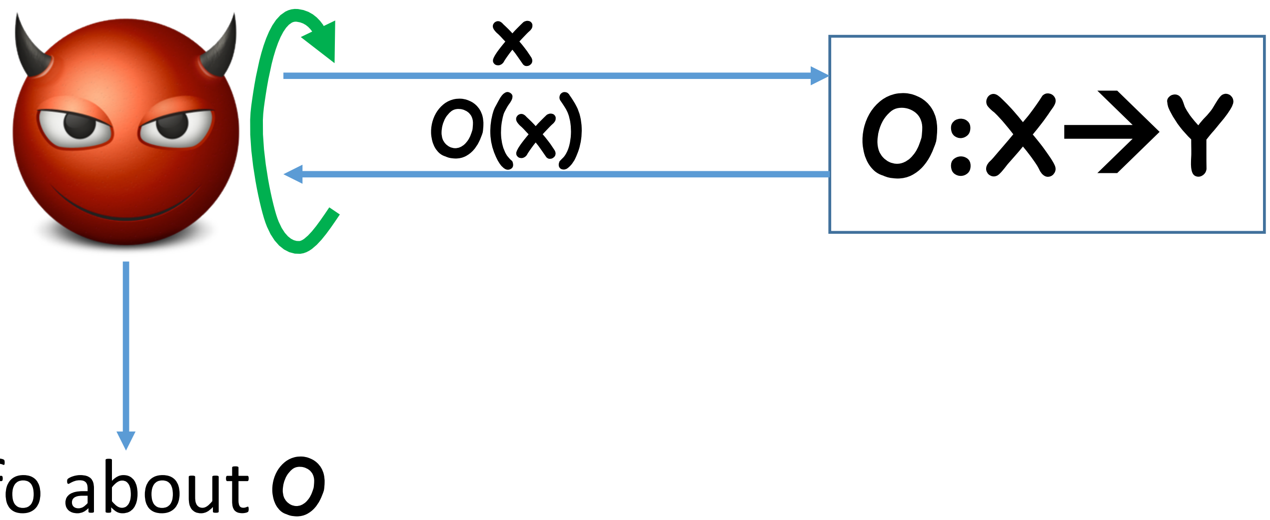
Quantum Oracle Classification

The Case of Group Structure

Mark Zhandry – Princeton University

Query Complexity



Examples:

- Pre-image of given output
- Collision
- Complete description of ***O***
- …

Motivations

Playground for theoretical computer science

- Don’t pay attention to running times
- Only care about number of queries
- Can actually give rigorous hardness proofs!

Motivations

Models "brute force" attacks on crypto

- E.g. Hardness of inverting a black box function

  $\geq$

  Hardness of inverting **any** concrete function

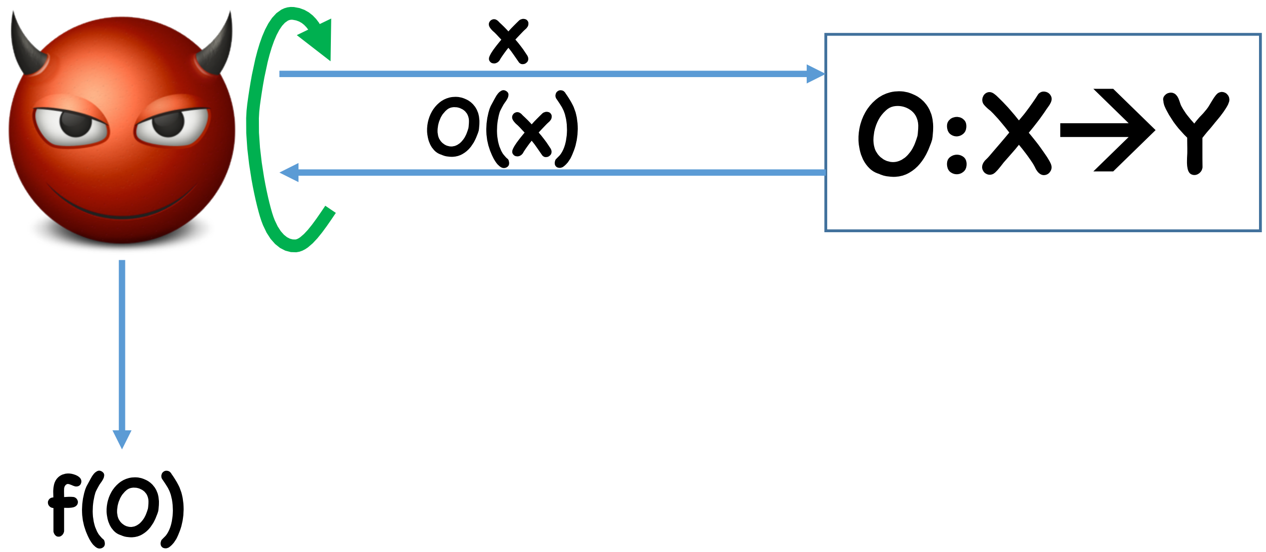
- Often, best known attacks are brute force
- Gives guidance for setting parameters

