Deutsch-Jozsa & Simon's Algorithms.

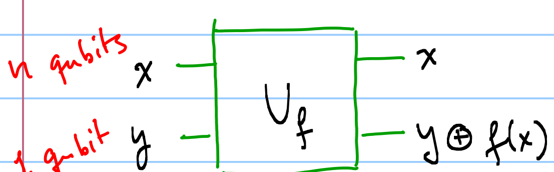
Note Title 4/13/2012

We saw last time that proving BPP $\neq$ BQP would imply that P $\neq$ PSPACE. Since the latter is a long-standing open question, we will set our sights on more modest goals and try and prove that quantum computers are more powerful than classical ones assuming that both have access to an oracle.

In an oracle model, there is some boolean function $f: \{0,1\}^n \longrightarrow \{0,1\}$. A computer can query the oracle by providing an input x and the oracle responds with $f(x)$. We want to solve some problem with this model — usually this involves discovering something about the function f.

The computer has only black box access to the function f in that the only way it can learn anything about f is to query the oracle as we have described. Usually we think of f as something difficult to compute so we evaluate our algorithm in terms of the number of queries to f required to solve the problem.

How would access to f look in a quantum model?
We assume that there is a function that takes
$|x, y\rangle$ to $|x, y \oplus f(x)\rangle$

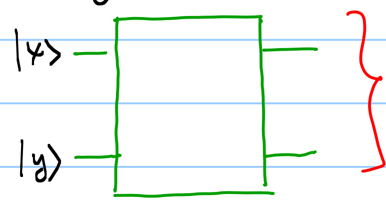


Note that defining access to f in this way ensures that U_f is unitary

The advantage that a quantum computer has over a classical one is that it can query f in superposition. In otherwords, it can query f on a superposition of multiple inputs and have that count as a single query. Of course the quantum algorithm is limited to how it can access this information by quantum measurement.

if $|\psi\rangle = \sum_x \alpha_x |x\rangle$

then



$\sum_x \alpha_x |x, y \oplus f(x)\rangle$

We will first examine the power of quantum computation under this query model. The following two examples have no particular practical applications but they are relatively simple examples illustrating that the quantum model is strictly more powerful than the classical one. They also led the way to more interesting applications.

Deutsch-Jozsa

in this problem, we are promised that f is either

all 0 : $f(x) = 0 \quad \forall x$.

balanced : $f(x) = 1$ for exactly half of $x \in \{0,1\}^n$

the problem is to determine which one of these is the case.

Note that in the worst case a deterministic classical algorithm will require $2^n/2 + 1$ queries. A probabilistic classical algorithm that uses k queries will have error 2^{-k}.

* $$\boxed{\left(H^{\otimes n} \otimes I_{n+1}\right)|0\rangle^n|0\rangle = |+\rangle^n|0\rangle = \left[\tfrac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\right]^{\otimes n} \otimes |0\rangle = \frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle|0\rangle}$$

We will show a quantum algorithm that can solve this problem with no error using only two queries.

Start with $|0\rangle^n|0\rangle$

Apply H to the first n qubits to get: $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle|0\rangle$ *

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

Apply Z to the last qubit to get: $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle(-1)^{f(x)}$

Applying U_f again will erase the last qubit $(f(x)\oplus f(x)=0)$

$$\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle\underbrace{(-1)^{f(x)}|0\rangle}_{f \text{ recorded in phase.}}$$

that if

We will no longer use the last qubit, so we will just drop it for the rest of the proof.

Define $|\phi_f\rangle = \frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle$

note that if we have two functions, $f + f'$, one of which is 0 and one of which is balanced, then $\langle\phi_f|\phi_{f'}\rangle=0$. if we apply a Unitary U to each, then $U|\phi_f\rangle$ and $U|\phi_{f'}\rangle$ are still orthogonal. The unitary we will choose is again $H^{\otimes n}$.

If f is 0, then

$$H^{\otimes n} |\phi_f\rangle = H^{\otimes n} |+\rangle^{\otimes n} = |0\rangle^{\otimes n}$$

If f' is balanced then $H^{\otimes n} |\phi_{f'}\rangle$ is orthogonal to $|0 - 0\rangle$.
That is, when $H^{\otimes n} |\phi_{f'}\rangle$ is expressed in the standard basis, the amplitude of $|0 \cdots 0\rangle$ is 0.

So we apply $H^{\otimes n}$ to $|\phi_f\rangle$ and then measure in the standard basis. If the outcome is all 0's then f is 0. If the measurement yields $|1\rangle$ on any qubit, then f is balanced.

This example is unsatisfying because a randomized classical algorithm requires only $O(1)$ queries to distinguish between a 0 and balanced f with high probability. We now look at a similar problem with a richer class of functions that will give an exponential gap (with an oracle) between BQP and BPP.

Simon's Algorithm

We are given a function f such that $\exists\, a \in \{0,1\}^n$ such that $\forall x \quad f(x) = f(x \oplus a)$
and if $f(x) = f(y)$ then $x = y$ or $x = y \oplus a$.

So a defines a pairing between strings. Pairs have the same value and different pairs have different values.
The outcome of f is arbitrary besides this 2-1 property.
(Range is of size 2^{n-1} — # output bits $m \geq n-1$).

Start with $\overbrace{|0 \cdots 0\rangle}^{n} \overbrace{|0 \cdots 0\rangle}^{m}$

Perform $H^{\otimes n}$ on first n qubits to get: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |0 - 0\rangle$

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

Now measure last register: $\frac{1}{\sqrt{2}} \sum_{x : f(x) = d} |x\rangle |d\rangle$ d chosen at random from range of f.

$= \frac{1}{\sqrt{2}} \left(|z\rangle + |z \oplus a\rangle \right) |d\rangle$ z chosen at random.

