Lecture: Quantum query complexity

What is the true source of quantum power?

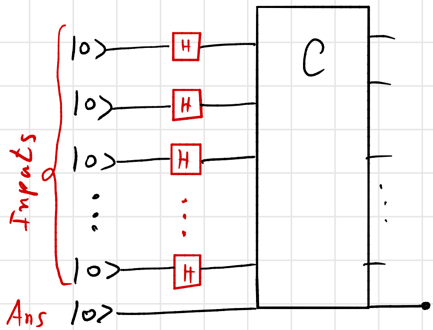
- Amplitude interference?
- Size of Hilbert space?
- Entanglement ?

Pairwise classically simulatable

My take: Query in superposition !

Let C be a quantum circuit implementing $f:\{0,1\}^n \to \{0,1\}$

i.e., $C\big(|x\rangle \otimes |0\rangle\big) = |x\rangle \otimes |f(x)\rangle$.



$$\left(\tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle\right) \otimes \cdots \otimes \left(\tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle = \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |0\rangle$$

$$\downarrow$$

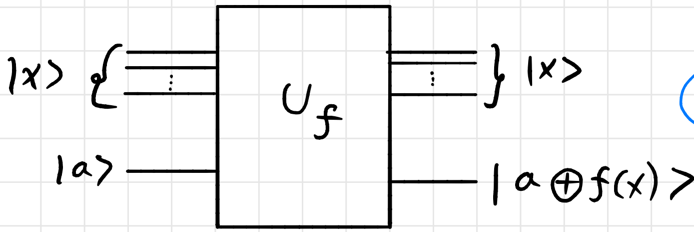
$$\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |f(x)\rangle$$

Deutsch - Simon

Quantum Query Complexity

- Guest
- Assignment
- Approach

We are given $f: \{0,1\}^n \to \{0,1\}$ as a black-box



Superposition!

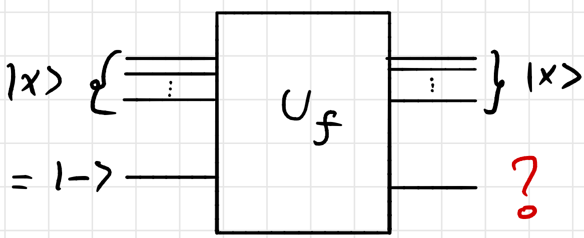
Usually $|a\rangle = |0\rangle \Rightarrow |a \oplus f(x)\rangle = |f(x)\rangle$

Goal: Learn about f (formally decide $f \in S$) using as little as possible queries.

why? implicit representation, sublinear-time

Phase queries:

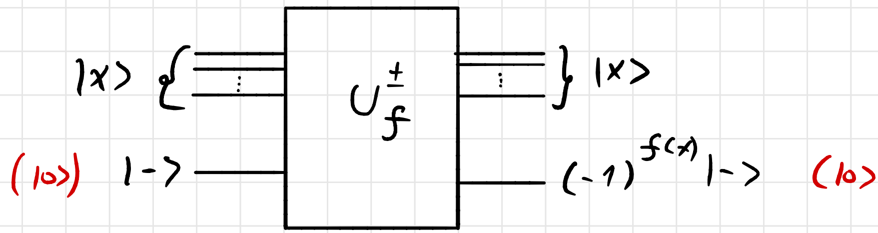
$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle$



$$\frac{1}{\sqrt{2}}|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|1 \oplus f(x)\rangle = \underbrace{\frac{1}{\sqrt{2}}|f(x)\rangle - \frac{1}{\sqrt{2}}|\neg f(x)\rangle}_{\substack{|-\rangle \text{ if } f(x)=0 \\ -|-\rangle \text{ if } f(x)=1}}$$

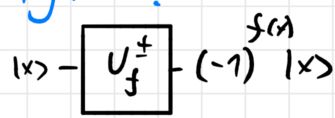
$$= (-1)^{f(x)}|-\rangle$$

In summary:



So $|x\rangle \otimes \overset{|0\rangle}{|-\rangle} \longrightarrow (-1)^{f(x)} |x\rangle \otimes \overset{|0\rangle}{|-\rangle}$

The $|-\rangle$ is just an ancilla – ignore.



Finally, in superposition:

- Standard queries: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |f(x)\rangle$

- Phase queries: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$

This is where the magic is happening!

But how can we use it? info is stuck in the amplitudes...

Rediscovering the first quantum algorithm

$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

Typically we teach: Problem, algorithm, analysis

This isn't how research always works...

Instead, we will try to understand the notion!

