MIT 6.543
Fall 2026

Quantum Complexity Theory

Instructor: Anand Natarajan

9/10/26

Lecture 1: Intro to BQP

- Complexity theory is about finding the mathematical limits of efficient computation

- Computers are physical objects, and efficiency is a physical notion (time, space, etc.)

So to understand complexity, need to use the right model of physics.. quantum mechanics

Mission #1 of quantum complexity theory:

Understand the power & limitations of quantum computers

– Not only are computers physical, many computational problems arise from physics as well.

Even natural processes can be thought of as computations.

Mission #2 of quantum complexity theory:

Use computation as a lens to understand nature!

The basic method of complexity theory is to classify problems into complexity classes typically defined by the computational resources needed to solve them

Recall $P = \{$ languages (= decision problems) solvable by poly-time classical algos $\}$

$BPP = \{$ " " " randomized classical algos w/ bounded error $\}$

if $x \in L$, $\Pr[M(x) = Yes] \geq 2/3$

$x \notin L$, $\Pr[M(x) = No] \leq 1/3$

} separated by a constant

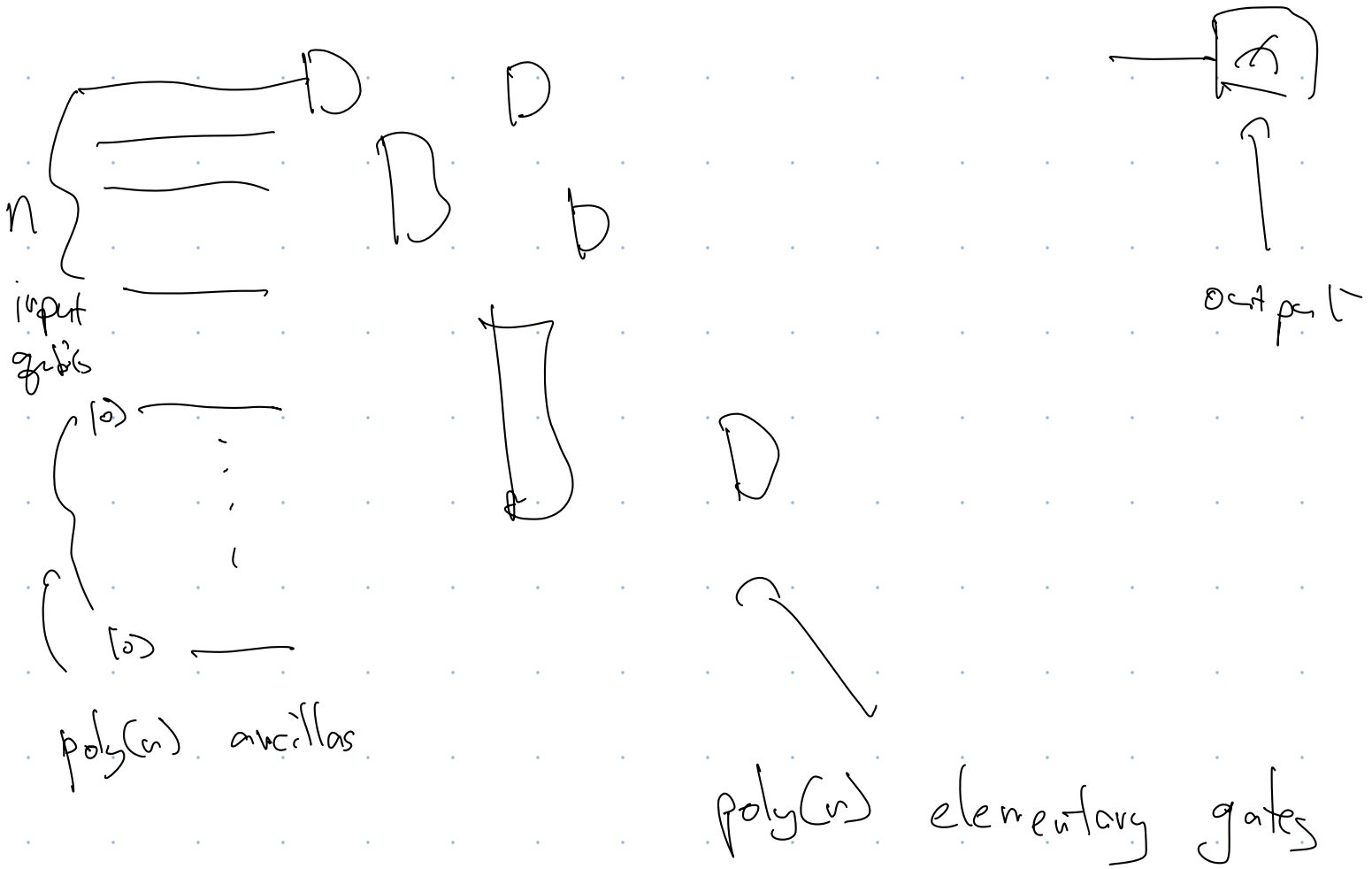
The star of quantum complexity theory is

$BQP = \{$ languages solvable by a poly-time b.e. quantum alg $\}$

what does this mean?

Classically, poly-time defined using Turing machines.