Now perform $H^{\otimes n}$ again. (we can drop last m qubits).

What is $H^n |z\rangle$ $z \in \{0,1\}^n$ (standard basis state) ?

$$\bigotimes_{j=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{z_j} |1\rangle \right) = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} \prod_{j=1}^{n} (-1)^{z_j x_j} |x\rangle$$

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

$x \cdot z = \sum_{j=1}^{n} x_j z_j \mod 2$

$$H^n \left[\frac{1}{\sqrt{2}} \left(|z\rangle + |z \oplus a\rangle \right) \right] = \frac{1}{\sqrt{2}} \cdot \frac{1}{2^{n/2}} \left[\sum_x (-1)^{x \cdot z} |x\rangle + \sum_x \underbrace{(-1)^{x \cdot (z \oplus a)}}_{(-1)^{x \cdot z} (-1)^{x \cdot a}} |x\rangle \right]$$

$$= \frac{1}{\sqrt{2}} \frac{1}{2^{n/2}} \left[\sum_x (-1)^{x \cdot z} \left[1 + (-1)^{x \cdot a} \right] |x\rangle \right]$$

if $x \cdot a = 1$ amplitude of $|x\rangle$ is 0, if $x \cdot a = 0$ $\alpha_x = \pm 1/2^{(n-1)/2}$

If we measure x, we get x such that $x \cdot a = 0$
chosen randomly among all x that have this property.

If we repeat this process, we get a collection of x's:
$x_1, x_2, \ldots, x_e \quad : \quad x_j \cdot a = 0$.

We can use Gaussian elimination to solve for a.
How many x's do we need? There are 2^{n-1} x's that satisfy $x \cdot a = 0$. So the space of all such x has dim. $n-1$.
We need to have $n-1$ x's that are linearly independent to solve for a.

Given that we have $x_1 \cdots x_k$ that are linearly independent what's the probability that the next one will still be linearly independent?

There are 2^k vectors in the space spanned by $x_1 \cdots x_k$.
The probability we pick something outside this set is:
$$\frac{2^{n-1} - 2^k}{2^{n-1}} = 1 - \frac{1}{2^{n-k-1}}$$

The probability that we pick $(n-1)$ linearly independent vectors x is:
$$\underbrace{1 \cdot \left(1 - \frac{1}{2^{n-2}}\right) \cdots \left(1 - \frac{1}{4}\right)\left(1 - \frac{1}{2}\right)}_{\geq 1 - \underbrace{\sum_{k=2}^{n} \frac{1}{2^k}}_{\frac{1}{4} + \frac{1}{8} + \cdots \geq \frac{1}{2}} \geq \frac{1}{2}} \geq \frac{1}{4}$$

A little reminder about Gaussian elimination over $\mathbb{Z}_2$.

We have: $x_1 \cdot a = x_2 \cdot a = x_3 \cdot a = \dots x_{n-1} \cdot a = 0$

Matrix x_{ij} : bit j of x_i. $\quad X \vec{a} = \vec{0}$

$(n-1) \times n$ matrix.

Start with $r = 1$, $c = 1$.

LOOP: while $r \le n-1$, $c \le n$.

Find smallest $s \ge r$ s.t. $x_{sc} \ne 0$. — if no such s exists, $c \leftarrow c+1$, go back to LOOP.

Swap row r + row s.

For every r' s.t. $x_{r'c} \ne 0$, add row r to row r'.

$c \leftarrow c+1$, $r \leftarrow r+1$.

If all x's are linearly indep, we get:

$$\begin{array}{lllll}
a_1 & & 0/1\ a_j & & = 0 \\
 & a_2 & \vdots & & \vdots \\
 & \ddots & 0/1\ a_j & & \\
 & & a_{j-1} + a_j & & = 0 \\
 & & & a_{j+1} & = 0. \\
 & & & \ddots & \\
 & & & a_{n-1} & = 0
\end{array}$$

$a_{j+1} = \dots = a_n = 0$.

$a_{j-1} = a_j = 1$

for $k < j-1$

$a_k + a_j = 0 \Rightarrow a_k = 1$

$a_k + 0 \cdot a_j = 0 \Rightarrow a_k = 0$.

If this process ends with $c = n+1$ and $r \le n-2$ (i.e there is an all zeroes row) then $\exists\ b_1 \dots b_n$ $b_1 x_1 + \dots + b_n x_n = \vec{0}$ and the x's are not linearly independents.

How well can a classical (probabilistic) algorithm do?

Pick a at random. $a \in \{0,1\}^n - \{0^n\}$

Suppose that k queries have been made: $x_1, \ldots, x_k$.

What do we learn about a?

if $x_i \neq x_j$ and $f(x_i) = f(x_j)$ then $a = x_i \oplus x_j$.

Otherwise $a \neq x_i \oplus x_j$ for all $i \neq j$.

we have eliminated $\binom{k}{2}$ possibilities but otherwise learned nothing about a.

The other $2^n - 1 - \binom{k}{2}$ possibilities are all still equally likely since they are all consistent with the observed $f(x_1) \ldots f(x_k)$.

The next choice of x_{k+1} will reveal a if it happens that $a = x_{k+1} \oplus x_j$ for some $j \in \{1, 2, \ldots, k\}$

This happens with probability $\dfrac{k}{2^n - 1 - \binom{k}{2}}$

The probability that one of m choices for $x_1 \ldots x_m$ reveal a is at most

$$\sum_{k=1}^{m} \frac{k}{2^n - 1 - \binom{k}{2}} \leq \sum_{k=1}^{m} \frac{k}{2^n - k^2} \leq \frac{m^2}{2^n - m^2}$$

in order for this probability to be const, we need $m = \Omega(2^{n/2})$.