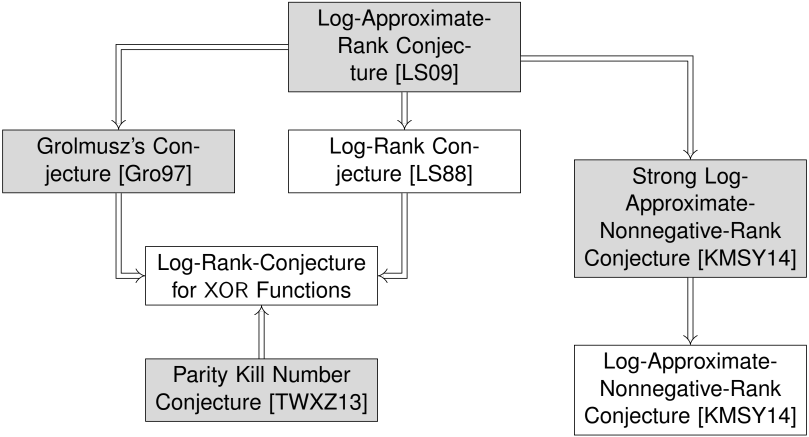
$RPDT(\mathsf{SINK}) \geq \Omega(m)$, $R(\mathsf{SINK} \circ \mathsf{XOR}) \geq \Omega(m)$

SINK is not easy

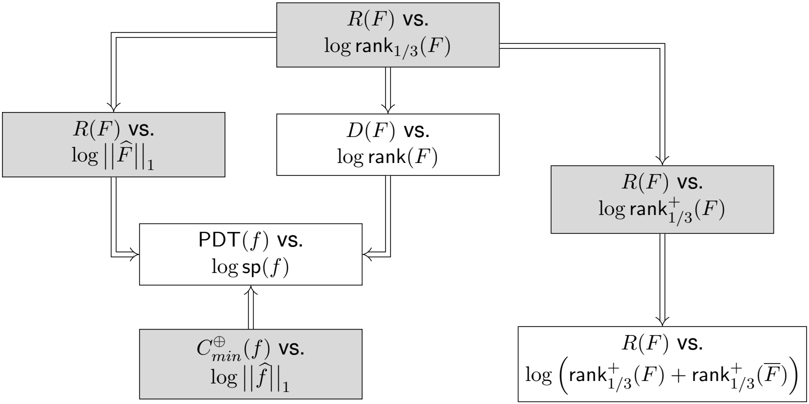
Theorem (Chattopadhyay Mande S '19)

$RPDT(\textsf{SINK}) \geq \Omega(m)$, $R(\textsf{SINK} \circ \textsf{XOR}) \geq \Omega(m)$ *and* SINK *has parity kill number* $\geq \Omega(m)$.

