Compressed Oracle as a Quantum Lazy Sampling Technique

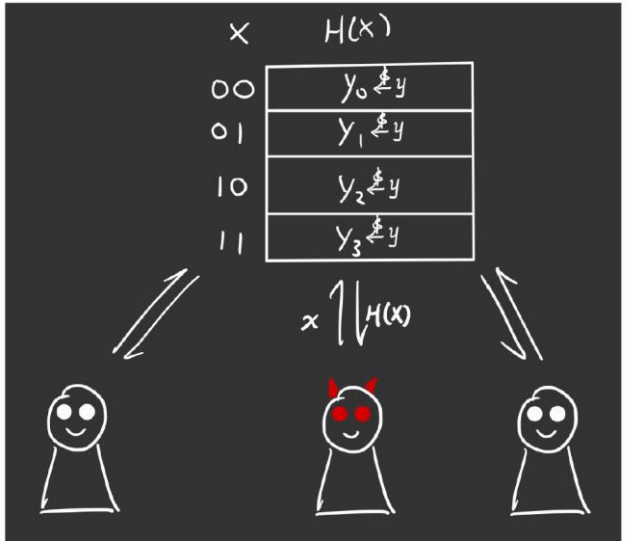
Kai-Min Chung
Academia Sinica, Taiwan

QUIQUES Workshop @ EUROCRYPT ‘21

Random Oracle Model:

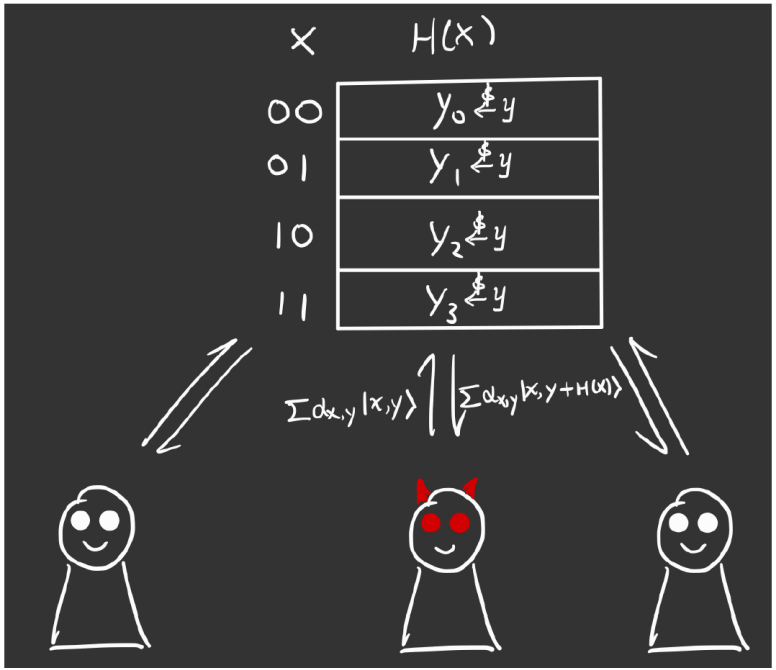
A way to analyze classical cryptographic schemes that use a hash function

$$\text{hash functions} \overset{\text{idealized}}{\approx} \begin{cases} \text{uniform sampled function,} \\ \text{everyone has access} \end{cases}$$



Quantum Random Oracle Model

$$\text{hash functions} \overset{\text{idealized}}{\approx} \begin{cases} \text{uniform sampled function,} \\ \text{everyone has quantum access} \end{cases}$$



Quantum Random Oracle Model

$$\text{hash functions} \overset{\text{idealized}}{\approx} \begin{cases} \text{uniform sampled function,} \\ \text{everyone has quantum access} \end{cases}$$

- Capture security against quantum adversaries
- However, much more challenging for proving security
- Lots of progress in proving QROM security in the past few years
  - O2H lemma, compressed oracle, quantum rewinding, etc…

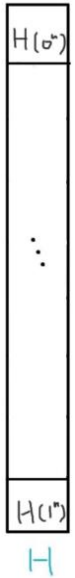
Compressed Oracle Technique [Z19]

- Powerful tool to reason about QROM security with many applications proposed by Mark Zhandry in 2018

- This talk: compressed oracle as a quantum lazy sampling technique
  - Best way to understand the compressed oracle technique
  - Mimic classical lazy sampling technique closely
  - [CFHL21]: A framework to lift classical lazy sampling proofs
    - prove (parallel-query) QROM security by purely classical reasoning
- Example problems:
  - 0-preimage finding: find x s.t. H(x) = 0
  - Collision finding: find x, x’ s.t. H(x) = H(x’)

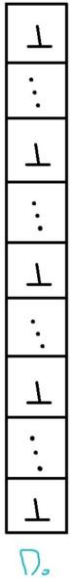
Classical Lazy Sampling

- Sample the random oracle $H: \{0,1\}^n \to \{0,1\}^m$ on the fly



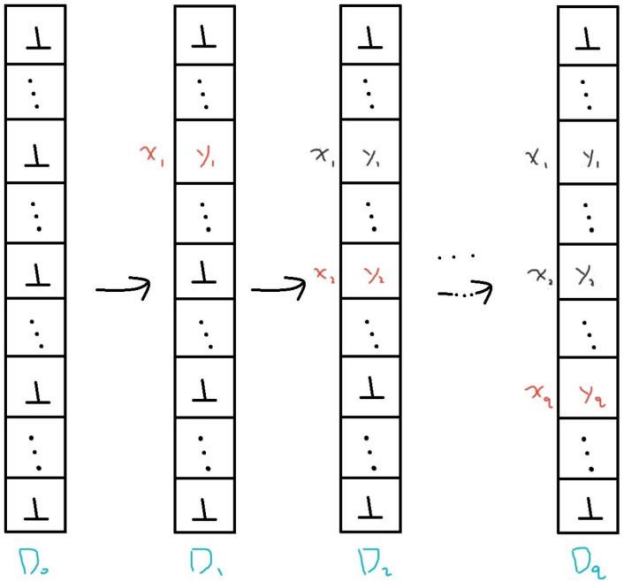
Classical Lazy Sampling

- Sample the random oracle $H: \{0,1\}^n \to \{0,1\}^m$ on the fly
  - Initialize as an all-$\perp$ oracle D_0



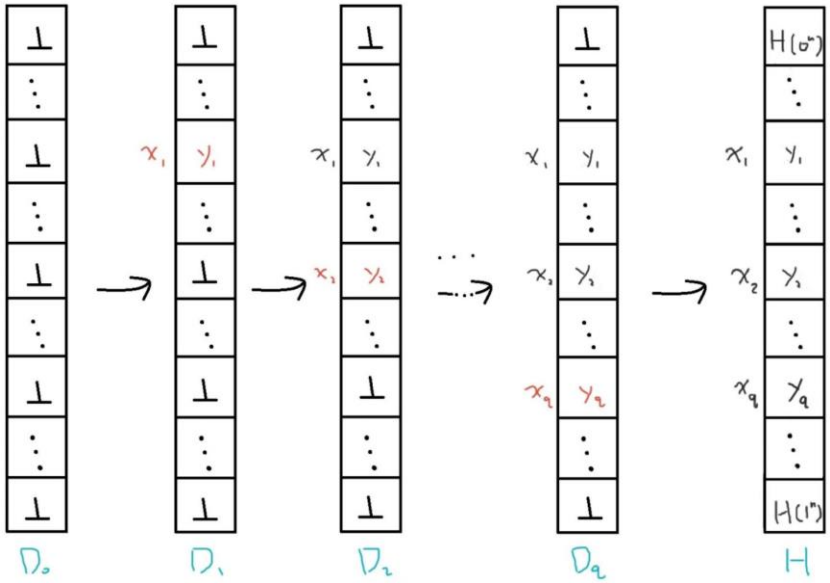
Classical Lazy Sampling

- Sample the random oracle $H: \{0,1\}^n \to \{0,1\}^m$ on the fly
  - Initialize as an all-⊥ oracle D_0
  - On query x: sample $D_i(x)$ uniformly if $D_{i-1}(x) = \perp$
- D_q: at most q non-⊥ entry; record Alg's info about H (“database”)



Classical Lazy Sampling

- Sample the random oracle $H: \{0,1\}^n \to \{0,1\}^m$ on the fly
  - Initialize as an all-⊥ oracle D_0
  - On query x: sample $D_i(x)$ uniformly if $D_{i-1}(x) = \perp$
- D_q: at most q non-⊥ entry; record Alg's info about H (“database”)
- Equivalence: back to H by sampling all ⊥ entry in D_q



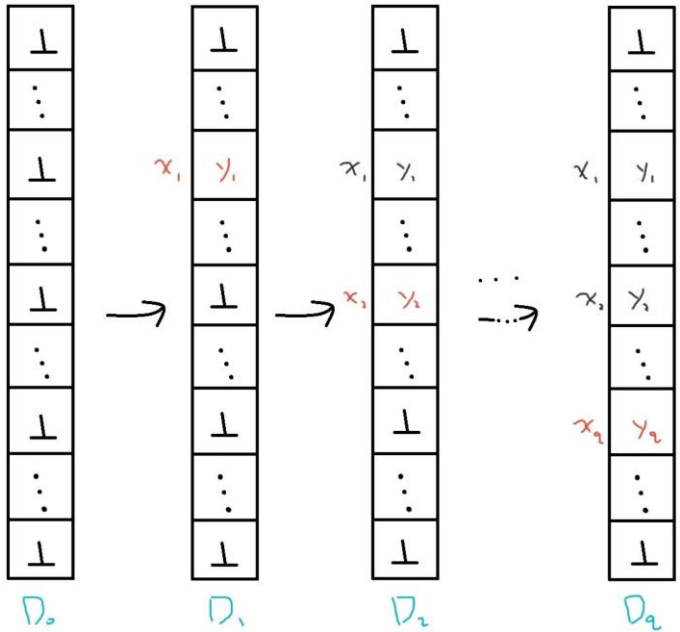
Classical Lazy Sampling for 0-preimage Finding

Let A = q query algorithm outputs x. Say A win if H(x) = 0.