Quantumly, easier to use <u>circuits</u>



- No need to allow for intermediate measurements (just simulate coherently w/ ancilla)

- Can pick any reasonable universal set of elementary gates

  (SK says you only pay a polylog(n) cost to switch)

- Need some notion of uniformity: can't have a totally different circuit for each n

  "∃ classical TM M that runs in poly time and on input 1^n, spits out a description of the circuit for inputs of length n"

Easy to see

$$P \subseteq BPP \subseteq BQP$$

↑ by definition

↑ simulate random coins using Hadamards

- Also easy to see BQP is closed under complement (switching YES & NO)
- The thresholds $2/3$, $1/3$ are arbitrary and can be amplified

Just run the circuit many times and check that the fraction of accepts is above a threshold

Hoeffding's: if $X_1, \dots, X_k$ are indep. r.v. in $[0,1]$

$$\Pr\left[\left|\sum_{i=1}^{k} X_i - \mathbb{E}\sum_{i=1}^{k} X_i\right| \geq t\right) \leq 2e^{\frac{-2t^2}{k}}$$

So we can get $1 - \exp(-k)$ vs $\exp(-k)$ w/ k repetitions

One nice property of P & BPP is that they "respect subroutines": a P machine given access to an oracle deciding another P language is equiv to a P machine without the oracle

The same is true for BQP, but proving it requires some care

How does a q. circuit access an oracle? for L

$$U_L |x\rangle|b\rangle = \begin{cases} |x\rangle|b\rangle & \text{if } x \notin L \\ |x\rangle|b \oplus 1\rangle & \text{if } x \in L \end{cases}$$

?

unitary!

Suppose L is decided by a BQP machine.

The circuit gives some unitary U

but it is not this one!

$$U|x\rangle|0^{poly(n)}\rangle = \alpha|0\rangle|\varphi_0\rangle + \beta|1\rangle|\varphi_1\rangle$$

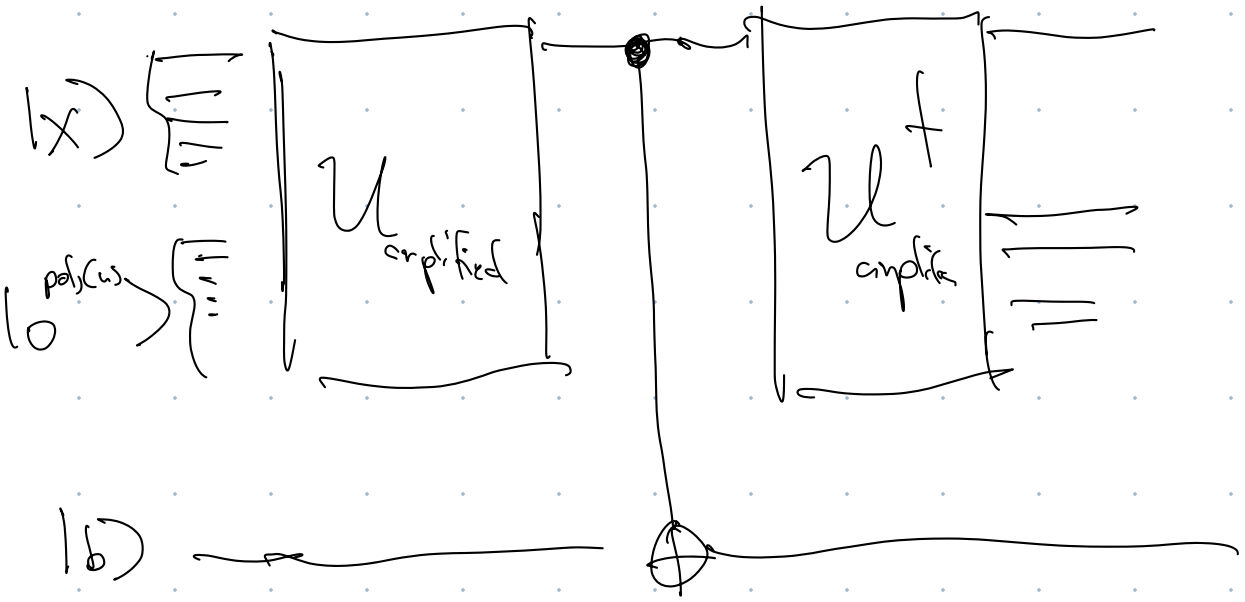
$|\beta|^2 \geq 2/3$ if $x \in L$

$\leq 1/3$ if $x \notin L$

can amplify to be exp close to 0 & 1

The real issue is the "junk" $|\varphi_0\rangle, |\varphi_1\rangle$

Solution: "uncompute"



This implements U_L as desired (except that we need ancilla, and there's some tiny error from amplification)

We have shown that

BQP

BQP = BQP

What about classical upper bounds on BQP?

$BQP \overset{??}{\subseteq} BPP$ We hope not!!