Motivations

Attack models for certain crypto primitives

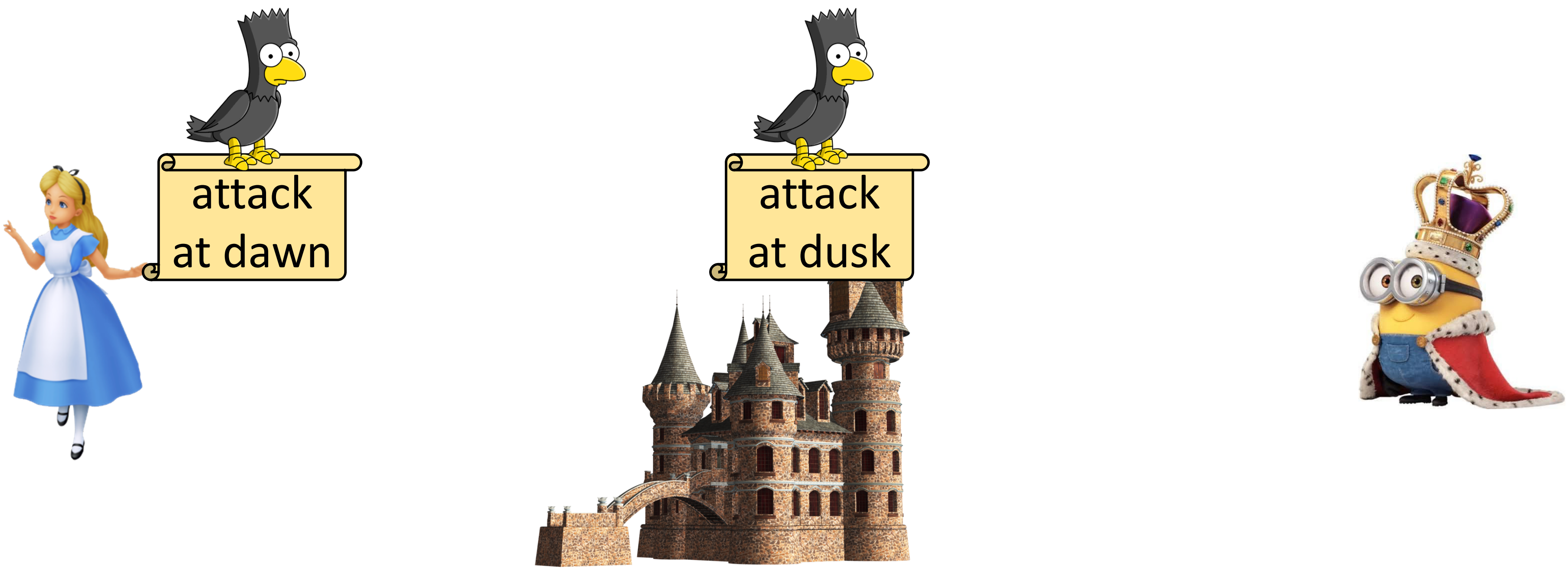
- More on this in a moment

Oracle Classification

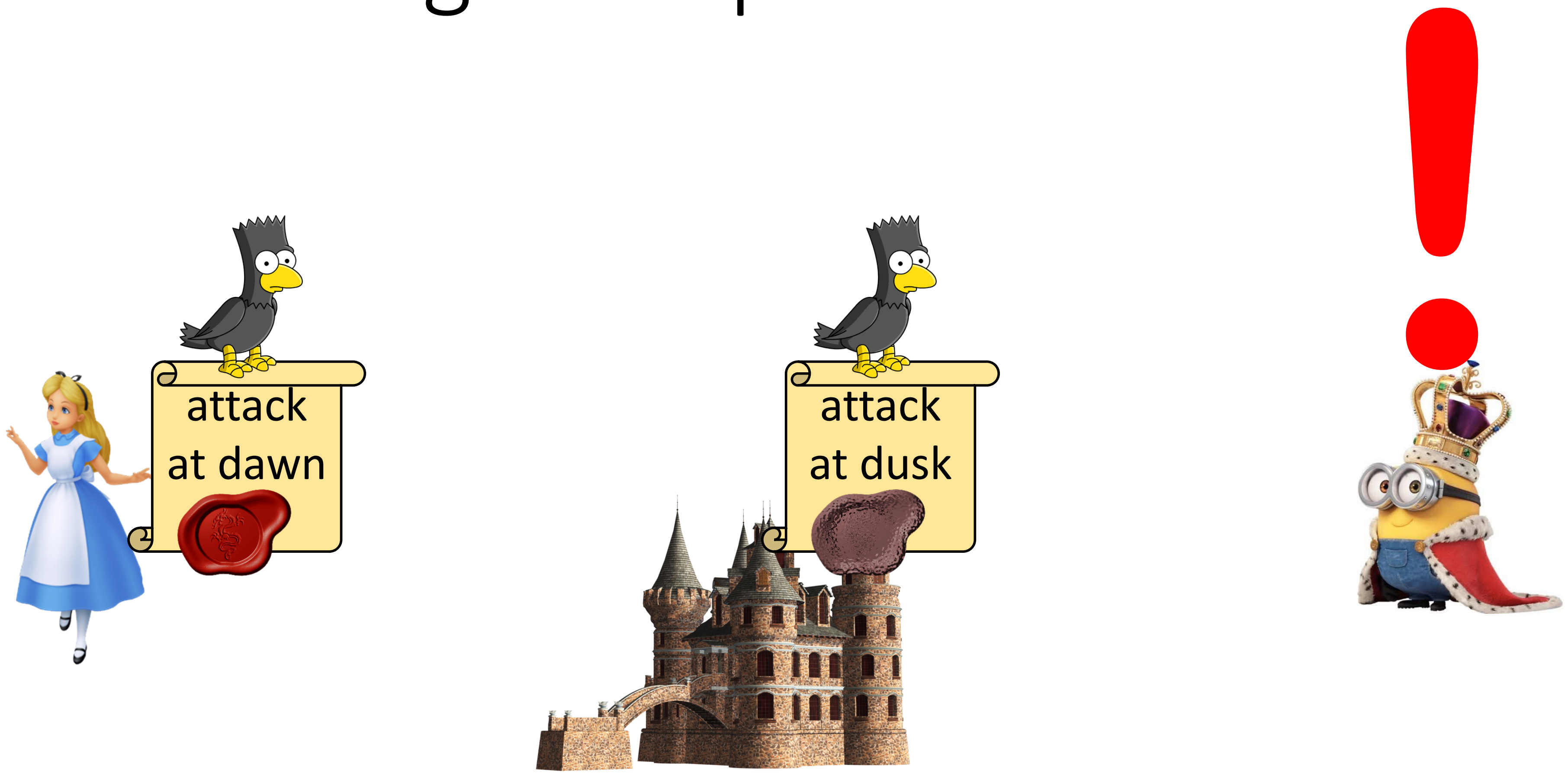


Excludes some problems like collision finding and inversion

Motivating Example: MACs



Motivating Example: MACs



Solution: Message Authentication Codes

Message Authentication Codes

MAC(k,m) → σ
Ver(k,m,σ) → Accept/Reject

Correctness: ∀ **k,m, Ver(k, m, MAC(k,m)) = Accept**

1-time security:

Given **m≠m', σ = MAC(k,m)**, impossible to produce **σ'** s.t. **Ver(k, m', σ') = Accept**

- Variants: adversary picks **m**, picks **m'** after seeing **σ'**

2-time security...