Show Pr[A win] $\leq$ (q+1)/M by lazy sampling:

- Pr[D_q contain 0] $\leq$ q/M

Notation: $N := 2^n, M := 2^m$

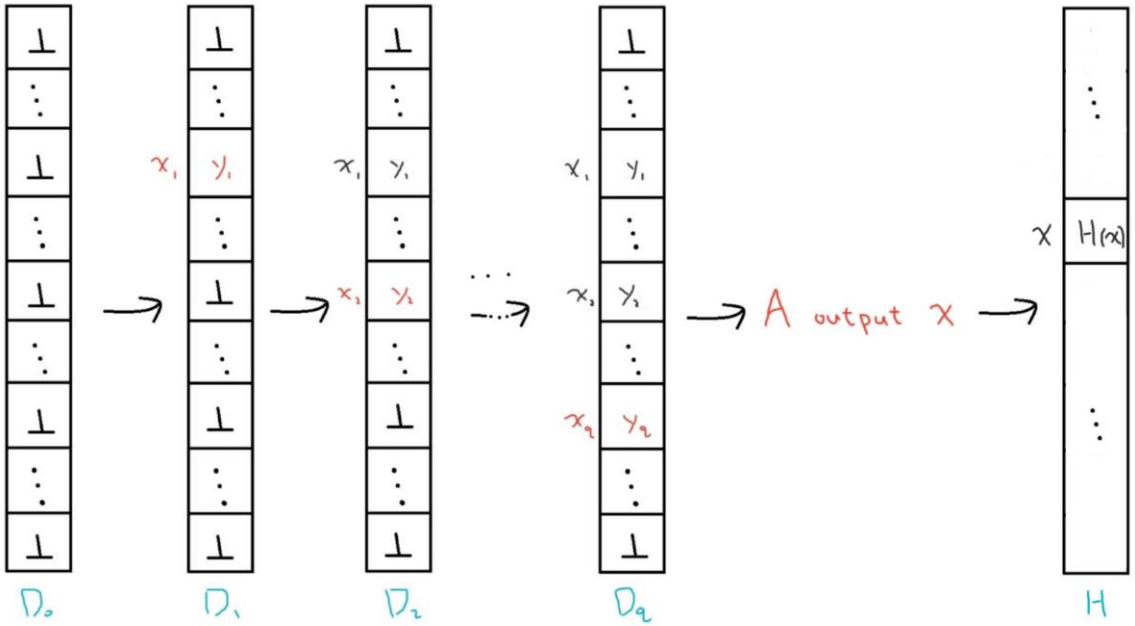


Classical Lazy Sampling for 0-preimage Finding

Let A = q query algorithm outputs x. Say A win if H(x) = 0.

Show Pr[A win] $\leq$ (q+1)/M by lazy sampling:

- Pr[D_q contain 0] $\leq$ q/M
- Pr[A win] $\leq$ Pr[D_q contain 0] + 1/M



Quantum Lazy Sampling

Mimic classical lazy sampling:

1. Initially, initialize an empty (all-$\perp$) “database”
2. Record A’s q query in the “database” in some way
3. Upper bound Pr[“database” has certain property]
4. Show Pr[A win] $\leq$ Pr[“database” has certain property] + something small

By several (equivalent) transformations to view quantum A’s behavior in right way:

- Superposition oracle
- Looking at the Fourier basis
- Augment an extra $\perp$ dimension and “compress” the oracle to database

Quantum Oracle Algorithm

A quantum algorithm A interacting with random oracle H consists of:

- Query register: $\mathbf{X}$
- Answer register: $\mathbf{Y}$
- Working space: $\mathbf{W}$
- Oracle register: $\mathbf{O}$

On the oracle register, the state is a random chosen function $|H\rangle$.

A can apply the query operator O (acting on $\mathbf{X}, \mathbf{Y}$ and $\mathbf{O}$) defined as

$$O: |x, y, H\rangle \mapsto |x, y \oplus H(x), H\rangle.$$

A q-query quantum algorithm A is specified by $q+1$ unitary operators $U_0, \ldots, U_{q+1}$.

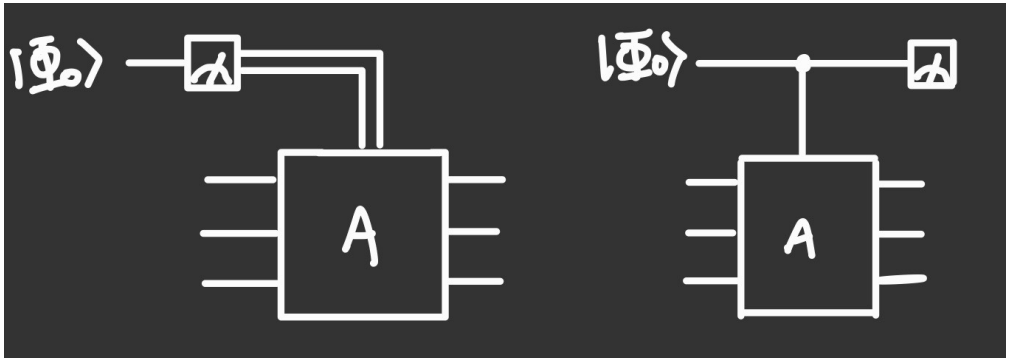
The final state after the execution is $U_{q+1} O U_q \ldots U_1 O U_0 |0\rangle_{\mathbf{XYW}} |H\rangle_{\mathbf{O}}$.

Superposition Oracle

In QROM, $H: \{0,1\}^n \to \{0,1\}^m$ is sampled uniformly at random in the beginning.

Equivalently, we can initialize a superposition oracle state $|\Phi_0\rangle = \frac{1}{\sqrt{M^N}} \sum_H |H\rangle$,and measure $|\Phi_0\rangle$ in the computational basis to obtain a random $|H\rangle$.

Equivalently, since A's operators commute with the measurement on $|\Phi_0\rangle$, we can run A with the superposition oracle state $|\Phi_0\rangle$ and defer the measurement to the end.



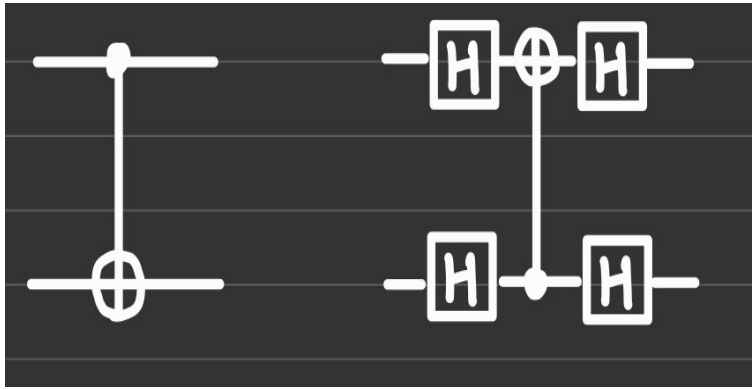
Looking at the Fourier basis

Let $\{|y\rangle\}_{y\in\{0,1\}^m}$ be the computational basis of $\mathbb{C}[\{0,1\}^m]$.

Its Fourier basis $\{|\hat{y}\rangle\}_{y\in\{0,1\}^m}$ is defined by $|\hat{y}\rangle = H^{\otimes m}|y\rangle$ (H: Hadamard matrix)

Two crucial properties:

- Uniform superposition $\frac{1}{\sqrt{M}}\sum_{y\in\{0,1\}^m}|y\rangle$ becomes $|\hat{0}\rangle$ in Fourier basis.
- The operator $|y\rangle|y'\rangle \mapsto |y \oplus y'\rangle|y'\rangle$ in computational basis becomes $|\hat{y}\rangle|\hat{y}'\rangle \mapsto |\hat{y}\rangle|\hat{y}' \oplus \hat{y}\rangle$ in Fourier basis.



In the Fourier basis, the query operator becomes

$$O: \left|x, \hat{y}, \hat{H}\right\rangle \mapsto |x, \hat{y}, \hat{H}(x) \oplus \hat{y}\rangle$$

Since the oracle starts with all $\hat{0}$'s, after q queries the oracle state contains at most q non-$\hat{0}$ entries.

“Compress” the Oracle: Switch between $|\hat{0}\rangle$ and $|\perp\rangle$

Extend $\mathbb{C}[\{0,1\}^m]$ by an extra dimension: The range becomes $\{0,1\}^m \cup \{\perp\}$ and the vector space becomes $\mathbb{C}[\{0,1\}^m \cup \{\perp\}] = \mathbb{C}[\{0,1\}^m] \oplus \mathbb{C}[\{\perp\}]$.

- By definition, $|\perp\rangle$ is its own Fourier basis, i.e., $|\widehat{\perp}\rangle = |\perp\rangle$.

Further extend the oracle register $\mathbf{O}$ entry-wise, and denote it by $\mathbf{cO}$.

- $\mathbf{cO}$ consists of all $|D\rangle$ with “database” $D: \{0,1\}^n \to \{0,1\}^m \cup \{\perp\}$ as basis states.

Define the “compression” operator $Comp_x$ acting on the x entry of $\mathbf{cO}$

$$Comp_x = \sum_{z \neq \hat{0}} |\hat{z}\rangle\langle\hat{z}| + |\perp\rangle\langle\hat{0}| + |\hat{0}\rangle\langle\perp|.$$

Define $Comp = \otimes_x Comp_x$. By definition, $Comp|\Phi_0\rangle = \otimes_x |\perp\rangle$.

- Given an oracle state $|\Phi\rangle$ in $\mathbf{O}$, we call $Comp|\Phi\rangle$ the “compressed oracle” of $|\Phi\rangle$.

Query Operator for Compressed Oracle

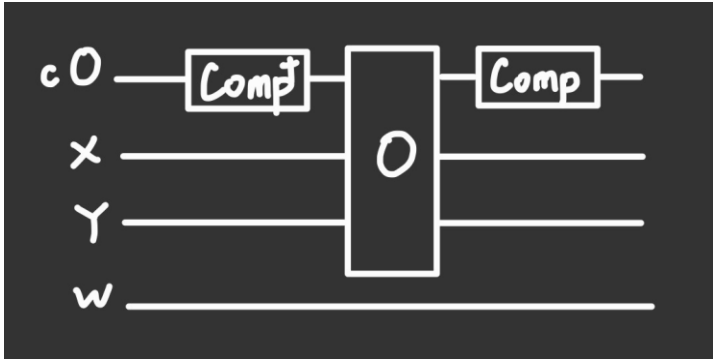
Recall that quantum query operator is defined as

$$O: |x, y, H\rangle \mapsto |x, y \oplus H(x), H\rangle$$

Define corresponding query operator cO for compressed oracle by

$$cO \coloneqq Comp \circ O \circ Comp^{\dagger}$$

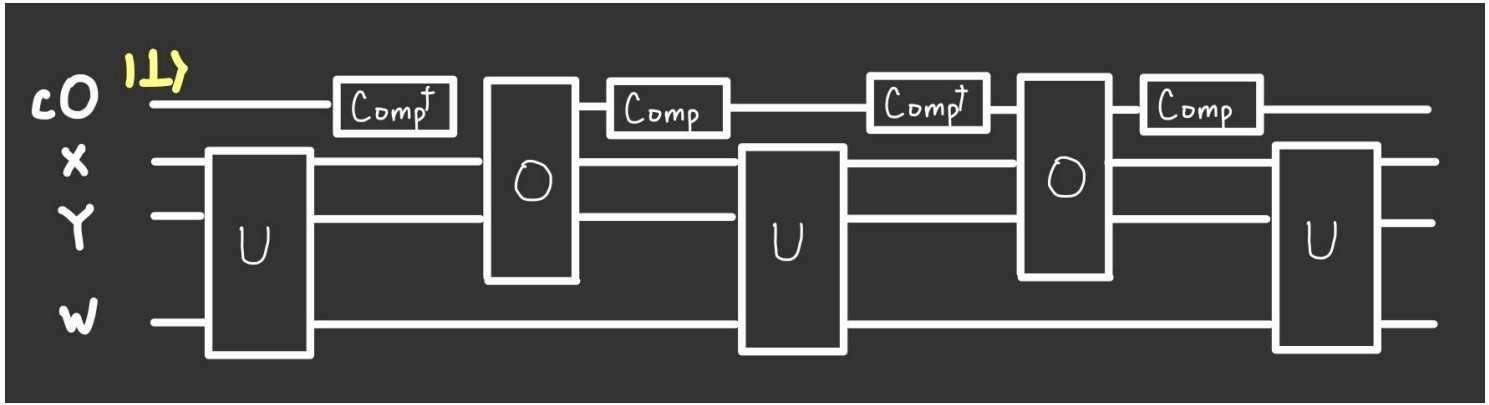
(namely, decompress, query, then compress back)



Compressed Oracle View

For a q-query quantum algorithm A specified by $U_0, \ldots, U_{q+1}$, define the execution "A interacts with compressed oracle D" or "run A in the compressed oracle view":

1. The initial state on the oracle register $\mathbf{cO}$ is $Comp|\Phi_0\rangle = \otimes_x |\perp\rangle$.
2. Every query operator O is replaced with $cO \coloneqq Comp \circ O \circ Comp^\dagger$.