$BQP \overset{??}{\subseteq} NP$ Probably not, stay tuned for evidence

$BQP \subseteq EXP$ ✓

Prove this w/ brute force simulation:

can calculate $\left| (|x\rangle\langle 0| \otimes I)\, U |x\rangle|0\rangle \right|^2$

in exp time

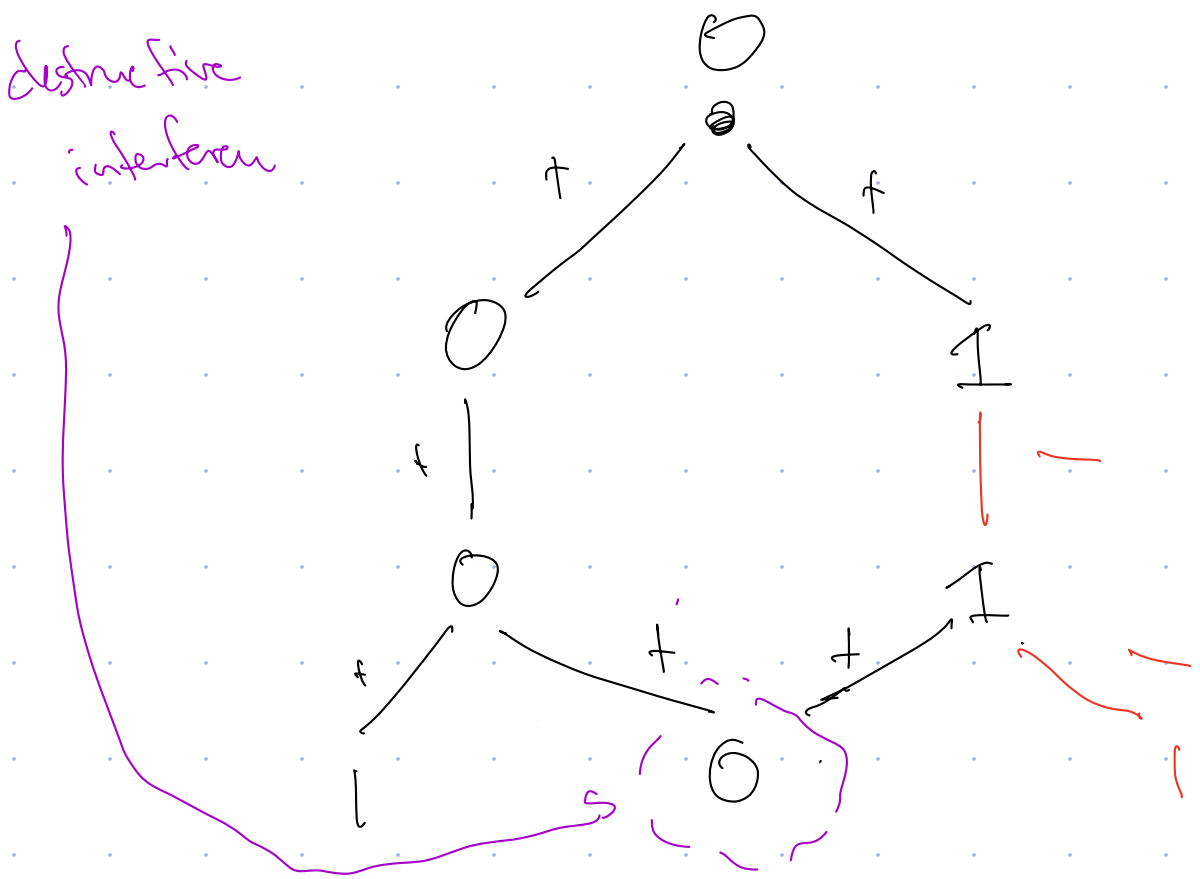
exp. big matrices and vectors

BQP ⊆ PSPACE

To show this, we need a new perspective: the "path integral"

Easiest to fix $\{H, CCZ\}$ as our universal set

destructive interference



In general, the state of the machine after a circuit with k Hadamard gates is

$\in \{\pm 1\}$

$$|\psi\rangle = \frac{1}{\sqrt{2^k}} \sum_{u \in \{0,1\}^n} \left(\sum_{\substack{\text{paths } p: \\ \text{end}(p) = u}} \text{sign}(p) \right) |u\rangle$$

can compute in poly time

$$\Pr[\text{accept}] = \frac{1}{2^k} \sum_{u \in \{0,1\}^{n-1}} \sum_{\substack{p, p': \\ \text{end}(p) = \text{end}(p') \\ = u}} \text{sign}(p)\,\text{sign}(p')$$

This is a big sum so it can be computed in PSPACE

$$BQP \subseteq PP$$

PP is like BPP without bounded error

if $x \in L$, $\Pr[\text{accept}] \geq 1/2$

if $x \notin L$, $\Pr[\text{accept}] < 1/2$

} no gap!

To show this, we use the path integral

$$\Pr[\text{accept}] = \frac{1}{2^k} \sum_{p, p' : \text{end}(p) = \text{end}(p')} \text{sign}(p)\,\text{sign}(p')\,(-1)^{\text{end}(p)_I}$$

$\Pr[\text{accept}] - \Pr[\text{reject}]$

β

$(-1)^{\text{end}(p)_I}$ ↑ output bit

PP machine needs to guess whether

β is $+$ or $-$.

Idea: just sample a random term in the sum!

PP alg:

- Sample p, p' uniformly at random
- If $\mathrm{end}(p) = \mathrm{end}(p')$:

  compute $S = \mathrm{sign}(p)\,\mathrm{sign}(p')\,(-1)^{\mathrm{end}(p)}$,

  if $S = +1$, output YES,

  else output NO.

Else:

output YES or NO uniformly at random.

$$\Pr[\text{accept}] = \Pr[\text{end}(p) \neq \text{end}(p')] \cdot$$

$$- \Pr[\text{reject}]$$

$$+ \Pr[\text{end}(p) = \text{end}(p')] \cdot$$ (prop. to. β)

Essentially this same calculation gives the tightest known classical upper bound on BQP

$$\text{BQP} \subseteq \text{AWPP}$$

↑ ugly def, look it up!