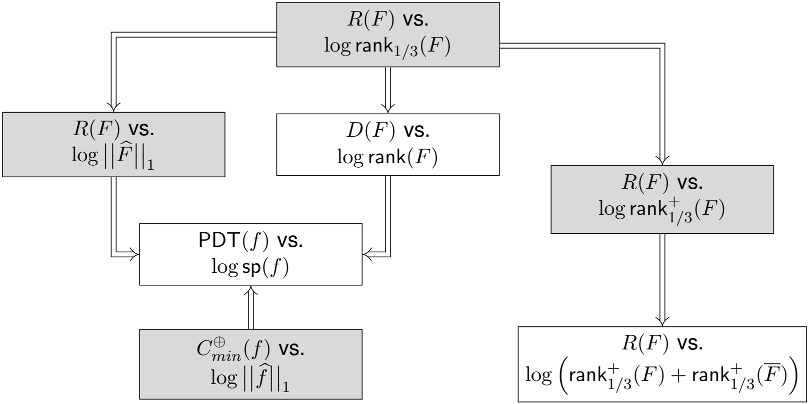
Sunken Conjectures



Sunken Conjectures

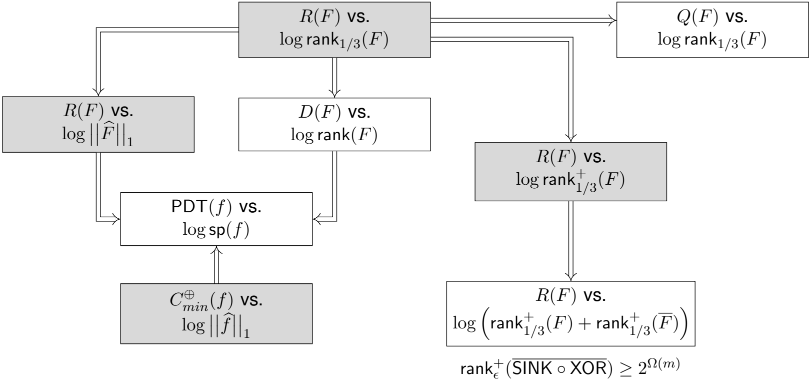


Sunken Conjectures



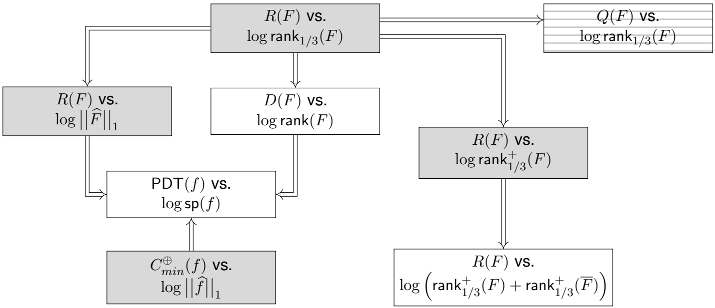
$\mathsf{rank}^+_{\epsilon}(\overline{\mathsf{SINK} \circ \mathsf{XOR}}) \geq 2^{\Omega(m)}$

Sunken Conjectures



Sunken Conjectures

[Anshu Boddu Touchette '18, Sinha & de Wolf '18]



$\mathsf{rank}^+_\epsilon(\overline{\mathsf{SINK} \circ \mathsf{XOR}}) \geq 2^{\Omega(m)}$

About the Log-Approximate-Nonnegative-Rank Conjecture

- Can we have a function f wherein $f^{-1}(1)$ is a disjoint union of subcubes AND $f^{-1}(0)$ is a disjoint union of subcubes BUT f has large RPDT complexity?

About the Log-Approximate-Nonnegative-Rank Conjecture

- Can we have a function f wherein $f^{-1}(1)$ is a disjoint union of subcubes AND $f^{-1}(0)$ is a disjoint union of subcubes BUT f has large RPDT complexity?
  No. Elegant proof follows from [Ehrenfeucht and Haussler '89].

About the Log-Approximate-Nonnegative-Rank Conjecture

- Can we have a function f wherein $f^{-1}(1)$ is a disjoint union of subcubes AND $f^{-1}(0)$ is a disjoint union of subcubes BUT f has large RPDT complexity?
  No. Elegant proof follows from [Ehrenfeucht and Haussler '89].
- The proof does not extend to disjoint unions of affine subspaces. Would be very interesting to settle this possibility.

Summary

- XOR functions behave well.
- PDTs are not well understood.
- Lots of juicy questions:
  - Are Randomized PDTs basically $\wedge$PDTs?
  - Can we close the avenue mentioned towards disproving the Log-Approximate-Nonnegative-Rank Conjecture?
  - Can we better the closeness between randomized complexity and approximate-rank? (SINK is quartically close.)
  - Can we attack the Log-Rank Conjecture? (The summation trick that SINK uses does not work.)
  - And more...

Summary

- XOR functions behave well.
- PDTs are not well understood.
- Lots of juicy questions:
  - Are Randomized PDTs basically $\wedge$PDTs?
  - Can we close the avenue mentioned towards disproving the Log-Approximate-Nonnegative-Rank Conjecture?
  - Can we better the closeness between randomized complexity and approximate-rank? (SINK is quartically close.)
  - Can we attack the Log-Rank Conjecture? (The summation trick that SINK uses does not work.)
  - And more...

Thank you all for attending. I am open to questions and discussions.

Vince Grolmusz.
On the power of circuits with gates of low L_1 norms.
Theor. Comput. Sci., 188(1-2):117–128, 1997.

Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff.
Approximate nonnegative rank is equivalent to the smooth rectangle bound.
In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 701–712, 2014.

László Lovász and Michael E. Saks.
Lattices, möbius functions and communication complexity.
In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 81–90, 1988.

Troy Lee and Adi Shraibman.
Lower bounds in communication complexity.

Foundations and Trends in Theoretical Computer Science, 3(4):263–398, 2009.

Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang.
Fourier sparsity, spectral norm, and the log-rank conjecture.

In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 658–667, 2013.