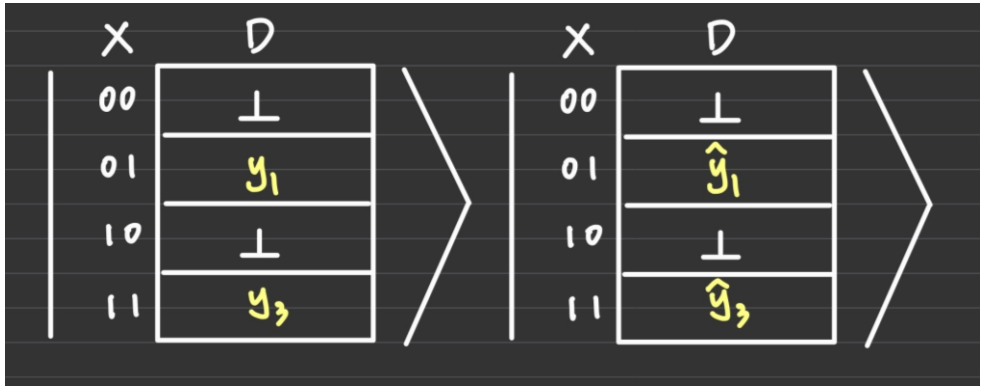


Note that by applying $Comp^\dagger$ (decompression) at the end, we get back the execution "A interacts with random oracle H" or "run A in the standard oracle view"

Size of the Compressed Oracle in Computational Basis

Recall that in the Fourier basis (in the standard oracle view), after q queries the oracle state contains at most q non-$\hat{0}$ entries.

In the compressed oracle view, after q queries the compressed oracle state contains at most q non-$\perp$ entries, in *both* Fourier and computational basis.



Understanding the Compressed Query Operator $cO_{x\hat{y}}$

Recall in the Fourier basis, standard query operator O maps $|x\rangle|\hat{y}\rangle \otimes |\widehat{H}\rangle$ to

$$|x\rangle|\hat{y}\rangle \otimes O_{x\hat{y}}\left|\widehat{H}\right\rangle \coloneqq |x\rangle|\hat{y}\rangle \otimes \left|\widehat{H}(x) \oplus \hat{y}\right\rangle.$$

where $O_{x\hat{y}}$ acts on the x entry of the standard oracle register only.

Recall $cO \coloneqq Comp \circ O \circ Comp^{\dagger}$, so cO maps $|x\rangle|\hat{y}\rangle \otimes |D\rangle$ to $|x\rangle|\hat{y}\rangle \otimes cO_{x\hat{y}}|D\rangle$ as well, where

$$cO_{x\hat{y}} \coloneqq Comp_x \circ O_{x\hat{y}} \circ Comp_x^{\dagger}$$

acting on the x entry of the compressed oracle register only.

Next, let's work out the matrix of $cO_{x\hat{y}}$ in computational basis.

Working Out $cO_{x\hat{y}}$ in Computational Basis

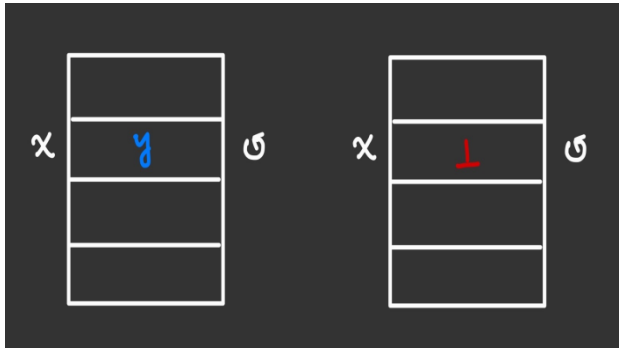
For x and $\hat{y} \neq \hat{0}$, (a, b)-entry in the table represent $|\langle a|cO_{x\hat{y}}|b\rangle|$ in computational basis.

$a \backslash b$	$\perp$	0^m	$0..01$	...	1^m
$\perp$	0	$1/\sqrt{M}$	$1/\sqrt{M}$	...	$1/\sqrt{M}$
0^m	$1/\sqrt{M}$	$1 - O\left(\frac{1}{M}\right)$	$O(1/M)$	...	$O(1/M)$
$0..01$	$1/\sqrt{M}$	$O(1/M)$	$1 - O\left(\frac{1}{M}\right)$	...	$O(1/M)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1^m	$1/\sqrt{M}$	$O(1/M)$	$O(1/M)$	...	$1 - O\left(\frac{1}{M}\right)$

Compare with Prob. Transition Matrix of Classical Query

In classical lazy sampling, the change to $\mathrm{D}(x)$ by a query can be described by a transition matrix:

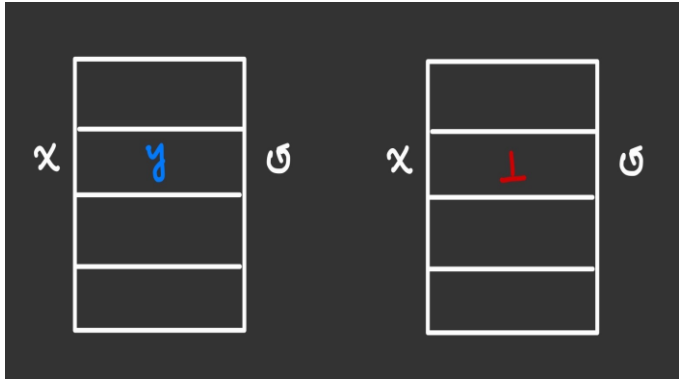
- If the entry is sampled before, i.e., it is non-$\perp$, then it remains the same.
- Otherwise, it is sampled uniformly at random from $\{0,1\}^m$.



	$\perp$	0^m	$0..01$	...	1^m
$\perp$	0	0	0	...	0
0^m	$1/M$	1	0	...	0
$0..01$	$1/M$	0	1	...	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1^m	$1/M$	0	0	...	1

$cO_{x\hat{y}}$ = "resample $\mathrm{D}(x)$"

- If $\mathrm{D}(x) \neq \perp$, then it remains the same "with high prob," and "resampled with small prob."
- Otherwise, it is "sampled" uniformly at random from $\{0,1\}^m$



a \ b	$\perp$	0^m	$0..01$	...	1^m
$\perp$	0	$1/\sqrt{M}$	$1/\sqrt{M}$	...	$1/\sqrt{M}$
0^m	$1/\sqrt{M}$	$1-O(\frac{1}{M})$	$O(1/M)$	...	$O(1/M)$
$0..01$	$1/\sqrt{M}$	$O(1/M)$	$1-O(\frac{1}{M})$	...	$O(1/M)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1^m	$1/\sqrt{M}$	$O(1/M)$	$O(1/M)$	...	$1-O(\frac{1}{M})$

Quantum Lazy Sampling, so far…

Mimic classical lazy sampling:

1. Initially, initialize an empty (all-$\perp$) “database”
2. Record A’s q query in the “database” in some way
3. Upper bound Pr[“database” has certain property]
4. Show Pr[A win] $\leq$ Pr[“database” has certain property] + something small

Let’s try to apply it to 0-preimage finding.

0-Premage Finding Algorithm in Compressed Oracle View

Let A = q query quantum algorithm outputs x. A win if H(x) = 0.

In the compressed oracle view, the final state is of the form

$$|\psi\rangle = U_{q+1}\, cO\, U_q \ldots U_1\, cO\, U_0 |0\rangle_{\mathbf{XYW}} |\perp\rangle_{\mathbf{cO}},$$

where the database state has at most q non-⊥ entries.

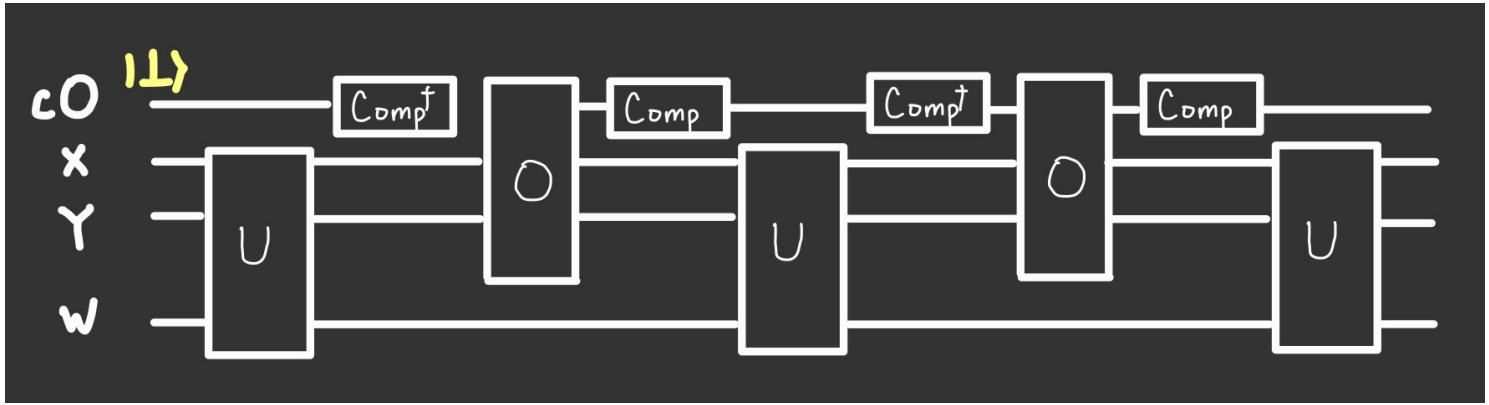
To capture the output of A, we relabel registers into the output register $\mathbf{X}'$ and working space $\mathbf{W}'$. The output is obtained my measuring register $\mathbf{X}'$.