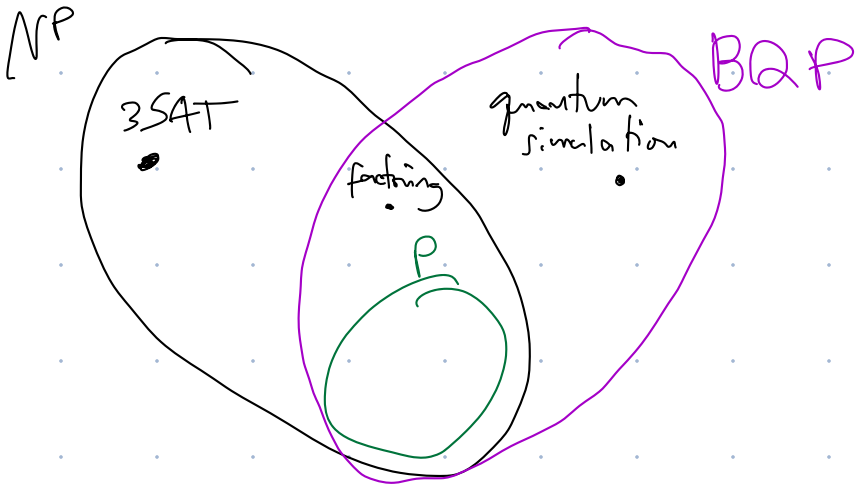
9/15/26

Lecture 2: BQP and NP, and the BBBV lower bound for unstructured search

Understanding the relationship between BQP & NP is of great importance

- If $BQP \stackrel{?}{\subseteq} NP$, then every quantum computation has an efficiently checkable certificate
- If $NP \stackrel{?}{\subseteq} BQP$, then all NP-complete problems (e.g. 3SAT, 3 coloring, TSP) can be solved by quantum computers!

We currently believe that neither is true.



Today we'll focus on evidence that $NP \not\subseteq BQP$.

What we will show is an oracle separation:

$\exists$ oracle O s.t.

$$NP^O \not\subseteq BQP^O$$

or unpacking this, there is some language L s.t. $L \in NP^O$ but $L \notin BQP^O$

L: related to O

$\notin$: provably

We can hope to prove this because deciding L requires some information about entries of O. So we will reduce the problem to the setting of "query complexity".

Note: An oracle separation does not imply that the classes are separated in reality!

$\exists$ oracles s.t. $P^O \neq NP^O$ and also

oracles s.t. $P^O = NP^O$

What it means is that any proof that settles P vs NP (or BQP vs NP) must be "non-relativizing": it must use techniques that break down in the presence of oracles.

Query complexity of search:

An oracle is a "magic box" that decides some language in unit cost.

Given input $x \in \{0,1\}^*$, O tells you "YES or NO"

Alternate perspective: The truth table of O is a computational input that can only be accessed by querying O on inputs $u \in \{0,1\}^*$

↑ index into truth table

This motivates the model of query complexity

- A "Query P" machine, given access to some input $X \in \{0,1\}^{2^n}$ (i.e. $N := 2^n$) should decide if $X \in S_{YES}^{(n)}$ or $X \in S_{NO}^{(n)}$ using $\text{poly}(n)$ queries to X ← only count queries, not runtime!
- A "Query BQP" machine is the same, but gets to make quantum queries

= apply $U_X : U_X|u\rangle|b\rangle \mapsto |u\rangle|b \oplus x_u\rangle$

A "Query NP" machine also gets a witness $w \in \{0,1\}^{poly(n)}$

- If $x \in S_{YES}^{(n)}$, $\exists$ w s.t. $M(w)$ outputs YES
- If $x \in S_{NO}^{(n)}$, $\forall w$, $M(w)$ outputs NO

It turns out that to find an oracle separation between BQP & NP, it will be enough to find a query promise problem specified by sets $S_{YES}^{(n)}$ & $S_{NO}^{(n)}$ for each n, s.t. a Query NP machine can solve it but a Query BQP machine cannot

Aside: a "promise problem" is a decision problem where the input is promised to come from some set. E.g. here, we only care about $x \in S_{YES}^{(n)} \cup S_{NO}^{(n)}$, not arbitrary $x \in \{0,1\}^n$

This "grey separation" can be lifted by very standard diagonalization arguments to an oracle separation $NP^O \not\subseteq BQP^O$, so we'll focus on the grey setting.

A candidate problem: Unstructured Search

NP is all about "finding a needle in a haystack." Think about SAT: want to find an input that makes the formula evaluate to 1, not 0.

Inspired by this, define the following problem

$$S_{YES}^{(n)} = \{ \text{all } x \in \{0,1\}^{2^n} \text{ except } 0^{2^n} \}$$

$$S_{NO}^{(n)} = \{ 0^{2^n} \}$$

Basically, want to decide if $\exists\, u \in \{0,1\}^n$ s.t. $x_u = 1$

& analogous to evaluating a SAT formula Φ at u

NP containment

Kind of obvious: a good witness is some w s.t. $\chi_w = 1$!

P lower bound:

Fix some Query P machine M.

If M solves Unstructured search, then

M must reject $x = 0^{2^n}$.

Let $u_1, u_2, \ldots, u_T$

↑ total # of queries

be the positions that M queries on

input 0^{2^n}. Since T is poly(n), it must be less than 2^n, so there is some location u' that has not been queried.