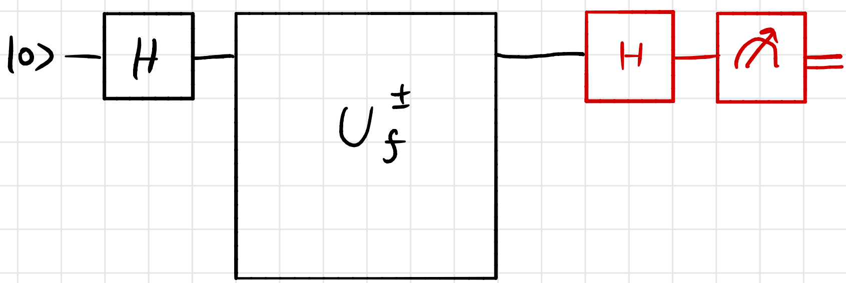
Goal: Understand the power of computing in Superposition.

This is amazing:

$$\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$$

But complicated...

Simplify: Consider $n = 1$. Let $f: \{0,1\} \to \{0,1\}$



$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \rightarrow \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right]$$

Cool, but how is that useful?

$$= \frac{(-1)^{f(0)}}{\sqrt{2}} \left[|0\rangle + (-1)^{f(1)-f(0)} |1\rangle \right]$$

Observation:

Global phase is indisting.

- If $f(0) = f(1) \rightsquigarrow \frac{(-1)^{f(0)}}{\sqrt{2}} \left(|0\rangle + |1\rangle \right) \simeq |+\rangle$
- If $f(0) \neq f(1) \rightsquigarrow \frac{(-1)^{f(0)}}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) \simeq |-\rangle$

Apply Hadamard: If $f(0) = f(1)$, we get $|0\rangle$

If $f(0) \neq f(1)$, we get $|1\rangle$

This is same as $XOR(f(0), f(1))$

<u>Deutsch's algorithm:</u> computes the parity (XOR) of two bits using 1 query. "Sublinear" effect

Classically, this requires 2 queries.

The Deutsch-Jozsa Algorithm

Recall that Deutsch's algorithm computes the parity (XOR) of two bits using 1 query.

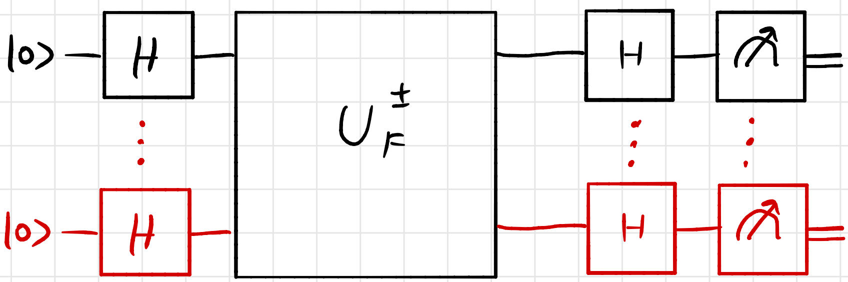
We looked at 2 bits for simplicity.

$\boxed{\text{Can we generalise to } n \text{ bits?}}$

Black-box approach: Compute $\bigoplus_{x \in \{0,1\}^n} F(x)$ by breaking into $\frac{2^n}{2}$ blocks.

Nice, but not great...

White-box approach:



Time 0: $|0\rangle^{\otimes n} = |0\rangle \otimes \cdots \otimes |0\rangle$ (Notation)

Time 1: $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$

($H^{\otimes n}$ as uniform superposition)

Time 2: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} |x\rangle$ (loading the func.)

Time 3: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} \underline{\underline{H^{\otimes n} |x\rangle}}$

We need a Fourier analysis interlude...

Boolean Fourier Transform (Hadamard transform)

What's $H^{\otimes n}|x\rangle$? (we know $H^{\otimes n}|0\rangle$)

Recall: $H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

Trick: For $b \in \{0,1\}$, write $H|b\rangle = \frac{|0\rangle + (-1)^b |1\rangle}{\sqrt{2}}$.

So, $H^{\otimes n}|x\rangle = \frac{|0\rangle + (-1)^{x_1}|1\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|0\rangle + (-1)^{x_n}|1\rangle}{\sqrt{2}}$

$$= \frac{1}{2^{n/2}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

where $x \cdot y = \sum x_i y_i \pmod 2) = \bigoplus_{i : x_i = 1} y_i = XOR_x(y)$

Later: Why is that a Fourier transform?

Now we can go back to Deutsch-Jozsa

Time 3: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} \underline{\underline{H^{\otimes n} |x\rangle}}$

$$\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} \cdot \frac{1}{2^{n/2}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

How to deal with that?