The final state can be written as

$$|\psi\rangle = \sum_{x,w} \alpha_{x,w} |x\rangle_{\mathbf{X}'} |w\rangle_{\mathbf{W}'} |\Gamma_{xw}\rangle_{\mathbf{cO}}.$$

0-Premage Finding Algorithm in Standard Oracle View

Recall that by applying $Comp^\dagger$ we can switch back to the standard oracle view.



The final state in the standard oracle view is

$$|\phi\rangle = Comp^\dagger|\psi\rangle = \sum_{x,w} \alpha_{x,w}|x\rangle_{\mathbf{X}}|w\rangle_{\mathbf{W}}\, Comp^\dagger|\Gamma_{xw}\rangle_{\mathbf{cO}}.$$

Compare Winning Probability in the Two Views

We say "A wins in H" if A output x s.t. $H(x) = 0$ when interacting with standard oracle, "A wins in D" if A output x s.t. $D(x) = 0$ when interacting with compressed oracle.

The winning probabilities can be expressed by

$$\Pr[A \text{ wins in } H] = \sum_{x,w} |\alpha_{x,w}|^2 \left\| P_{x,0} Comp^{\dagger} |\Gamma_{xw}\rangle \right\|_2^2$$

and

$$\Pr[A \text{ wins in } D] = \sum_{x,w} |\alpha_{x,w}|^2 \left\| P_{x,0} |\Gamma_{xw}\rangle \right\|_2^2$$

where $P_{x,0}$ is the projector that check if the x entry of the oracle register = 0.

Bounding the Difference in Probability

$$\Pr[A \text{ wins in } H] - \Pr[A \text{ wins in } D]$$

$$= \sum_{x,w} |\alpha_{x,w}|^2 \cdot \left(\left\|P_{x,0} Comp^\dagger |\Gamma_{xw}\rangle\right\|_2^2 - \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|_2^2\right)$$

$$\leq \max_{x,w} \left(\left\|P_{x,0} Comp^\dagger |\Gamma_{xw}\rangle\right\|_2^2 - \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|_2^2\right)$$

$$\leq \max_{x,w} \left(\left\|P_{x,0} Comp^\dagger |\Gamma_{xw}\rangle\right\| + \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|\right) \cdot \left(\left\|P_{x,0} Comp^\dagger |\Gamma_{xw}\rangle\right\| - \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|\right)$$

$$\leq 2 \max_{x,w} \left(\left\|P_{x,0} Comp^\dagger |\Gamma_{xw}\rangle\right\| - \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|\right)$$

$$\leq 2 \max_{x,w} \left(\left\|P_{x,0} Comp_x^\dagger |\Gamma_{xw}\rangle\right\| - \left\|P_{x,0} |\Gamma_{xw}\rangle\right\|\right)$$

$$\leq 2 \max_{x,w} \left(\left\|P_{x,0} (I - Comp_x^\dagger) |\Gamma_{xw}\rangle\right\|\right)$$

$$\leq 2 \left\|P_{x,0} (I - Comp_x^\dagger)\right\|_{op}.$$

Bounding $\left\| P_{x,0}(I - Comp_x) \right\|_{op}$

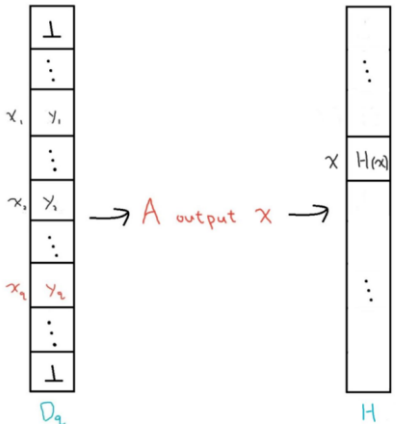
Capture the “error” due to compression

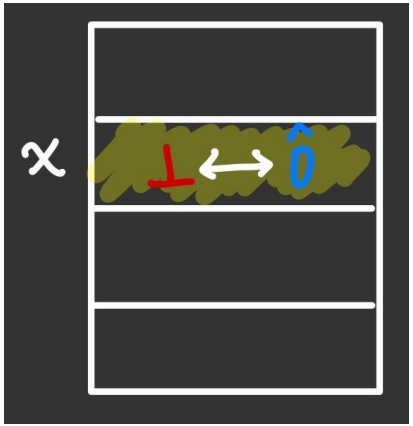
Can be bound by a direct calculation using Frobenius norm,

$$\left\| P_{x,0}(I - Comp_x) \right\|_{op} \leq \left\| P_{x,0}(I - Comp_x) \right\|_F \leq \frac{1}{\sqrt{M}}$$

which implies

$$\Pr[A \text{ wins in } H] \leq \Pr[A \text{ wins in } D] + \frac{2}{\sqrt{M}}$$





In General…

Lemma 1: Let $R \subseteq X^\ell \times Y^\ell$ be a relation specifying the "winning condition".
Let A be an oracle quantum algorithm that outputs $(\boldsymbol{x}, \boldsymbol{y}) \in X^\ell \times Y^\ell$.
We say "A wins in H" if $(\boldsymbol{x}, \boldsymbol{y}) \in R$ and $\boldsymbol{y} = H(\boldsymbol{x})$ in standard oracle view, and "A wins in D" if $(\boldsymbol{x}, \boldsymbol{y}) \in R$ and $\boldsymbol{y} = \mathrm{D}(\boldsymbol{x})$ in compressed oracle view. Then,

$$\Pr[A \text{ wins in } H] \leq \Pr[A \text{ wins in } D] + 2\sqrt{\ell/M}.$$

Therefore, we can bound $\Pr[A \text{ wins in } H]$ by bounding

$$\Pr[A \text{ wins in } D] \leq \Pr[D \in P^R],$$

where database property $P^R := \{D | \exists \boldsymbol{x} \text{ s.t. } \big(\boldsymbol{x}, D(\boldsymbol{x})\big) \in R\}$.

Quantum Lazy Sampling, so far…

Mimic classical lazy sampling:

1. Initially, initialize an empty (all-⊥) “database”
2. Record A’s q query in the “database” in some way
3. Upper bound Pr[“database” has certain property]
4. Show Pr[A win] ≤ Pr[“database” has certain property] + something small

In 0-preimage finding, it remains to bound $\Pr[D \text{ contain } 0]$.

Back to the Classical Case

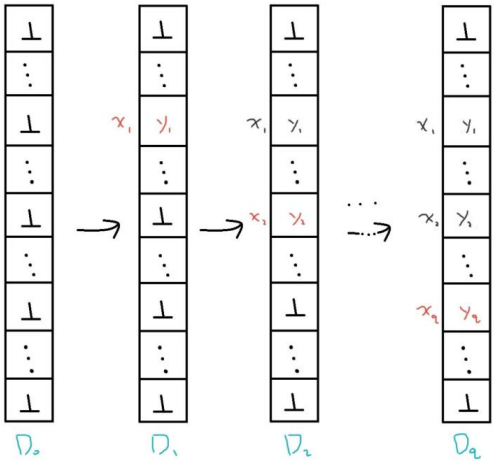
Define database property: $ZR := \{D: \{0,1\}^n \to \{0,1\}^m \cup \{\perp\} \mid \exists \bar{x}: D(\bar{x}) = 0\}$

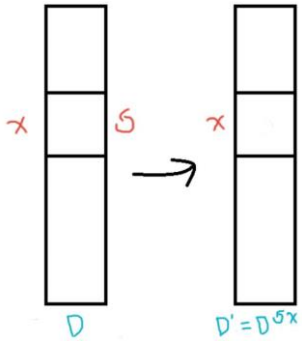
Bound Pr[D_q contain 0] = Pr[$D_q \in ZR$] step by step:

$$\Pr[D_q \in ZR] \leq \sum_{i=1}^{q} \Pr[D_i \in ZR \mid D_{i-1} \notin ZR]$$

$$\leq q \cdot \max_{\substack{D \in \neg ZR \\ x \in \{0,1\}^n}} \Pr[D^{\circlearrowleft x} \in ZR]$$

where $D^{\circlearrowleft x}$ = database after query x on D





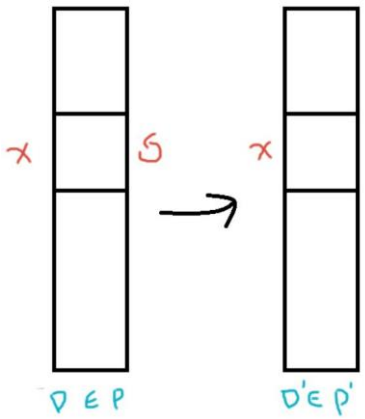
Define Classical Transition Capacity

Let P, P' be database properties. Define *classical transition capacity* from P to P'

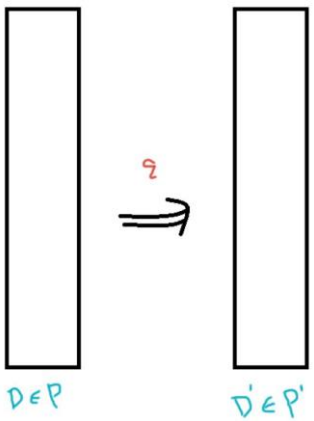
$$[P \to P'] := \max_{\substack{D \in P \\ x \in \{0,1\}^n}} \Pr[D^{\circlearrowleft x} \in P']$$

$$\left[P \overset{q}{\Rightarrow} P'\right] := \max_{\substack{D \in P \\ D' = D \text{ after } q \text{ query}}} \Pr[D' \in P']$$

$[P \to P']$



$\left[P \overset{q}{\Rightarrow} P'\right]$



Define Classical Transition Capacity

Let P, P' be database properties. Define *classical transition capacity* from P to P'

$$[P \to P'] := \max_{\substack{D \in P \\ x \in \{0,1\}^n}} \Pr[D^{\circlearrowleft x} \in P']$$

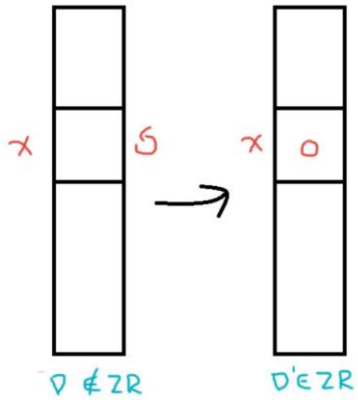
$$\left[P \overset{q}{\Rightarrow} P'\right] := \max_{\substack{D \in P \\ D' = D \text{ after } q \text{ query}}} \Pr[D' \in P']$$

Classical analysis using transition capacity:

$$\Pr[A\ win] \leq \Pr\left[D_q \in ZR\right] + \frac{1}{M}$$

$$\leq \left[\perp \overset{q}{\Rightarrow} ZR\right] + \frac{1}{M}$$

$$\leq q \cdot [\neg ZR \to ZR] + \frac{1}{M} = \frac{q+1}{M}$$



where database property ⊥ = { D: ∀x, D(x) = ⊥}

Define Quantum Transition Capacity

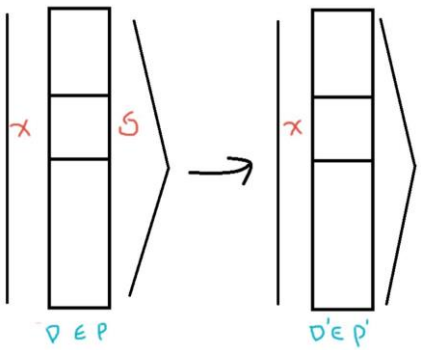
Let P, P' be database properties. Define *quantum transition capacity* from P to P'

$$[\![P \to P']\!] := \max_{x,\hat{y}} \left\| P' c O_{x\hat{y}} P \right\|_{op}$$