$x = 0\,0\,0 \ldots 0\,0\,0 \ldots 0\,0\,0 \ldots 0$

$u_1 \quad u_3 \quad u' \quad u_2 \quad u_T$

$x' = 0\,0\,0 \ldots 0\,1\,0 \ldots 0\,0\,0$

flip the value at u'

Now, $x' \in S_{YES}$, but M must still output NO on x', because x & x' look

identical to M!

Therefore, M gets x' wrong. 

In fact, this argument applies to any deterministic machine that makes fewer than 2^n queries. Can even extend to randomized machines (see pset 1!)

BQP lower bounds

None of the ideas from the P case seem to work!

Problem 1: quantum algs. query in superposition. Even 1 query can "see" the entire input x

$$\frac{1}{\sqrt{2^n}} \sum_{u \in \{0,1\}^n} |u\rangle |0\rangle \xrightarrow{U_x} \frac{1}{\sqrt{2^n}} \sum_{u \in \{0,1\}^n} |u\rangle |x_u\rangle$$

We know this state can sometimes reveal a lot of information about x! E.g. the Bernstein Vazirani alg uses this to learn an entire linear function in one query.

Problem 2: specifically for unstructured search, Grover showed we can do way better than $\Omega(2^n)$ queries.

The # of queries needed is

$$O(\sqrt{2^n})$$

It turns out we can get around these problems and show that Grover's alg. is essentially optimal, by some more careful accounting.

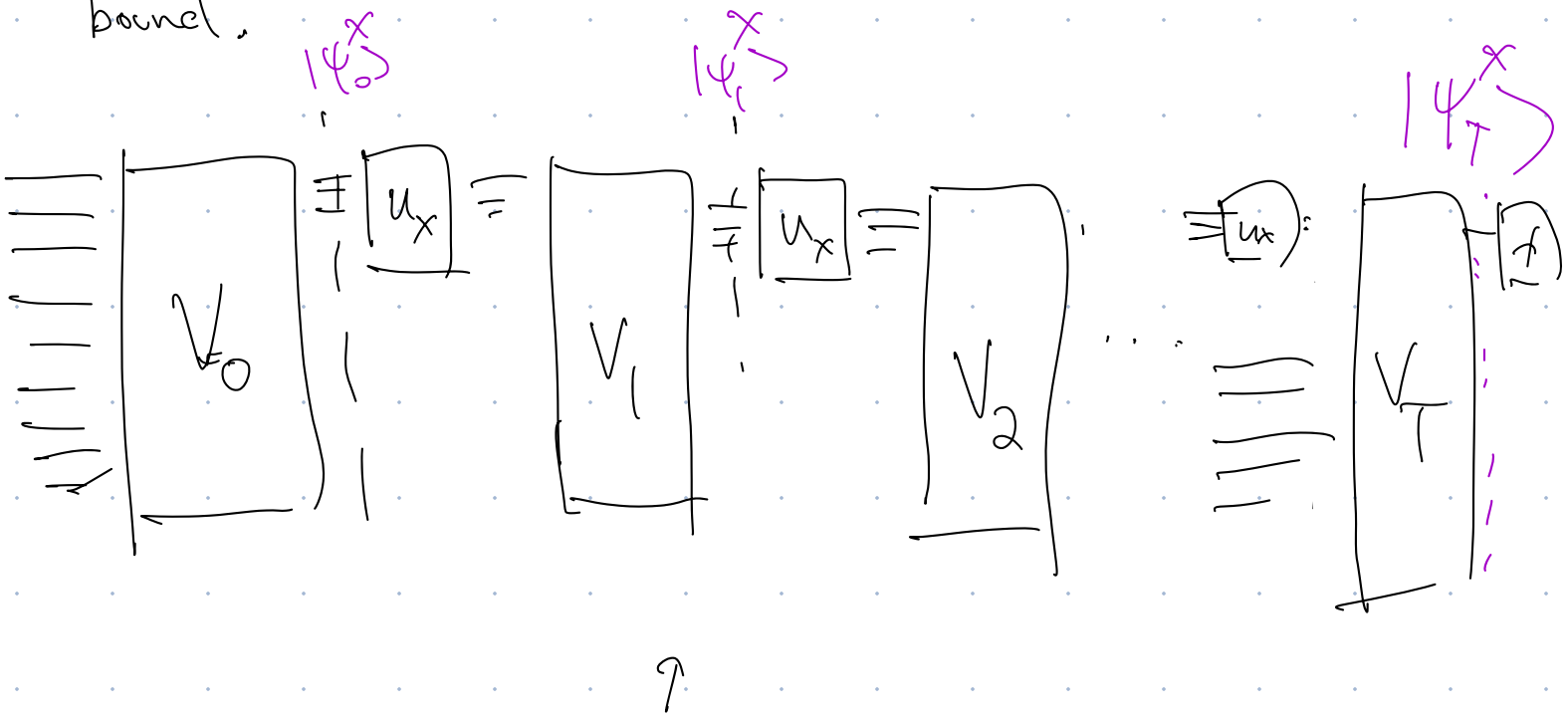
This is the

older than Grover's alg!