Constructing MACs

1-time secure construction:

$k = (a,b)$

$MAC(k, m) = a\ m + b$

$Ver(k, m, \sigma) = \text{Accept iff } \sigma = a\ m + b$

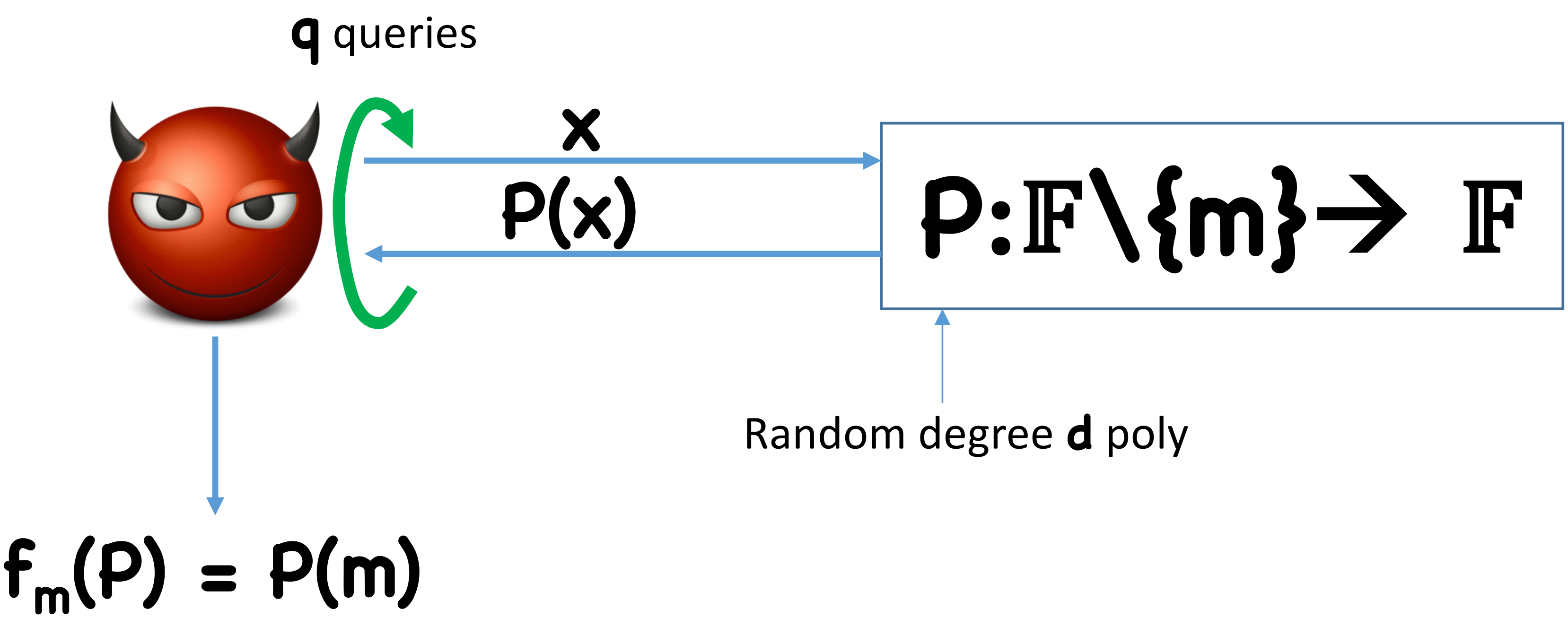
q-time secure construction:

$k =$ random degree $d=q$ polynomial P

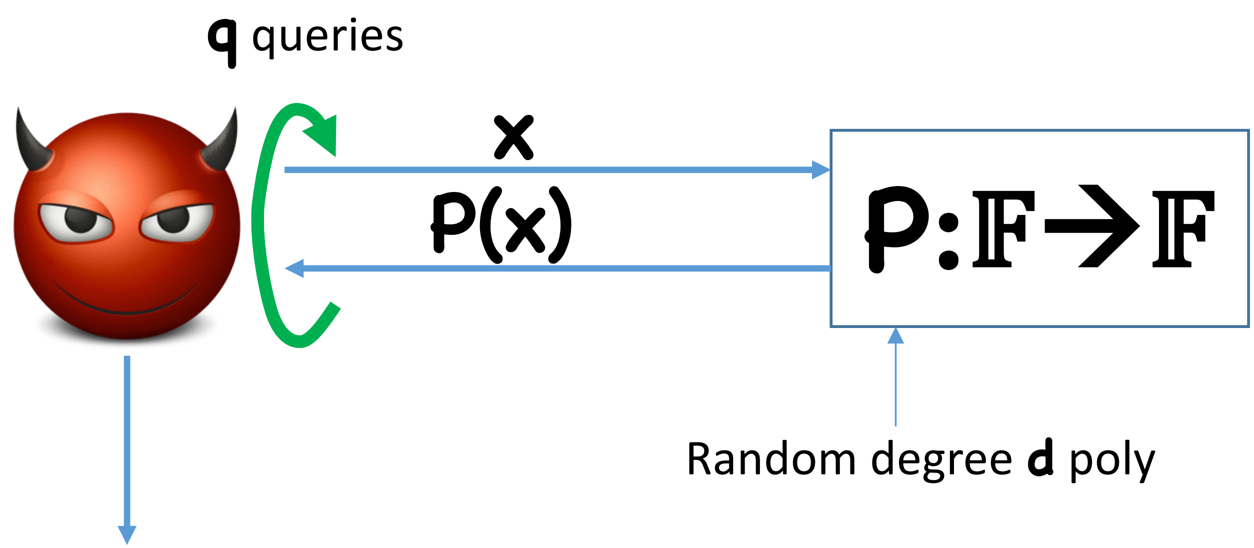
$MAC(P,m) = P(m)$

$Ver(P, m, \sigma) = \text{Accept iff } \sigma = P(m)$

q-time MACs as Oracle Classification



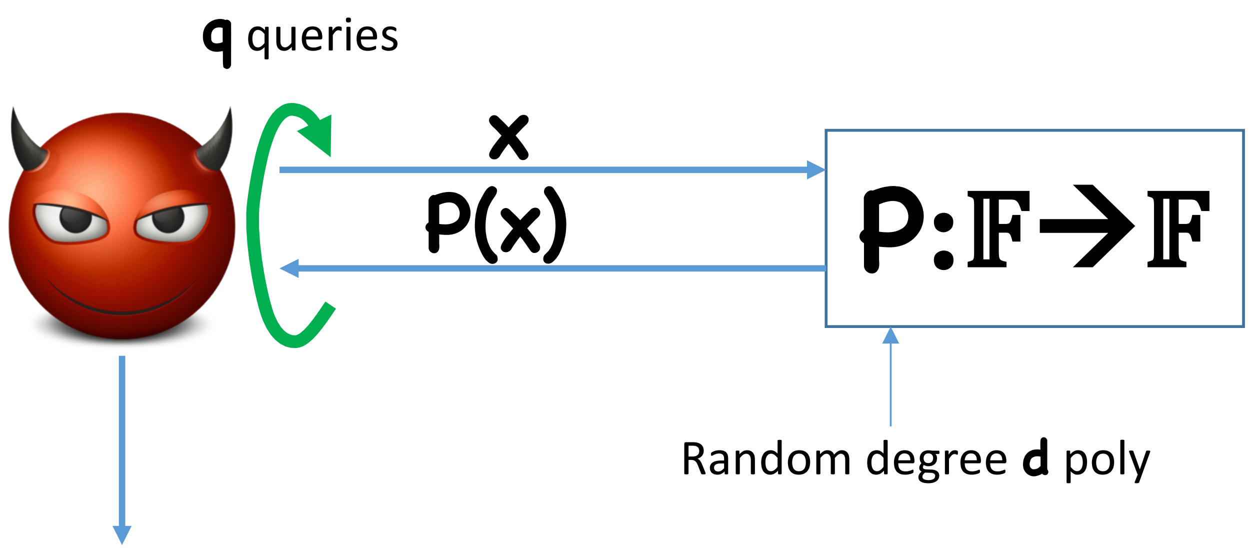
q-time MACs as Oracle Classification



$$f_{m_0,m_1,\ldots,m_q}(P) = (\ P(m_0),\ P(m_1),\ \ldots,\ P(m_q)\ )$$

For MAC experiment, really want to let adversary choose $m_0, \ldots, m_q$

q-time MACs as Oracle Classification

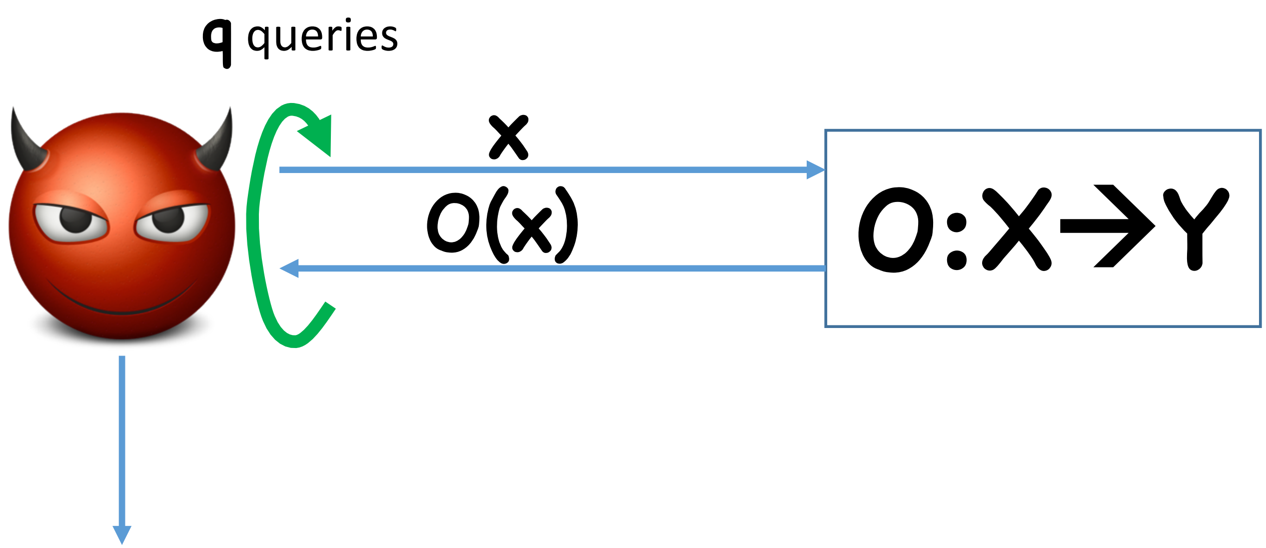


f, f(P) Where $f \in \mathcal{F}^{eval}_{q+1} = \{f_{m_0,m_1,\ldots,m_q}\}$

Straightforward:

Maximal success probability for **d≥q** is **1/𝔽**

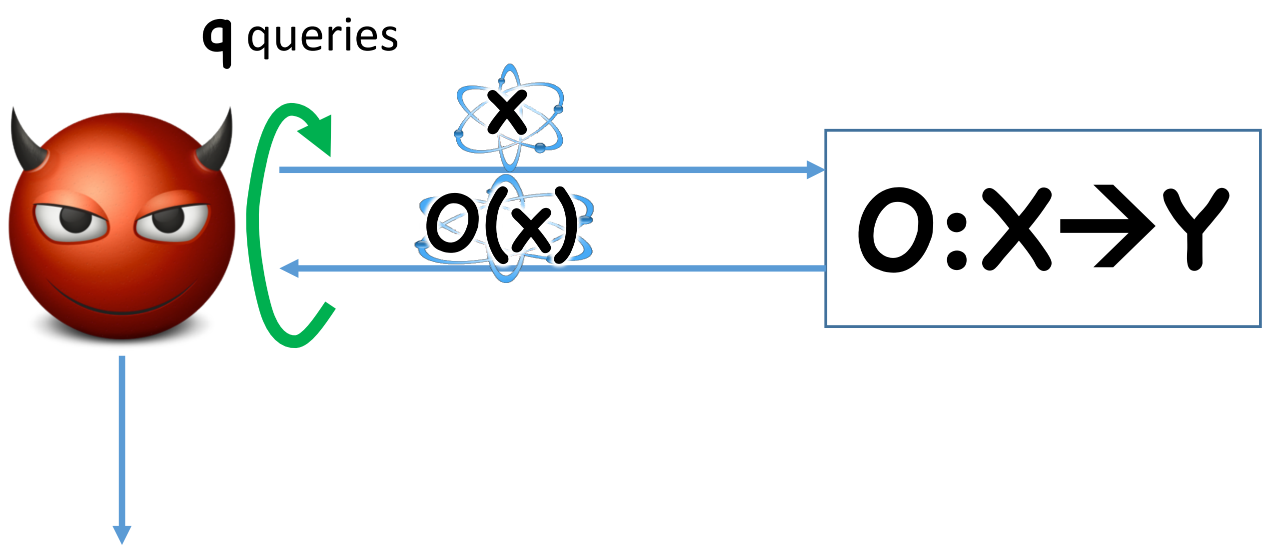
“Adaptive” Oracle Classification



f, f(O) Where $\mathbf{f} \in \mathcal{F}$

And now for quantum…

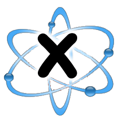
Quantum Oracle Classification



f, **f(O)** Where **f** $\in \mathcal{F}$

Quantum Background

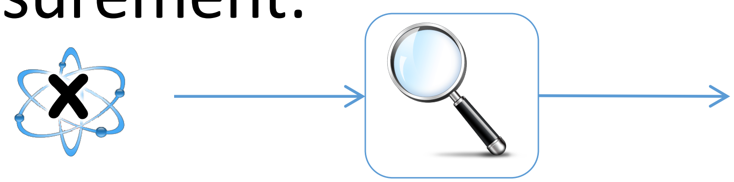
Quantum states:



= superposition of **all** messages

$= \Sigma \alpha_x |x\rangle \qquad (\Sigma |\alpha_x|^2 = 1)$

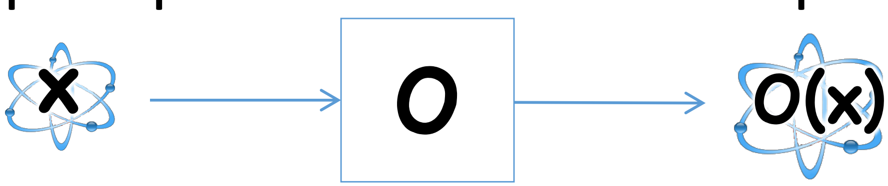
Measurement:



x with probability $|\alpha_x|^2$

Operations: Unitary transformations on amplitude vectors

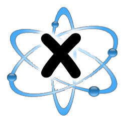
Example op: simulate classical ops in superposition



$= \Sigma \alpha_x |O(x)\rangle$

Quantum Background

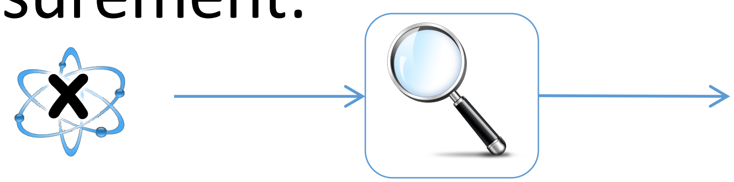
Quantum states:



= superposition of **all** messages

$= \Sigma \alpha_x |x\rangle \qquad (\Sigma|\alpha_x|^2 = 1)$

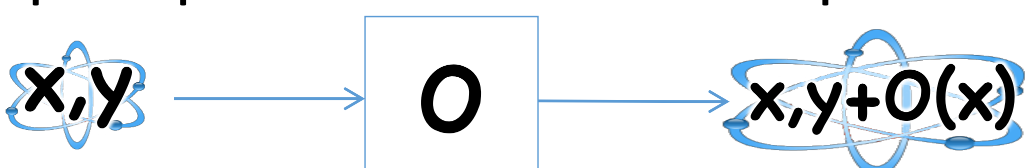
Measurement:



x with probability $|\alpha_x|^2$

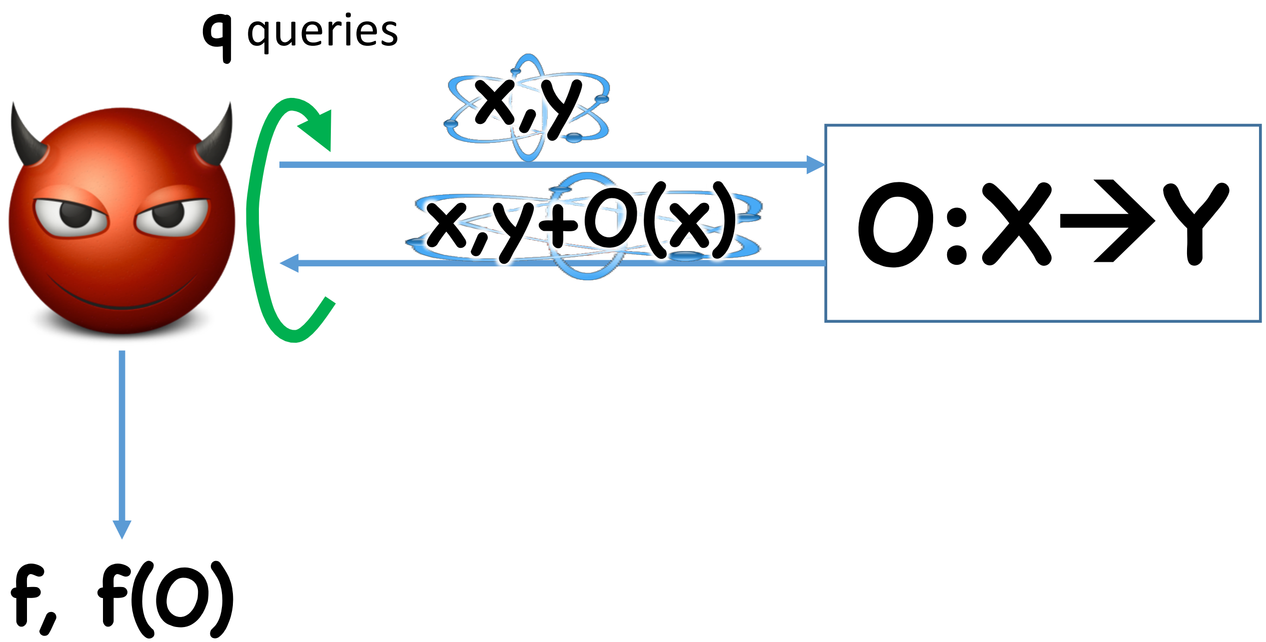
Operations: Unitary transformations on amplitude vectors

Example op: simulate classical ops in superposition:



$= \Sigma \alpha_{x,y} |x, y+O(x)\rangle$

Quantum Oracle Classification



High-Level Questions

Speedup vs classical queries?

Sequential vs parallel queries?