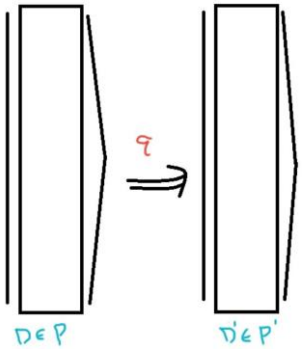
$$\left[\!\left[P \overset{q}{\Rightarrow} P'\right]\!\right] := \sup_{U_0,\ldots,U_q} \left\| P' U_q c O \ldots U_1 \, c O U_0 P \right\|_{op}$$

Operator norm:
$\|A\|_{op} := \max_{|\psi\rangle} \|A|\psi\rangle\|$

$[\![P \to P']\!]$



$\left[\!\left[P \overset{q}{\Rightarrow} P'\right]\!\right]$



Define Quantum Transition Capacity

Let P, P' be database properties. Define *quantum transition capacity* from P to P'

$$[\![P \to P']\!] \coloneqq \max_{x,\hat{y}} \left\| P' c O_{x\hat{y}} P \right\|_{op}$$

$$\left[\!\left[P \overset{q}{\Rightarrow} P'\right]\!\right] \coloneqq \sup_{U_0,\ldots,U_q} \left\| P' U_q c O \ldots U_1 \, c O U_0 P \right\|_{op}$$

Compare with classical transition capacity

- Both are worst-case notions
- Instead of probability, work with norm (square root of probability)

Quantum Analysis Using Quantum Transition Capacity

$$\Pr[A \text{ win in } H] \leq \Pr[A \text{ win in } D] + 2\sqrt{1/M}$$

$$\leq \Pr[D \in ZR] + 2\sqrt{1/M}$$

$$\leq \left[\!\left[\perp \overset{q}{\Rightarrow} ZR \right]\!\right]^2 + 2\sqrt{1/M}$$

$$\leq \left(\sum_{s=1}^{q} [\![\neg ZR \rightarrow ZR]\!]\right)^2 + 2\sqrt{1/M}.$$

Step 1. break up transition capacity

$$\leq O\left(\frac{q^2}{M}\right)$$

Step 2. bound one-step transition capacity

Step 1. Breaking Up Quantum Transition Capacity

Lemma 2: For any sequence of database properties $P_0, P_1, \dots, P_q$,

$$\left[\!\left[\neg P_0 \overset{q}{\Rightarrow} P_q\right]\!\right] \leq \sum_{s=1}^{q} [\![\neg P_{s-1} \to P_s]\!] .$$

Proof: By means of inserting $I = P_q + (I - P_q)$ before U_q and using the fact that U_q and P_q commute since they act on different subsystem, we obtain

$$\left\|P_q U_q cO \dots cOU_0 (I - P_0)\right\|_{op} \leq \left\|P_{q-1} U_q cO \dots cOU_0 \,(I - P_0)\right\|_{op} + \left\|P_q cO\big(I - P_{q-1}\big)\right\|_{op}.$$

To the first term, we break it up by inserting $I = P_i + (I - P_i)$ inductively.

As for the second term, it follows that

$$\left\|P_q cO(I - P_{q-1})\right\|_{op} \leq \max_{x,\hat{y}} \left\|P_q cO_{x\hat{y}}\big(I - P_{q-1}\big)\right\|_{op} = [\![\neg P_{q-1} \to P_q]\!],$$

where inequality holds since $|x\rangle|\hat{y}\rangle \otimes \mathbf{cO}$ is an invariant subspace of $P_q cO(I - P_{q-1})$.

Properties of the operator norm

1. Suppose $\mathcal{H}$ is a vector space and $A: \mathcal{H} \to \mathcal{H}$ is a linear operator, if

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \cdots \oplus \mathcal{H}_m$$

such that A acts invariantly on each subspace, i.e., $A(\mathcal{H}_i) \subseteq \mathcal{H}_i$, then

$$\|A\| = \max_i \left\| A|_{\mathcal{H}_i} \right\|_{op},$$

where $A|_{\mathcal{H}_i}$ is the restriction of A on the subspace $\mathcal{H}_i$.

2. Given a rank-r matrix A, it holds that

$$\frac{1}{\sqrt{r}} \|A\|_F \leq \|A\|_{op} \leq \|A\|_F,$$

where the Frobenius norm $\|A\|_F = \sum_{ij} A_{ij}^2$. It says Frobenius norm is a good approximation for the operator norm of a low-rank matrix.

Step 1. Breaking Up Quantum Transition Capacity

By setting $P_0 = P_1 = \cdots = P_q = ZR$ in Lemma 2, we have

$$\left[\!\left[\bot \overset{q}{\Rightarrow} ZR \right]\!\right] \leq \sum_{s=1}^{q} [\![\neg ZR \rightarrow ZR]\!]$$

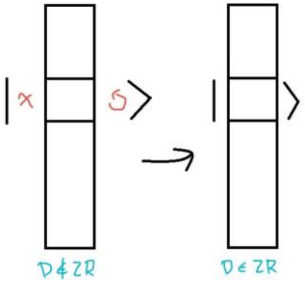
Step 2. Bounding $[\![\neg ZR \to ZR]\!] = \max_{x,\hat{y}} \left\| ZR\ cO_{x\hat{y}}(I - ZR) \right\|_{op}$

Given any $x, \hat{y}$, recall $cO_{x\hat{y}}$ acting as "resampling" x does not perturb other entry.

For any x and D, define D_{-x} as the partial database obtained by removing the x entry.

Note that the space $|D_{-x}\rangle \otimes \mathbf{cO}_x$ is an invariant subspace of $cO_{x\hat{y}}$.

$|D_{-x}\rangle \otimes \mathbf{cO}_x$ is also invariant under projectors ZR and $(I - ZR)$.



By the property of operator norm, it holds that

$$\left\| ZR\ cO_{x\hat{y}}(I - ZR) \right\|_{op} \leq \max_{D_{-x}} \left\| ZR|_{D_{-x}}\ cO_{x\hat{y}}(I - ZR)|_{D_{-x}} \right\|_{op}$$

where $ZR|_{D_{-x}}$ and $(I - ZR)|_{D_{-x}}$ are the projectors restricted in $|D_{-x}\rangle \otimes \mathbf{cO}_x$.

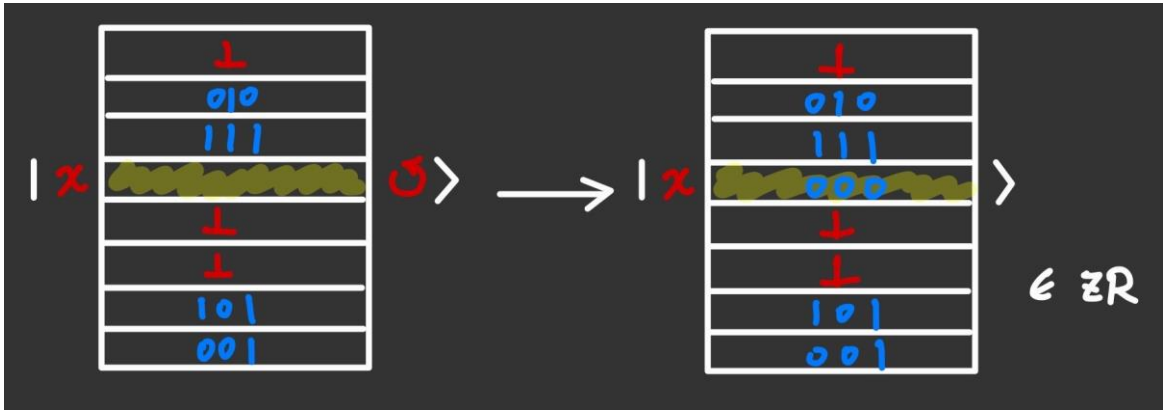
Restriction of Database Property

Let P be a database property and D_{-x} a partial database.

$P|_{D_{-x}} \subseteq \{0,1\}^m$ is the set of value for $D(x)$ to make $D \in P$

For example, for any given x and D,

$$ZR|_{D_{-x}} = \begin{cases} \{0,1\}^m, \text{if } D(\bar{x}) = 0^m \text{ for some } \bar{x} \neq x \\ 0^m \quad\quad, \text{otherwise} \end{cases}$$



Step 2. Bounding $[\![\neg ZR \to ZR]\!]$

Now, we have

over $\mathbf{cO}$

$$[\![\neg ZR \to ZR]\!] = \boxed{\max_{x,\hat{y}} \left\| ZRcO_{x\hat{y}}(I - ZR) \right\|_{op}}$$

$$\leq \boxed{\max_{x,\hat{y},D} \left\| ZR|_{D_{-x}} cO_{x\hat{y}}(I - ZR)|_{D_{-x}} \right\|_{op}}$$

over $\mathbf{cO}_x = \mathbb{C}[\{0,1\}^m] \oplus \mathbb{C}[|\perp\rangle]$

	$\perp$	0^m	$0..01$	...	1^m
$\perp$	0	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{\sqrt{M}})$	...	$O(\frac{1}{\sqrt{M}})$
0^m	$O(\frac{1}{\sqrt{M}})$	$1 - O(\frac{1}{M})$	$O(\frac{1}{M})$	...	$O(\frac{1}{M})$
$0..01$	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{M})$	$1 - O(\frac{1}{M})$	...	$O(\frac{1}{M})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1^m	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{M})$	$O(\frac{1}{M})$	...	$1 - O(\frac{1}{M})$

For any $x, \hat{y}$ and D, bounding by Frobenius norm we have

$$\left\| ZR|_{D_{-x}} cO_{x\hat{y}}(I - ZR)|_{D_{-x}} \right\|_{op} \leq \left\| ZR|_{D_{-x}} cO_{x\hat{y}}(I - ZR)|_{D_{-x}} \right\|_F$$

$$\leq \sum_{\blacksquare} |\blacksquare|^2 \leq \sqrt{10/M}$$

Thus, $[\![\neg ZR \to ZR]\!] \leq O(1/\sqrt{M})$.

Properties of the operator norm

1. Suppose $\mathcal{H}$ is a vector space and $A: \mathcal{H} \to \mathcal{H}$ is a linear operator, if

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \cdots \oplus \mathcal{H}_m$$

such that A acts invariantly on each subspace, i.e., $A(\mathcal{H}_i) \subseteq \mathcal{H}_i$, then

$$\|A\| = \max_i \left\| A|_{\mathcal{H}_i} \right\|_{op},$$

where $A|_{\mathcal{H}_i}$ is the restriction of A on the subspace $\mathcal{H}_i$.