B '94 B B V

Bennett Bernstein Bassard Vazirani

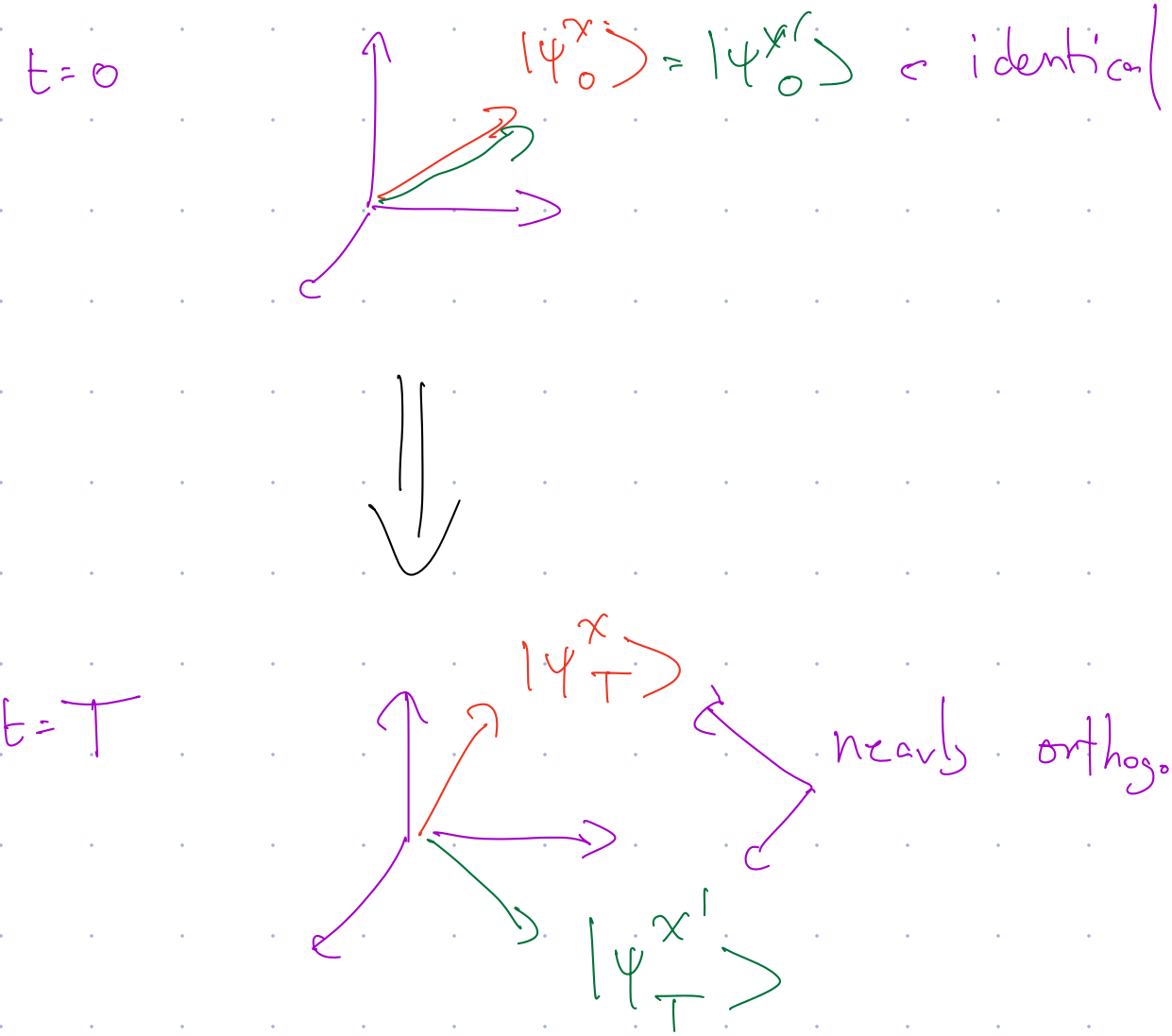
bound.



A general circuit making T queries.

Idea: For different x, x', track how far apart $|\psi_t^x\rangle$ and $|\psi_t^{x'}\rangle$ are

If $x \in S_{NO}$, $x' \in S_{YES}$, need



If a quantum alg. can pull apart the two states by only a little bit in Euclidean distance with each query, then T must be large for any successful quantum alg.

More precisely:

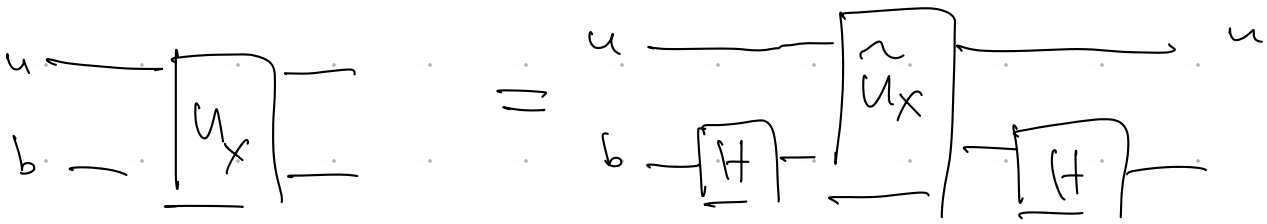
First, a slight change to the oracle:

instead of

$$U_x |u\rangle|b\rangle = |u\rangle|b + x_u\rangle,$$ work with

$$\tilde{U}_x |u\rangle|b\rangle = (-1)^{x_u \cdot b} |u\rangle|b\rangle$$

This is totally equivalent



Let $|\psi\rangle = \sum_{u,b} \alpha_{u,b} |u\rangle|b\rangle |\varphi_{u,b}\rangle$