Shortcut: what is the amplitude of $|00\cdots0\rangle$?

$$\alpha := \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{F(x)}$$

Observation: If F is const ($F(x) = b \ \forall x$) $\Rightarrow \alpha \in \{-1, 1\}$.

If F is balanced ($\sum F(x) = \frac{2^n}{2}$) $\Rightarrow \alpha = 0$

Thus, if F const., we'll always measure $|0\rangle$

, if F bal., we'll never measure $|0\rangle$

Deutsch-Jozsa problem: distinguish between constant and balanced functions using 1 query (w.p. 1!)

Digest (Deutsch-Jozsa)

Recall that Deutsch's algorithm computes the parity (XOR) of two bits using 1 query.

We wanted to generalise to parity of n bits.

Blackbox doesn't scale gracefully.

Whitebox gave us something entirely different:

The Deutsch-Jozsa alg. distinguishes between constant and balanced functions using 1 query (w.p. 1!)

Classically, this requires $\frac{2^n}{2}+1$ (det) queries.

But with randomness, suffices $O(1)$ queries to distinguish w.h.p. We want more!

What about XORing n-bits: requires $\frac{n}{2}$ queries!

The Bernstein-Vazirani Algorithm

We can't use superposition to query n bits

using 1-query (or even $o(n)$ queries)

Next: The first quantum alg. that does exp

better than any (randomised) classical alg.

Problem: Let $F: \{0,1\}^n \to \{0,1\}$ s.t. $F(x) = S \cdot X \pmod 2$

for $S \in \{0,1\}^n$.

Given phase-queries to F, find S.

$\chi_S(x) = (-1)^{S \cdot X} = \mathrm{Parity}^{\pm}_S(X) =$ "character"

(element of the Fourier basis for bits)

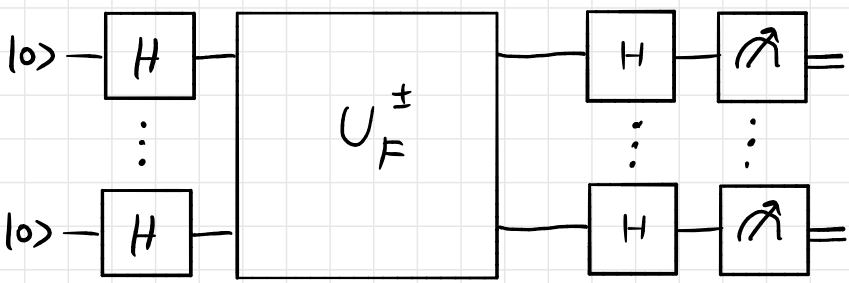
XOR-patterns

Example: $S = (\overset{x_1}{0}, \overset{x_2}{1}, \overset{x_3}{0}, \overset{x_4}{1}, \overset{x_5}{1})$, $F(x) = x_2 \oplus x_4 \oplus x_5$

Classically: requires n queries:
$$\begin{cases} F(10000) = s_1 \\ F(01000) = s_2 \\ F(00100) = s_3 \\ F(00010) = s_4 \\ F(00001) = s_5 \end{cases}$$

Since each query only provides 1-bit of info about s, we cannot do better!

The Bernstein-Vazirani alg.



Wait, what? This seems familiar...

$$H^{\otimes n}\left(\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{F(x)} |x\rangle\right) = H^{\otimes n}\left(\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle\right)$$

since $F(x) = s \cdot x$, for $s \in \{0,1\}^n$.

Let figure this out!

$$H^{\otimes n}\left(\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{s\cdot x}|x\rangle\right)$$

$$= H^{\otimes n}\left(\frac{|0\rangle+(-1)^{s_1}|1\rangle}{\sqrt{2}}\otimes\dots\otimes\frac{|0\rangle+(-1)^{s_n}|1\rangle}{\sqrt{2}}\right)$$

If $s_i = 0 \Rightarrow \frac{|0\rangle+(-1)^{s_i}|1\rangle}{\sqrt{2}} = |+\rangle$

$$H^{\otimes n}\left(\frac{|0\rangle+(-1)^{s_i}|1\rangle}{\sqrt{2}}\right) = |0\rangle$$

If $s_i = 1 \Rightarrow \frac{|0\rangle+(-1)^{s_i}|1\rangle}{\sqrt{2}} = |-\rangle$

$$H^{\otimes n}\left(\frac{|0\rangle+(-1)^{s_i}|1\rangle}{\sqrt{2}}\right) = |1\rangle$$

$$= |s\rangle \; !$$

Measure, and we get $|s\rangle$ w.p. 1.

Interference.

Next: This was all Fourier analysis!

Convenient Notation

Last time

GHZ

Function-vector duality: $g: D^n \to R \simeq v \in R^{D^n}$,

where $v = (g(\alpha_1), g(\alpha_2), \dots, g(\alpha_{|D^n|}))^\dagger$.