2. Given a rank-r matrix A, it holds that

$$\frac{1}{\sqrt{r}} \|A\|_F \leq \|A\|_{op} \leq \|A\|_F,$$

where the Frobenius norm $\|A\|_F = \sum_{ij} A_{ij}^2$. It says Frobenius norm is a good approximation for the operator norm of a low-rank matrix.

Quantum Analysis Using Quantum Transition Capacity

Let $A = q$ query quantum algorithm outputs x. A win if $H(x) = 0$.

$$\Pr[A \text{ win in } H] \leq \Pr[A \text{ win in } D] + 2\sqrt{1/M}$$

$$\leq \Pr[D \in ZR] + 2\sqrt{1/M}$$

$$\leq \left[\!\left[\perp \overset{q}{\Rightarrow} ZR \right]\!\right]^2 + 2\sqrt{1/M}$$

$$\leq \left(\sum_{s=1}^{q} [\![\neg ZR \rightarrow ZR]\!]\right)^2 + 2\sqrt{1/M}.$$

$$\leq O\left(\frac{q^2}{M}\right)$$

Quantum Lazy Sampling, so far…

Mimic classical lazy sampling:

1. Initially, initialize an empty (all-$\perp$) “database”
2. Record A’s q query in the “database” in some way
3. Upper bound Pr[“database” has certain property]
4. Show Pr[A win] $\leq$ Pr[“database” has certain property] + something small

Let’s apply it to one more example: collision finding

Classical Lazy Sampling for Collision Finding

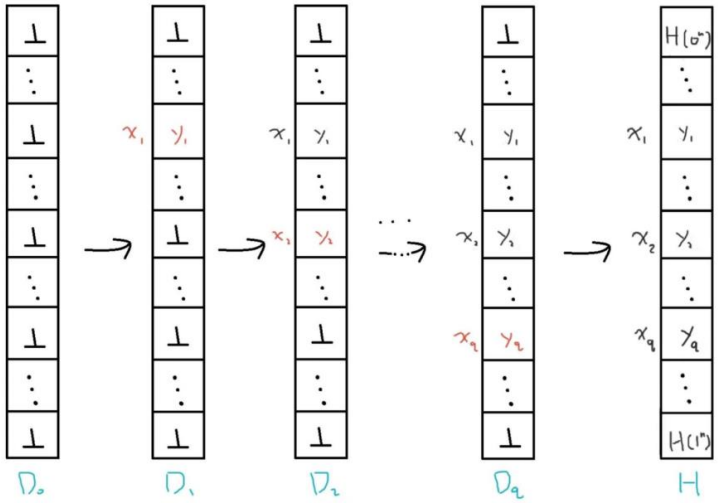
Let $A = q$ query algorithm outputs x, x'. Say A win if $H(x) = H(x')$.

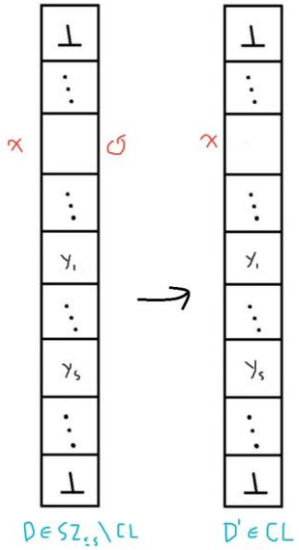
Show $\Pr[\, A\ win\,] \le O(q^2/M)$ by lazy sampling:

- Define $CL := \{D \mid \exists\, x \neq x' \colon D(x) = D(x') \neq \perp\}$
- $\Pr[A\ win] \le \Pr[D_q \in CL] + 1/M$

$SZ_{\le s} := \{D \colon \{0,1\}^n \to \{0,1\}^m \mid |\{x \colon D(x) \neq \perp\}| \le s\}$

- $\Pr[D_q \in CL] \le \left[\perp \overset{q}{\Rightarrow} CL\right] \le \sum_{s=0}^{q-1} [SZ_{\le s} \setminus \mathrm{CL} \to CL] \le \sum_{s=0}^{q-1} s/M = q(q-1)/2M$





Step 1. Breaking Up Quantum Transition Capacity

By setting $P_s = \neg(SZ_{\leq s} \setminus CL) = CL \cup \neg SZ_{\leq s}$ in Lemma 2, we have

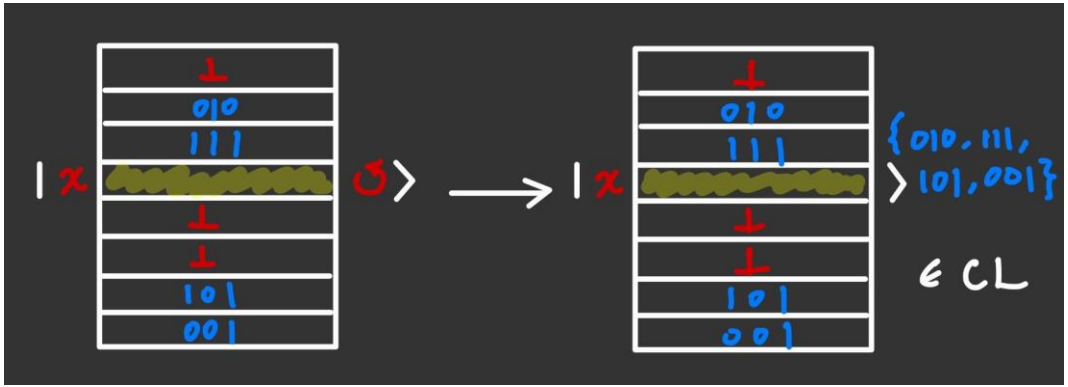
$$\left[\!\left[\bot \overset{q}{\Rightarrow} CL\right]\!\right] \leq \sum_{s=0}^{q-1} [\![SZ_{\leq s} \setminus CL \to CL \cup \neg SZ_{\leq s}]\!] \leq \sum_{s=0}^{q-1} [\![SZ_{\leq s} \setminus CL \to CL]\!].$$

Step 2. Bounding $[\![SZ_{\leq s} \setminus CL \to CL]\!]$

By the same argument, we have

$$[\![SZ_{\leq s} \setminus CL \to CL]\!] = \max_{x,\hat{y}} \left\| CL\, cO_{x\hat{y}}\, (SZ_{\leq s} \setminus CL) \right\|_{op} \leq \max_{x,\hat{y},D \in SZ_{\leq s}} \left\| CL|_{D_{-x}} cO_{x\hat{y}} (I - CL)|_{D_{-x}} \right\|_{op}$$

For any x and $D \in SZ_{\leq s}$, $CL|_{D_{-x}} = \{D(\bar{x}) | \bar{x} \neq x : D(\bar{x}) \neq \perp\} \subseteq \{0,1\}^m$ has size at most s.



Step 2. Bounding $[\![SZ_{\leq s} \setminus CL \to CL]\!]$

$$[\![SZ_{\leq s} \setminus CL \to CL]\!] = \max_{x,\hat{y}} \left\| CLcO_{x\hat{y}}(SZ_{\leq s} \setminus CL) \right\|_{op}$$

$$\leq \max_{x,\hat{y},D \in SZ_{\leq s}} \left\| CL|_{D_{-x}} cO_{x\hat{y}}(I - CL)|_{D_{-x}} \right\|_{op}$$

As before, for any $x,\hat{y}$ and $D \in SZ_{\leq s}$, bounding operator norm by Frobenius norm we have

$$\left\| CL|_{D_{-x}} cO_{x\hat{y}}(I - CL)|_{D_{-x}} \right\|_{op}$$

$$\leq \left\| CL|_{D_{-x}} cO_{x\hat{y}}(I - CL)|_{D_{-x}} \right\|_{F}$$

$$\leq \sum_{\blacksquare} |\blacksquare|^2 \leq \sqrt{10s/M} \leq \sqrt{10q/M}$$

Thus, $[\![SZ_{\leq s} \setminus CL \to CL]\!] = O(\sqrt{q/M})$.

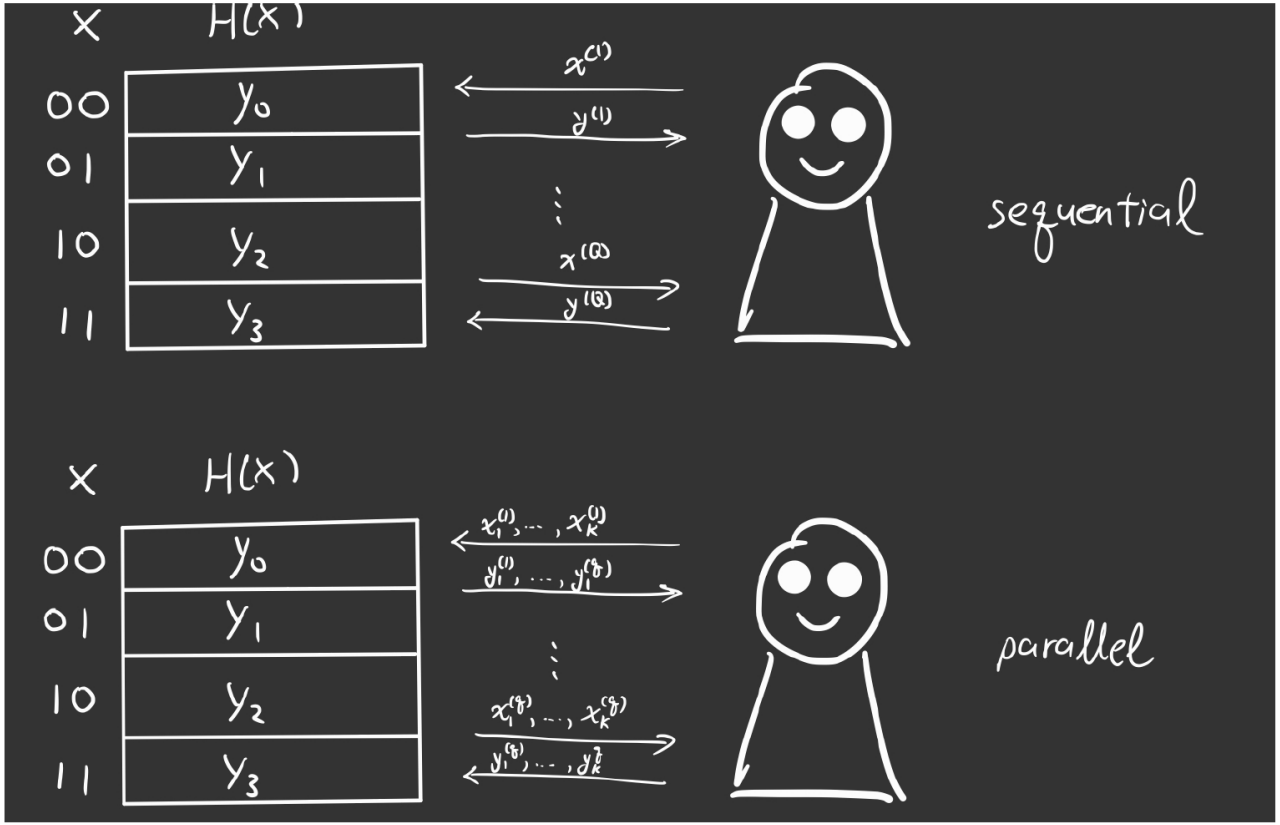
	$\perp$	0^m	$0..01$	...	1^m
$\perp$	0	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{\sqrt{M}})$	...	$O(\frac{1}{\sqrt{M}})$
0^m	$O(\frac{1}{\sqrt{M}})$	$1 - O(\frac{1}{M})$	$O(\frac{1}{M})$	...	$O(\frac{1}{M})$
$0..01$	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{M})$	$1 - O(\frac{1}{M})$	...	$O(\frac{1}{M})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1^m	$O(\frac{1}{\sqrt{M}})$	$O(\frac{1}{M})$	$O(\frac{1}{M})$	...	$1 - O(\frac{1}{M})$