Adaptively vs statically chosen $\mathbf{f}$?

Average case vs worst case?

Low Level Questions

Calculate exact number of queries needed (classically/quantumly, **f** before/after, sequential/parallel)

Better yet: calculate exact optimal success probability given certain number of queries

Difficulty:

- Quantum algorithms “see” entire oracle
- But, info is stuck in quantum superposition
- Difficult to determine how much info can be extracted via measurement

Group Structure

$\mathbf{Y}$ = additive abelian group

Notice: Set of functions $\mathbf{O}$ forms group $\equiv \mathbf{Y}^{|X|}$

$\mathbf{A}$ = subspace of $\mathbf{Y}^{|X|}$

$\mathbf{O}$ sampled uniformly from $\mathbf{A}$

$\mathcal{F}$ = subset of homomorphisms on $\mathbf{A}$

(Y,A,$\mathcal{F}$,q)−Group Quantum Oracle Classification :
Determine maximal success probability of **q**-query quantum algorithm

Examples

Function Classes:

- All functions
- (single/multivariate) Polynomials of given degree

Homomorphisms:

- Identity: $\mathbf{f(O) = O}$
- Evaluation: $\mathbf{f_S(O) = (\ O(x)\ )_{x \in S}}$
- Summation: $\mathbf{f(O) = \Sigma_{x \in X} O(x)}$

Captures Many Known and New Problems

- Parity: $\mathbf{\Sigma O(x) \bmod 2}$
- Polynomial interpolation: Learn $\mathbf{P}$ entirely
- Polynomial extrapolation: Learn $\mathbf{P(x)}$
- Oracle Interrogation: $\mathbf{(P(x_1),\ldots,P(x_n))}$ for **n>q**
- **Polynomials as q-time MACs**

This Work: “Complete” Solution to Quantum Group QOC problem

Notation

Let $\mathbf{P}_{qm,sp,as,wa}$ for

- **qc** ∈ **{Quantum, Classical}**
- **sp** ∈ **{Sequential, Parallel}**
- **as** ∈ **{Adaptive, Static}**
- **wa** ∈ **{Worst, Average}**

be the optimal **wa**-case success probability for algorithms making **sp qc** queries, and where **f** is chosen **as**-ly.

Trivialities

Classical ≤ Quantum
Parallel ≤ Sequential
Static ≤ Adaptive
Worst ≤ Average

High-Level Theorems

Thm (easiest): Worst = Average

Thm (less easy): If **qc = Classical**,
Parallel = Sequential
Static = Adaptive
Plus: simplish* expression for $P_{classical}$

Thm (hard): If **qc = Quantum**,
Parallel = Sequential
Static = Adaptive
Plus: simplish* expression for $P_{Quantum}$

*based on structure of groups only, no mention of "quantum" or "classical"

High-Level Theorems

Thm (easiest): Worst = Average

Thm (less easy): If qc = Classical,
Parallel = Sequential

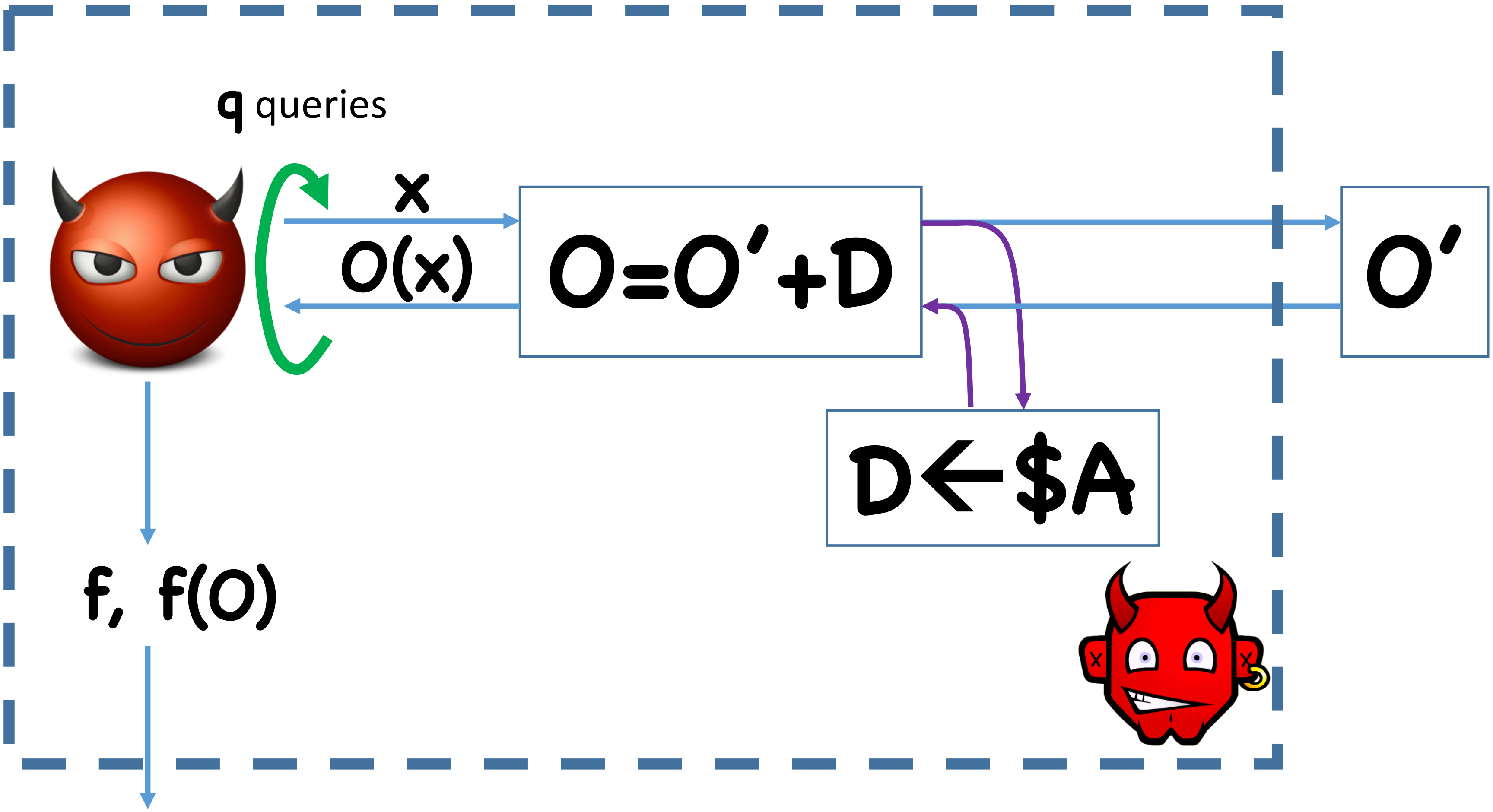
Thus, only distinction for group setting is:

classical vs **quantum**

Thm (hard): If qc = Quantum,
Parallel = Sequential
Static = Adaptive
Plus: simplish* expression for P_{Quantum}