Example:

$x_1\ x_2$	$AND(x_1, x_2)$
0 0	0
0 1	0
1 0	0
1 1	1

$$v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

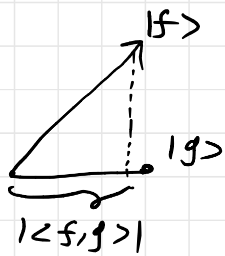
So $\sum |g(x)|^2 = 1$

Def: Let $g: \{0,1\}^n \to \{-1,1\}$. $|g\rangle := \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} g(x) |x\rangle$

Note: $\langle f | g \rangle = \frac{1}{2^{n/2}} (f(\bar{0}) \cdots f(\bar{1}))^* \cdot \frac{1}{2^{n/2}} \begin{pmatrix} g(\bar{0}) \\ \vdots \\ g(\bar{1}) \end{pmatrix}$

$$= \frac{1}{2^n} \sum_x f(x) \cdot g(x)$$

Geometrically, this is a projection:



Next: Fourier analysis in parallel universes !

Boolean Fourier transform

Just a change of basis: standard $\to$ XOR.

The Fourier basis over $GF(2^n)$ (Bits...)

is $\{\chi_s\}_{s\in\{0,1\}^n}$, where $\chi_s(x) = (-1)^{s\cdot x}$. (Parity)

Fourier expansion: Let $f:\{0,1\}^n \to \{-1,1\}$. Then

$$f(x) = \sum_{s\in\{0,1\}^n} \hat{f}(s)\cdot \chi_s(x).$$

where $\hat{f}(s) = \langle f | \chi_s\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)\cdot \chi_s(x).$

(Simply the projection!)

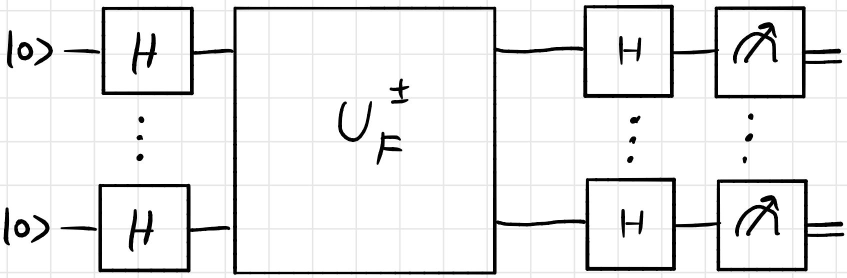
Fourier transform: $|f\rangle \underset{H^{\otimes n}}{\overset{H^{\otimes n}}{\rightleftarrows}} |\hat{f}\rangle$

Decomposing Boolean functions to XORs

Analogously to decomp. signals to sine & cosines!

Fourier Sampling

Let's distill the idea behind D-J & B-V.



$$\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{F(x)} \, \underline{\underline{H^{\otimes n}|x\rangle}}$$

$$= \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{F(x)} \cdot \frac{1}{2^{n/2}} \sum_{s\in\{0,1\}^n} (-1)^{x\cdot s} |s\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} f(x) \cdot \frac{1}{2^{n/2}} \sum_{s\in\{0,1\}^n} \chi_s(x) |s\rangle$$

$$= \sum_{s\in\{0,1\}^n} \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)\cdot\chi_s(x) \right) |s\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{s\in\{0,1\}^n} \langle f|\chi_s\rangle |s\rangle = \sum_{s\in\{0,1\}^n} \hat{f}(s)|s\rangle$$!!!

Since the final state is $\sum_{s\in\{0,1\}^n} \hat{f}(s)|s\rangle$,

When measuring (in comp. basis):

$\forall s\in\{0,1\}^n \quad Pr[|s\rangle] = |\hat{f}(s)|^2$!

By Parseval's Theorem: $\sum_{s\in\{0,1\}^n} |\hat{f}(s)| = \langle f, f\rangle = 1$

All algorithm we saw so far simply sampled from the Fourier spectrum!

<u>Deutsch-Josza:</u> Balanced vs constant

Note $\hat{f}(0) = \langle f | \chi_0 \rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} f(x)\cdot\chi_0(x) = \mathbb{E}[f]$

- If Balanced: $\mathbb{E}[f]=0 \Rightarrow \hat{f}(0)^2 = 0$

- If Constant: $\mathbb{E}[f] = \pm 1 \Rightarrow \hat{f}(0)^2 = 1$

<u>Berstein-Vazirani:</u>

If $f = \chi_s$, then $\hat{f}(s) = 1$ (since $f = \sum \hat{f}(s)\cdot\chi_s$)