Quantum Lazy Sampling for Collision Finding

Let $A = q$ query quantum algorithm outputs x, x'. A win if $H(x) = H(x')$.

$$\begin{aligned}
\Pr[A \text{ win in } H] &\leq \Pr[A \text{ win in } D] + 2\sqrt{2/M} \\
&\leq \Pr[D \in CL] + 2\sqrt{2/M} \\
&\leq \left[\!\left[\perp \overset{q}{\Rightarrow} ZR \right]\!\right]^2 + 2\sqrt{2/M} \\
&\leq \left(\sum_{s=1}^{q} [\![SZ_{\leq s} \setminus CL \to CL]\!]\right)^2 + 2\sqrt{2/M}. \\
&\leq O\left(\frac{q^3}{M}\right)
\end{aligned}$$

Q(ROM) with Parallel Queries



Security against Parallel Quantum Algorithm

Let A = quantum oracle algorithm making q k-parallel queries.

What is Pr[A win] for 0-preimage finding and Collision Finding?

	Sequential		**k-parallel**	
	Upper bound	Algorithm	Upper bound	Algorithm
0-preimage Finding	$O(q^2/M)$	Grover: $\Omega(q^2/M)$	$O(kq^2/M)$	Parallel Grover: $\Omega(kq^2/M)$
Collision Finding	$O(q^3/M)$	BHT: $\Omega(q^3/M)$	$O(k^2q^3/M)$	“Parallel” BHT: $\Omega(k^2q^3/M)$

Parallel Query Operator for (Compressed) Oracle

For any integer k, a k-parallel query operator is defined as

$$O^k: |\boldsymbol{x}\rangle|\boldsymbol{y}\rangle|H\rangle \rightarrow |\boldsymbol{x}\rangle|\boldsymbol{y} \oplus H(\boldsymbol{x})\rangle|H\rangle$$

for any $\boldsymbol{x} = (x_1, \dots, x_k)$ and $\boldsymbol{y} = (y_1, \dots, y_k)$.

The operator $cO^k = Comp \circ O^k \circ Comp^\dagger$ is defined as

$$cO^k: |\boldsymbol{x}\rangle|\hat{\boldsymbol{y}}\rangle|D\rangle \rightarrow |\boldsymbol{x}\rangle|\hat{\boldsymbol{y}}\rangle cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}|D\rangle,$$

where $cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}$ is the product of $cO_{x_1\hat{y}_1} \dots cO_{x_k\hat{y}_k}$.

Quantum Transition Capacity for Parallel Query

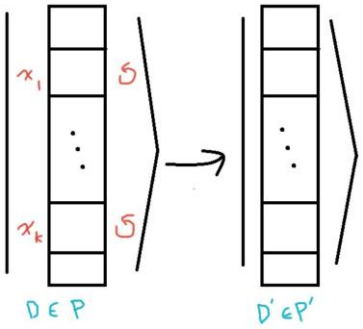
Let P, P' be database properties.

Define *quantum transition capacity* from P to P' for k-parallel query

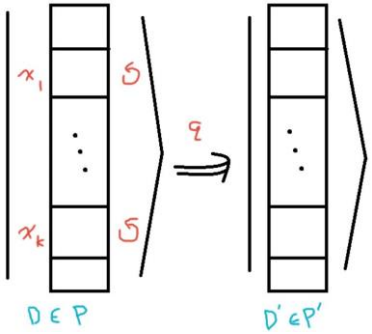
$$[\![P \xrightarrow{k} P']\!] := \max_{\boldsymbol{x},\hat{\boldsymbol{y}}} \left\| P' c O_{\boldsymbol{x}\hat{\boldsymbol{y}}} P \right\|_{op}$$

$$[\![P \overset{k,q}{\Longrightarrow} P']\!] := \sup_{U_0,\ldots,U_q} \left\| P' U_q c O^k \ldots U_1 \, c O^k U_0 P \right\|_{op}$$

$[\![P \xrightarrow{k} P']\!]$



$[\![P \overset{k,q}{\Longrightarrow} P']\!]$



Using Quantum Transition Capacity for Parallel 0-preimage Finding

Let A = q k-parallel query algorithm outputs x. Say A win if H(x) = 0.

$$\begin{aligned}
\Pr[A \text{ win in } H] &\leq \Pr[A \text{ win in } D] + 2\sqrt{1/M} \\
&\leq \Pr[D \in ZR] + 2\sqrt{1/M} \\
&\leq [\![\perp \overset{q,k}{\Longrightarrow} ZR]\!]^2 + 2\sqrt{1/M} \\
&\leq \left(\sum_{s=1}^{q} [\![\neg ZR \overset{k}{\rightarrow} ZR]\!]\right)^2 + 2\sqrt{1/M}. \\
&\leq O\left(\frac{kq^2}{M}\right)
\end{aligned}$$

Step 1. Breaking Up Quantum Transition Capacity

Lemma 3: For any sequence of database properties $P_0, P_1, \ldots, P_q$,

$$\left[\!\left[\neg P_0 \overset{k,q}{\Longrightarrow} P_q\right]\!\right] \leq \sum_{s=1}^{q} \left[\!\left[\neg P_{s-1} \overset{k}{\rightarrow} P_s\right]\!\right].$$

Proof: Identical to the sequential case.

By setting $P_0 = P_1 = \cdots = P_q = ZR$ and Lemma 3, we have

$$[\![\bot \overset{k,q}{\Longrightarrow} ZR]\!] \leq \sum_{s=1}^{q} [\![\neg ZR \overset{k}{\rightarrow} ZR]\!]$$

Step 2. Bounding $[\![\neg ZR \xrightarrow{k} ZR]\!] = \max_{\boldsymbol{x},\hat{\boldsymbol{y}}} \left\| ZRcO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - ZR) \right\|_{op}$

A naïve attempt: for any $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$, decompose

$$\left\| ZR\ cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - ZR) \right\|_{op} \leq \sum_{i=1}^{k} \left\| ZR\ cO_{x_i\hat{y}_i}(I - ZR) \right\|_{op},$$

by inserting $I = ZR + (I - ZR)$ and bounding $\left\| ZR\ cO_{x_i\hat{y}_i}(I - ZR) \right\|_{op}$ as before.

However, this only gives

$$[\![\neg ZR \xrightarrow{k} ZR]\!] \leq O(k/\sqrt{M})$$

and only implies

$$\Pr[A \text{ win in } H] \leq O(k^2 q^2/\sqrt{M}).$$

Step 2. Bounding $[\![\neg ZR \xrightarrow{k} ZR]\!] = \max_{\boldsymbol{x},\hat{\boldsymbol{y}}} \left\| ZRcO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - ZR) \right\|_{op}$

Given any $\boldsymbol{x}, \hat{\boldsymbol{y}}$, $cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}$ acting as "resampling" $\boldsymbol{x}$ which does not perturb other entries.

For any fixed $\boldsymbol{x}$ and D, define $D_{-\boldsymbol{x}}$ as the partial database obtained by removing the $\boldsymbol{x}$ entries.

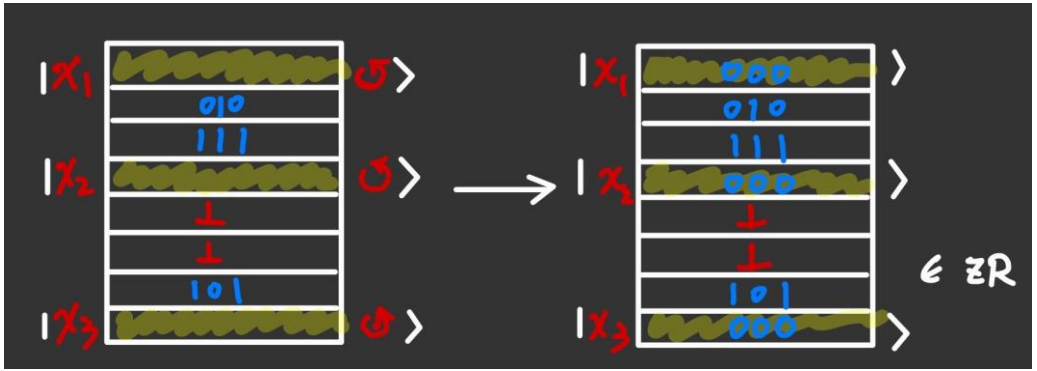
Note that the space $|D_{-\boldsymbol{x}}\rangle \otimes \mathbf{cO}_{\boldsymbol{x}}$ is an invariant subspace of $cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}$.

$|D_{-\boldsymbol{x}}\rangle \otimes \mathbf{cO}_{\boldsymbol{x}}$ is also invariant under projectors ZR and $(I - ZR)$.

By the property of operator norm,

$$\left\| ZR\ cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}\ (I - ZR) \right\|_{op}$$
$$\leq \max_{D_{-\boldsymbol{x}}} \left\| ZR|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}} (I - ZR)|_{D_{-\boldsymbol{x}}} \right\|_{op}$$

where $ZR|_{D_{-\boldsymbol{x}}}$ and $(I - ZR)|_{D_{-\boldsymbol{x}}}$ are the projectors restricted in $|D_{-\boldsymbol{x}}\rangle \otimes \mathbf{cO}_{\boldsymbol{x}}$



Restriction of Database Property

Let P be a database property and $D_{-\boldsymbol{x}}$ a partial database.

The restricted property $P|_{D_{-\boldsymbol{x}}}$ is now a subset of partial databases

$$\widehat{D}: \{x_1, \dots, x_k\} \to \{0,1\}^m$$

For example, for any given $\boldsymbol{x} = \{x_1, \dots, x_k\}$ and D s.t. $D(\bar{x}) \neq 0 \; \forall \; \bar{x} \notin \{x_1, \dots, x_k\}$,

$$ZR|_{D_{-\boldsymbol{x}}} = \{\widehat{D} | \exists x \in \{x_1, \dots, x_k\}: \widehat{D}(x) = 0\}.$$

“Local properties” “recognizing” $ZR|_{D_{-\boldsymbol{x}}}$

For any given $\boldsymbol{x} = \{x_1, \dots, x_k\}$ and D s.t. $D(\bar{x}) \neq 0 \; \forall \; \bar{x} \notin \{x_1, \dots, x_k\}$,

$$ZR|_{D_{-\boldsymbol{x}}} = \{\widehat{D} | \exists x \in \{x_1, \dots, x_k\} : \widehat{D}(x) = 0\}.$$

For any $i \in [k]$, define the “local property”

$$L_i|_{D_{-\boldsymbol{x}}} = \{\widehat{D} : \widehat{D}(x_i) = 0\}.$$

Note that $ZR|_{D_{-\boldsymbol{x}}} = \bigcup_i L_i|_{D_{-\boldsymbol{x}}}$.