*based on structure of groups only, no mention of "quantum" or "classical"

Worst = Average



f, f(O)–f(D) = f(O–D) = f(O')

Works equally well for classical and quantum queries

Proof Sketch: Classical Case

Queries $O(x_1),\ldots O(x_q)$ yield homomorphism $e \in \mathcal{F}^{eval}{}_q$

q queries $\Rightarrow$ $e(O)$ for some $e \in \mathcal{F}^{eval}{}_q$

- i.e. learn O up to value $Q \in Ker(e)$

Can learn $f(O)$ with certainty if $Ker(e) \subseteq Ker(f)$

- More generally, success prob $=$

$$p_{classical} = \frac{|\ Ker(f) \cap Ker(e)\ |}{|\ Ker(e)\ |}$$

Proof Sketch: Classical Case

Optimal success probability:

$$P_{classical} = \max_{\substack{e \in \mathcal{F}^{eval}_q \\ f \in \mathcal{F}}} \left(\frac{|\,Ker(f) \cap Ker(e)\,|}{|\,Ker(e)\,|} \right)$$

Straightforward to show that sequential queries, adaptive **f** don't help

- Intuition: query responses independent of kernel structure

Quantum Case?

More complicated…

For this talk, consider special case:

Y is a field, **f** are linear transformations

Notation

Let $\mathbf{B} = \mathrm{Ker}(\mathbf{f})$

- Let $\{\mathbf{b}_1 \ \dots \ \mathbf{b}_r\}$ be basis for $\mathbf{B}$

Identify $\mathbf{f}(\mathbf{O})$ with coset of $\mathbf{B}$ that contains $\mathbf{O}$

Define $\mathbf{C} = \mathbf{A}/\mathbf{B}$

- $\mathbf{f} \equiv (\mathbf{B},\mathbf{C})$
- Let $\{\mathbf{c}_1 \ \dots \ \mathbf{c}_s\}$ be a basis for $\mathbf{C}$

Notation

For vector $\bar{x} \in X^q$, define

$$B(\bar{x}) = \begin{pmatrix} b_1(x_1) & b_1(x_2) & \cdots & b_1(x_q) \\ b_2(x_1) & b_2(x_2) & \cdots & b_2(x_q) \\ \vdots & \vdots & & \vdots \\ b_r(x_1) & b_r(x_2) & \cdots & b_r(x_q) \end{pmatrix}$$

$$C(\bar{x}) = \begin{pmatrix} c_1(x_1) & c_1(x_2) & \cdots & c_1(x_q) \\ c_2(x_1) & c_2(x_2) & \cdots & c_2(x_q) \\ \vdots & \vdots & & \vdots \\ c_r(x_1) & c_r(x_2) & \cdots & c_r(x_q) \end{pmatrix}$$

Theorem: Quantum Case

Optimal success probability:

$$P_{quantum} = \max_{B,C,h} \left(\frac{|\{C(\bar{x}) \cdot \bar{r} : \; B(\bar{x}) \cdot \bar{r} = h\}|}{|C|} \right)$$

Where $\bar{x} \in X^q$, $\bar{r} \in Y^q$

Extends to any setting where we can induce a ring structure on **Y** such that **B,C** are free modules

Proving the Theorem…

Proof Sketch for Quantum Theorem

First attempt:

Let $|\Psi_O\rangle$ be final state of query algorithm

Rank method([BZ'13]):

- Bound on dimension of $\mathrm{Span}\{|\Psi_O\rangle\}$ in terms of q
- Success probability/random guessing $=$ $\mathrm{Span}\{|\Psi_O\rangle\}$

Gives immediate upper bound on success prob

- Works well when all functions are possible, goal is to find entire function

Proof Sketch for Quantum Theorem

First attempt:

Problem:

- Rank grows with number of possible functions
- Guessing probability shrinks with number of possible outputs
- Mismatch when either:
    - Constraints on oracles (e.g. polynomials)
    - Goal isn't to find entire function

- Works well when all functions are possible, goal is to find entire function

Proof Sketch for Quantum Theorem

Second attempt:

For a given $\mathbf{v}$, let $\boldsymbol{\rho}_\mathbf{v}$ be the "state" representing $|\Psi_O\rangle$ for a random O such that $\mathbf{f}(O) = \mathbf{v}$

- Called a "mixed" state
- Intuition: maybe rank only grows with number of equivalence classes induced by $\mathbf{f}$?

Problem: No general Rank method for "mixed" states

Proof Sketch for Quantum Theorem

Final solution:

For a given $\mathbf{v}$, let $\boldsymbol{\rho}_{\mathbf{v}}$ be the "state" representing $|\Psi_O\rangle$ for a random $\boldsymbol{O}$ such that $\mathbf{f}(O) = \mathbf{v}$

Use group structure to "purify" mixed state
- Analyze rank of purified state
- Get bound on success probability
- "Luckily" turns out to be optimal for group structure

Analysis still depends on kernels of homomorphisms
- Adaptivity/sequentiality don't help

Applying the Theorem…

Quantum Oracle Summation

Compute $\sum O(x)$ for a random function O

- Write $X = [0,\ldots,N-1]$

- $B = \{O$ such that $\sum O(x) = 0\}$
  $\Rightarrow b_i(x) = \delta_{i,x} - \delta_{0,x}$ for $i=1,\ldots,N-1$

- $C = \{O$ such that $O(x)=0 \;\; \forall x \neq 0\}$
  $\Rightarrow c(x) = \delta_{0,x}$

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
  - If $\bar{\mathbf{x}}$ does **not** contain **0**:

$$\mathbf{B}(\bar{\mathbf{x}}) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

← **q** **1**'s in rows corresponding to elements in $\bar{\mathbf{x}}$

$\Rightarrow$ **h** must be **0** in all but **q** (that is, **N–1–q**) positions