oracle acts here only

We want to compute

$$\| \tilde{U}_x |\psi\rangle - \tilde{U}_{x'} |\psi\rangle \|_2$$

$$= \left\| \sum_{u,b} \alpha_{u,b} |u\rangle \left((-1)^{b \cdot x_u} - (-1)^{b \cdot x'_u} \right) |b\rangle |\varphi_{u,b}\rangle \right\|_2$$

$$= \left\| \sum_{u : x_u \neq x'_u} \pm 2 \alpha_{u,1} |u\rangle |1\rangle |\varphi_{u,1}\rangle \right\|_2$$

$$= 2 \sqrt{\sum_{u : x_u \neq x'_u} |\alpha_{u,1}|^2}$$

So to pull the states apart, the alg. must have high "superposition weight" on positions where x, x' differ.

We can now show our lower bound by a hybrid argument + induction

$$\Delta_t^{x,x'} :=$$

$$\| |\psi_t^x\rangle - |\psi_t^{x'}\rangle \|_2 = \| V_t \tilde{u}_x |\psi_{t-1}^x\rangle - V_t \tilde{u}_{x'} |\psi_{t-1}^{x'}\rangle \|_2$$

$$= \| \hat{u}_x |\psi_{t-1}^x\rangle - \hat{u}_{x'} |\psi_{t-1}^{x'}\rangle \|_2$$

$$= \| \hat{u}_x |\psi_{t-1}^x\rangle - \hat{u}_{x'} |\psi_{t-1}^x\rangle + \hat{u}_{x'} |\psi_{t-1}^x\rangle - \hat{u}_{x'} |\psi_{t-1}^{x'}\rangle \|_2$$

$$\leq \| (\hat{u}_x - \hat{u}_{x'}) |\psi_{t-1}^x\rangle \|_2 \quad := \eta_t^{x,x'}$$

triangle ineq.

$$+ \| \tilde{u}_{x'} (|\psi_{t-1}^x\rangle - |\psi_{t-1}^{x'}\rangle) \|_2$$

$$= \Delta_{t-1}^{x,x'}$$

Claim: Let $x = 0^N$, & let μ be the uniform distribution over weight-1 strings in $\{0,1\}^N$. Then

$$\mathop{\mathbb{E}}_{x' \sim \mu} \eta_t^{x,x'} \leq \frac{2}{\sqrt{N}} \quad \forall t$$

Pf: Let $|\psi_{t-1}^x\rangle = \sum \alpha_{u,b} |u\rangle |b\rangle |\varphi_{u,b}\rangle$

By our calculation above,

$$\mathop{\mathbb{E}}_{x' \sim \mu} \eta_t^{x,x'} = \mathop{\mathbb{E}}_{u \in [N]} 2 |\alpha_{u,1}|$$

index where $x'_u = 1$

$$= \frac{2 \sum_{u \in [N]} |\alpha_{u,I}|}{N}$$

$$= \frac{2}{N} \left\langle \vec{\alpha}, \overrightarrow{\mathrm{sgn}(\alpha)} \right\rangle$$

$$\begin{pmatrix} \alpha_{0,1} \\ \alpha_{1,1} \\ \alpha_{2,1} \\ \vdots \\ \alpha_{N,1} \end{pmatrix} \qquad \begin{pmatrix} \mathrm{sgn}(\alpha_{0,1}) \\ \mathrm{sgn}(\alpha_{1,1}) \\ \vdots \\ \mathrm{sgn}(\alpha_{N,1}) \end{pmatrix}$$

entries are ± 1

Cauchy-Schwarz

$$\leq \frac{2}{N} \cdot \|\vec{\alpha}\|_2 \cdot \|\overrightarrow{\mathrm{sgn}(\alpha)}\|_2$$

normalization of state

$$\leq \frac{2}{N} \cdot 1 \cdot \sqrt{N} = \frac{2}{\sqrt{N}}$$

Now, putting it all together, we have

$$\mathop{\mathbb{E}}_{x' \sim \mu} \Delta_T^{O^N, x'} \leq \mathop{\mathbb{E}}_{x' \sim \mu} \left(\eta_T^{O^N, x'} + \Delta_{T-1}^{O^N, x'} \right)$$

$$\leq \frac{2}{\sqrt{N}} + \mathop{\mathbb{E}}_{x' \sim \mu} \Delta_{T-1}^{O^N, x'}$$

$$\leq \frac{2}{\sqrt{N}} + \frac{2}{\sqrt{N}} + \mathop{\mathbb{E}}_{x' \sim \mu} \Delta_{T-2}^{O^N, x'}$$

$$\cdots$$

$$\leq \frac{2T}{\sqrt{N}} + \mathop{\mathbb{E}}_{x' \sim \mu} \Delta_0^{O^N, x'}$$

$$= 0$$

$\Rightarrow$ Any quantum alg. that solves unstructured search w/ bounded error needs $T = \Omega(\sqrt{N})$ queries!

9/17/26

Lecture 3A: The Polynomial Method

The BBBV method is hard to directly apply to most problems. Sometimes alternate perspectives get us much farther by exploiting <u>symmetry</u> in the problem.

A very powerful one is the **polynomial method**, which views an algorithm as a low-degree polynomial.