A Decomposition Lemma

Lemma 4: Let P and $\{L_i\}_i$ be database properties such that

$$P = \cup_i L_i,$$

then for all $\boldsymbol{x}, \hat{\boldsymbol{y}}$

$$\left\|P\ cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-P)\right\|_{op}^2 \leq \sum_{i=1}^{k}\left\|L_i\ cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-L_i)\right\|_{op}^2.$$

Proof: Define $M_i = L_i \setminus \cup_{j<i} L_j$ for every i. Note that $\{M_i\}$ are mutually orthogonal.

$$\left\|PcO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-P)|\right\|_{op}^2 \leq \sum_{i=1}^{k}\left\|M_i cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-P)|\right\|_{op}^2$$

$$\leq \sum_{i=1}^{k}\left\|L_i cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-L_i)\right\|_{op}^2 = \sum_{i=1}^{k}\left\|L_i cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I-L_i)\right\|_{op}^2,$$

where we use property of operator norm, $P \leq \sum_i \mathrm{M}_i \leq \sum_i L_i$ and $\mathrm{M}_i \leq L_i \leq P$ for any i.

Properties of the operator norm

3. Let A be an operator and $\{M_i\}$ be a set of mutually orthogonal operators. We have

$$\|\sum_i M_i A\|_{op}^2 \leq \sum_i \|M_i A\|_{op}^2.$$

Step 2. Bounding $[\![\neg ZR \xrightarrow{k} ZR]\!] = \max_{\boldsymbol{x},\hat{\boldsymbol{y}}} \left\| ZRcO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - ZR) \right\|_{op}$

For any $\boldsymbol{x}, \hat{\boldsymbol{y}}, D_{-\boldsymbol{x}}$, applying Lemma 4 to $ZR|_{D_{-\boldsymbol{x}}}$ and $L_i|_{D_{-\boldsymbol{x}}}$,

$$\left\| ZR|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - ZR)|_{D_{-\boldsymbol{x}}} \right\|_{op}^2 \leq \sum_{i=1}^{k} \left\| L_i|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - L_i)|_{D_{-\boldsymbol{x}}} \right\|_{op}^2$$

$$\leq \sum_{i=1}^{k} \left\| L_i|_{D_{-\boldsymbol{x}}} cO_{x_i\hat{y}_i}(I - L_i)|_{D_{-\boldsymbol{x}}} \right\|_{op}^2 \leq \sum_{i=1}^{k} \left\| L_i|_{D_{-\boldsymbol{x}}} cO_{x_i\hat{y}_i}(I - L_i)|_{D_{-\boldsymbol{x}}} \right\|_{F}^2 \leq \frac{10k}{M}$$

Therefore, we have $[\![\neg ZR \to ZR]\!] = O(\sqrt{k/M})$.

Quantum Lazy Sampling for Parallel 0-preimage Finding

Let A = q k-parallel query algorithm outputs x. Say A win if H(x) = 0.

$$\begin{aligned}
\Pr[A \text{ win in } H] &\leq \Pr[A \text{ win in } D] + 2\sqrt{1/M} \\
&\leq \Pr[D \in ZR] + 2\sqrt{1/M} \\
&\leq [\![\perp \overset{q,k}{\Longrightarrow} ZR]\!]^2 + 2\sqrt{1/M} \\
&\leq \left(\sum_{s=1}^{q} [\![\neg ZR \overset{k}{\rightarrow} ZR]\!]\right)^2 + 2\sqrt{1/M}. \\
&\leq O\left(\frac{kq^2}{M}\right)
\end{aligned}$$

Parallel Collision Finding: Bounding $\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right]$

In the parallel setting, for each query, collision occurs not only between new queries and database but also occurs between different points of the query.

Consider any fixed $\boldsymbol{x} = (x_1, \dots, x_k)$ and $D \in SZ_{\leq ks}$ with $s \leq q - 1$.

For any $i \neq j$, we define

$$CL_i|_{D_{-\boldsymbol{x}}} = \left\{\widehat{D} \middle| \bar{x} \notin \{x_1, \dots, x_k\} : \widehat{D}(x_i) = D(\bar{x}) \neq \perp\right\}$$

$$\text{and } CL_{i,j}|_{D_{-\boldsymbol{x}}} = \{\widehat{D} | \widehat{D}(x_i) = \widehat{D}(x_j) \neq \perp\}.$$

It holds that $CL|_{D_{-\boldsymbol{x}}} = \left(\bigcup_i CL_i|_{D_{-\boldsymbol{x}}}\right) \cup \left(\bigcup_{i.j} CL_{i,j}|_{D_{-\boldsymbol{x}}}\right)$.

Parallel Collision Finding: Bounding $\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right]$

For any $\boldsymbol{x}\ \hat{\boldsymbol{y}}, D_{-\boldsymbol{x}}$, applying Lemma 4,

$$\left\| CL|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - CL)|_{D_{-\boldsymbol{x}}} \right\|_{op}^2$$

$$\leq \sum_i \left\| CL_i|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}\left(I - CL_i|_{D_{-\boldsymbol{x}}}\right) \right\|_{op}^2 + \sum_{i,j} \left\| CL_{i,j}|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - CL_{i,j}|_{D_{-\boldsymbol{x}}}) \right\|_{op}^2$$

Case 1 ($CL_i|_{D_{-\boldsymbol{x}}}$): For any i, we have

$$\left\| CL_i|_{D_{-\boldsymbol{x}}} cO_{\boldsymbol{x}\hat{\boldsymbol{y}}}(I - CL_i|_{D_{-\boldsymbol{x}}}) \right\|_{op}^2 = \left\| CL_i|_{D_{-\boldsymbol{x}}} cO_{x_i\hat{y}_i}(I - CL_i|_{D_{-\boldsymbol{x}}}) \right\|_{op}^2 \leq \frac{10k(q-1)}{M}.$$

Parallel Collision Finding: Bounding $\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right]$

Case 2 ($CL_{i,j}|_{D_{-x}}$): Insert $\underbrace{(CL_{i,j}|_{D_{-x}} + (I - CL_{i,j}|_{D_{-x}}))}$

$$\left\| CL_{i,j}|_{D_{-x}} cO_{x\hat{y}}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op} = \left\| CL_{i,j}|_{D_{-x}} cO_{x_i\hat{y}_i} \cdot cO_{x_j\hat{y}_j}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op}$$

$$\leq \left\| CL_{i,j}|_{D_{-x}} cO_{x_i\hat{y}_i} CL_{i,j}|_{D_{-x}} cO_{x_j\hat{y}_j}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op}$$

$$+ \left\| CL_{i,j}|_{D_{-x}} cO_{x_i\hat{y}_i}(I - CL_{i,j}|_{D_{-x}}) cO_{x_j\hat{y}_j}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op}$$

$$\leq \left\| CL_{i,j}|_{D_{-x}} cO_{x_j\hat{y}_j}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op} + \left\| CL_{i,j}|_{D_{-x}} cO_{x_i\hat{y}_i}(I - CL_{i,j}|_{D_{-x}}) \right\|_{op}$$

$$\leq \sqrt{10/M} + \sqrt{10/M} = 2\sqrt{10/M}.$$

Parallel Collision Finding: Bounding $\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right]$

Combine everything together, for every s we have

$$\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right]^2$$

$$\leq \sum_i \left\| CL_i|_{D_{-x}} cO_{x\hat{y}} \left(I - CL_i|_{D_{-x}} \right) \right\|_{op}^2$$

$$+ \sum_{i,j} \left\| CL_{i,j}|_{D_{-x}} cO_{x\hat{y}} \left(I - CL_{i,j}|_{D_{-x}} \right) \right\|_{op}^2$$

$$\leq k \cdot \frac{10k(q-1)}{M} + k^2 \cdot \frac{40}{M} = O(k^2 q/M).$$

Thus, it holds that $\left[\!\left[SZ_{\leq ks} \setminus CL \xrightarrow{k} CL \right]\!\right] \leq O\left(\sqrt{\frac{k^2 q}{M}}\right)$.

Quantum Lazy Sampling for Parallel Collision Finding

Let $A = q$ k-parallel query algorithm outputs (x, x'). Say A win if $H(x) = H(x')$.

$$\begin{aligned}
\Pr[A \text{ win in } H] &\leq \Pr[A \text{ win in } D] + 2\sqrt{2/M} \\
&\leq \Pr[D \in CL] + 2\sqrt{2/M} \\
&\leq [\![\perp \overset{q,k}{\Longrightarrow} CL]\!]^2 + 2\sqrt{2/M} \\
&\leq \left(\sum_{s=0}^{q-1} [\![SZ_{\leq ks} \setminus CL \overset{k}{\rightarrow} CL]\!]\right)^2 + 2\sqrt{2/M}. \\
&\leq O\left(\frac{k^2 q^3}{M}\right).
\end{aligned}$$

Summary: Compressed Oracle as Quantum Lazy Sampling

Compressed oracle as a quantum lazy sampling technique.

Mimic classical lazy sampling closely:

1. Initially, initialize an empty (all-⊥) “database”
2. Record A’s q query in the “database” in some way
3. Upper bound Pr[“database” has certain property]
4. Show Pr[A win] ≤ Pr[“database” has certain property] + something small

- Powerful enough to obtain tight (parallel) bound for
  - 0-preimage finding & collision finding
- Also more advanced examples:
  - zk-SNARK [CMS’19], PoSW [BLZ’21,CFHL’21], etc
- Can be abstracted to capture and lift classical lazy sampling proofs [CFHL’21]

Many More Applications of Compressed Oracle Techniques

[Zhandry'19]
Indifferentiability of MD

[Hosoyamada-Iwata'19]
4-round Luby-Rackoff

[Grilo-Hövelmanns-Hülsing-Majenz'21]
Adaptive programming QROM

[Hosoyamada-Iwata'21]
Tight security for HMAC and NMAC

[Bindel-Hamburg-Hövelmanns-Hülsing-Persichetti'19]
Tighter CCA security proofs

[Chiesa-Manohar-Spooner'19]
zk-SNARKs

[Blocki-Lee-Zhou'21, C-Fehr-Huang-Liao'21]
Proof of Sequential Work

[Alagic-Majenz-Russell-Song'20]
Quantum-secure signature seperation

[Don-Fehr-Majenz-Schaffner'21]
Online extraction

[Liu-Zhandry'19a]
Tight bounds for multi-collision problem

[C-Guo-Liu-Qian'20]
Quantum Time-space tradeoffs

[Liu-Zhandry'19b]
Quantum Fiat-Shamir

Thank you! Questions?