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
  - If $\bar{\mathbf{x}}$ **does** contain **0**:

$$\mathbf{B}(\bar{\mathbf{x}}) = \begin{pmatrix} -1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$$

(by reordering $\bar{\mathbf{x}}, \bar{\mathbf{r}}$, can assume **0** is first coordinate of $\bar{\mathbf{x}}$)

$\Rightarrow$ $\mathbf{h}$ must be $\mathbf{r}_1$ in all but $\mathbf{q-1}$ (that is, $\mathbf{N-q}$) positions

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
- Determine $\mathbf{z} = \mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$
    - If $\bar{\mathbf{x}}$ does not contain $\mathbf{0}$:

$$\mathbf{C}(\bar{\mathbf{x}}) = (0\ 0\ 0)$$

$$\Rightarrow \mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{0}$$

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
- Determine $\mathbf{z} = C(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$
  - If $\bar{\mathbf{x}}$ does contain $\mathbf{0}$:

$$C(\bar{\mathbf{x}}) = (1 \; 0 \; 0)$$

$$\Rightarrow C(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = r_1$$

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
- Determine $\mathbf{z} = \mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$
- Count $\mathbf{z}$'s:
  - Non-zero $\mathbf{z}$'s set $\mathbf{M-q}$ coordinates of $\mathbf{h}$
  - $\mathbf{z=0}$ sets $\mathbf{M-q-1}$ coordinates
  - $\mathbf{k}$ = total number of possible $\mathbf{z}$'s for any $\mathbf{h}$:

$$M-q-1 + (k-1)\,(M-q) \leq M-1$$

$$k \leq \lfloor M/(M-q) \rfloor$$

Quantum Oracle Summation

- Fix some $\mathbf{h}$
- Solve $\mathbf{B}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}} = \mathbf{h}$
- Determine $\mathbf{z} = \mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$
- Count $\mathbf{z}$'s: $\leq \lfloor M/(M-q) \rfloor$
- Maximum success probability: $\dfrac{\lfloor M/(M-q) \rfloor}{|Y|}$

To beat random guessing, need $\mathbf{q} \geq M/2$

To answer perfectly, need $\mathbf{q} \geq M\ (1 - 1/|Y|)$

Generalizes [FGGS'09,BBCdW'01], improves [MP'11]

Quantum Polynomial Interpolation

For a random degree-**d** polynomial **P** over **Y**, find **P**

- **B** is empty
- $\mathbf{C}(\bar{\mathbf{x}})$ are Vandermonde matrices

$$\mathbf{C}(\bar{\mathbf{x}}) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_q \\ \vdots & \vdots & & \vdots \\ x_1^d & x_2^d & \cdots & x_q^d \end{pmatrix}$$

- Goal: count vectors of form $\mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$

Quantum Polynomial Interpolation

For a random degree-$\mathbf{d}$ polynomial $\mathbf{P}$ over $\mathbf{Y}$, find $\mathbf{P}$

- Goal: count vectors of form $\mathbf{C}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{r}}$
- Easy upper bound:

$$\binom{|Y|}{q} |Y|^q$$

- Turns out, essentially tight

$$P_{quantum} \approx \binom{|Y|}{q} \Big/ |Y|^{d+1-q}$$

Quantum Polynomial Interpolation

For a random degree-$\mathbf{d}$ polynomial $\mathbf{P}$ over $\mathbf{Y}$, find $\mathbf{P}$

$$P_{quantum} \approx \binom{|Y|}{q} \Big/ |Y|^{d+1-q}$$

Think $|Y| \gg q \Rightarrow P_{quantum} \approx |Y|^{2q-d-1}/q!$

- $q > (d+1)/2$: success probability close to $\mathbf{1}$
- $q < (d+1)/2$: success probability close to $\mathbf{0}$
- $q = (d+1)/2$: success probability close to $\mathbf{1/q!}$

Proved concurrently by [CvDHS'15]

Degree **d** Polys as **q**-time MACs

Find $(P(t_0), \ldots, P(t_q))$

- $B = \{P$ such that $P(t_0) = \ldots = P(t_q) = 0\}$

  Let R(x) be the degree-$(q+1)$ monic polynomial with roots at $\{t_0, \ldots, t_q\}$

$$B(\bar{x}) = \begin{pmatrix} R(x_1) & \cdots & R(x_q) \\ R(x_1)x_1 & \cdots & R(x_q)x_q \\ \vdots & & \vdots \\ R(x_1)x_1^{d-q-1} & \cdots & R(x_q)x_q^{d-q-1} \end{pmatrix}$$

- For upper bound, suffices to count solutions to $B(\bar{x}) \cdot \bar{r} = h$

Degree **d** Polys as **q**-time MACs

Find $(P(t_0), \ldots, P(t_q))$

- $B = \{P$ such that $P(t_0) = \ldots = P(t_q) = 0\}$
- For upper bound, suffices to count solutions to $B(\bar{x}) \cdot \bar{r} = h$
- If $q \leq d/2$, number of solutions bounded by:

$$(q+1)q\ e^{2\sqrt{q}}$$

- So success probability in breaking MAC:

$$\leq (q+1)q\ e^{2\sqrt{q}}/|Y| = \text{negligible}$$

- Thus, degree **2q** polynomials are good **q**-time quantum-secure MACs
  - Optimal, improves on **3q** required by [BZ'13]

High level takeaways…

Comparing Classical and Quantum

$$P_{quantum} = \underset{B,C,h}{MAX}\left(\frac{|\{C(\bar{x})\cdot\bar{r}:\ B(\bar{x})\cdot\bar{r} = h\}|}{|C|}\right)$$

$$P_{classical} = \underset{B,C,h,\bar{x}}{MAX}\left(\frac{|\{C(\bar{x})\cdot\bar{r}:\ B(\bar{x})\cdot\bar{r} = h\}|}{|C|}\right)$$

Where $\bar{x} \in X^q$, $\bar{r} \in Y^q$

Observation

Only modest quantum speedups for problems analyzed

Explanation:

- Quantum algorithms have much higher success probability (by a factor of up to **$|X|$^q**)
- But, success probability increases significantly every for every query made
- Don’t need many extra classical queries to compensate

Conclusion

Give complete solution to wide class of problems

Gain some level of intuition for why quantum queries help

Future direction:

Gain intuition for more general problems

Thanks!