$\equiv [V_0] = [\Theta] = [V_1] = [\Theta] = [V_2] \ldots$

<u>Lemma:</u> After t steps, the state $|\psi_t\rangle = \sum \alpha_i |i\rangle$ (α_i: degree-t poly in the bits $x_1 \ldots x_N$ of the oracle)

<u>Pf:</u> By induction!

Base case: $t=0$, obvious

$t \to t+1$

$$O|\psi_t\rangle = O \sum_{i,b,z} \alpha_{i,b,z} |i\rangle|b\rangle|z\rangle$$

$$= \sum_{i,b,z} (-1)^{b \cdot x_i} \alpha_{i,b,z} |i\rangle|b\rangle|z\rangle$$

$$= \sum_{i,b,z} (1 - 2bx_i)\, \alpha_{i,b,z} \quad |i\rangle|b\rangle|z\rangle$$

degree 1 · degree t

degree $t+1$ polynomial.

∎

Cor: $\Pr[\text{acceptance}^x] =$ degree $2T$ poly in $x_1 \dots x_N$ of T-query alg.

The 2 comes from squaring!

So if you can show that no low-degree poly matches the promise problem you want to decide, you show that there's no q. alg!

But it's very hard to reason about multivariate polynomials with lots of variables. So often one uses symmetry of some kind to reduce the number of variables.

The unstructured search problem:

Idea: consider distributions on inputs that are symmetric under permutations of x

$\mu_r :=$ uniform dist. over x s.t. $|x| = r$

(Hamming weight)

Claim: If $p(x_1, \dots, x_N)$ is degree d multivariate,

$q(r) := \mathop{\mathbb{E}}_{x \sim \mu_r} p(x_1, \dots, x_N)$ is degree d poly in r.

Pf: $\mathop{\mathbb{E}}_{x \sim \mu_r} x_1 x_3 x_{17} = \Pr_{x \sim \mu_r}[\text{all 3 of } x_1, x_3, x_{17} \text{ are equal to } 1]$

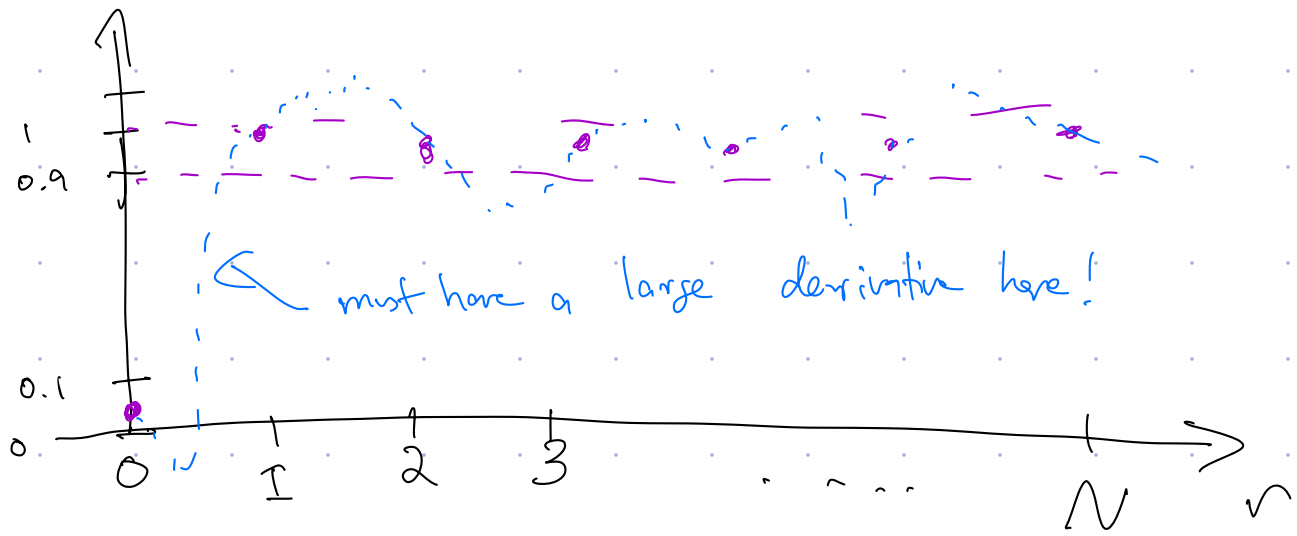
$$= \frac{\binom{N-3}{r-3}}{\binom{N}{r}}$$

$$= \frac{\frac{(N-3)!}{(r-3)!\,(N-r)!}}{\frac{N!}{r!\,(N-r)!}}$$

$$= \frac{r(r-1)(r-2)}{N(N-1)(N-2)}$$ ← degree 3 poly in r!

Easily generalizes to monomials of any degree.

Now, if we have an alg that solves unstructured search in T queries $\Rightarrow$ a degree-$2T$ polynomial q that looks like this



Note: no constraint on q outside of integer points (need not even be positive)

It turns out no low-degree poly can look like this!

Markov's other inequality:

$$\max_{-1 \le x \le 1} \left| \frac{dp(x)}{dx} \right| \le d^2 \max_{-1 \le x \le 1} |p(x)|$$

d^2: degree(p)

This immediately gives us a useful bound if q stays bounded at non-integer points.

If q gets large at non-integer points, then you get a large derivative in between integer points and you can still apply this.

Upshot: Need $\deg(q) \ge \Omega(\sqrt{N})$.

Getting the right symmetrization is a bit of an art!