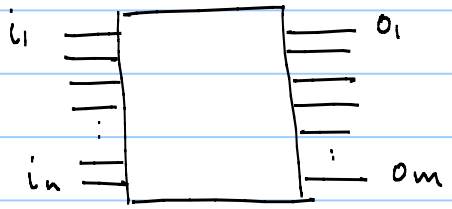
Quantum Complexity Classes

Note Title 4/10/2012

Let's first consider classical circuits that compute a function from n bits to m bits. (m could be larger than n, but is typically smaller than n).

$$f: \{0,1\}^n \longrightarrow \{0,1\}^m$$



the circuit is composed of gates from a finite set (e.g. OR, NOT, AND, etc)

A Universal Gate Set can be used to compute any boolean function. For example, the following list of sets are all universal:

$$\{\text{AND}, \text{NOT}\}, \{\text{OR}, \text{NOT}\}, \{\text{NAND}\}, \ldots$$

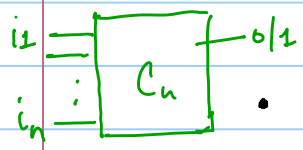
We are typically interested in decision problems in which the number of output bits is 1. $f: \{0,1\}^n \to \{0,1\}$. These can also be expressed as a language which is a subset of all binary strings $L \subseteq \{0,1\}^*$

$x \in \{0,1\}^*$: $f(x) = 1$ if and only if $x \in L$

Now let's define the class P (deterministic poly-time) This is a set of languages.

$L \in P$ iff $\exists$ polynomial $p(n)$ and a family $\mathcal{F}$ of circuits $\{C_n\}_{n \in \mathbb{N}}$ such that

→ C_n has n inputs & 1 output.



- $|C_n|$ (# of gates) $\leq p(n)$ $\forall\ n \in \mathbb{N}$
- if $|x| = n$ then $C_n(x) = (x \in L?)$
- there exist a polynomial time Turing Machine that on input 1^n outputs C_n

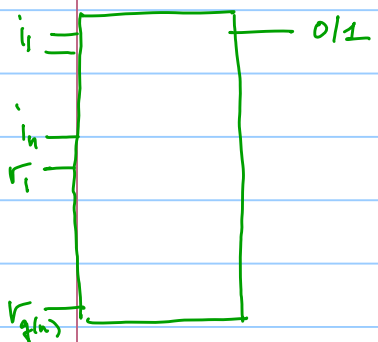
→ This is known as the "uniformity" condition. It prevents us from encoding the answer to hard problems in the circuit itself. (e.g. hard-code the output of C_n to be 0 or 1 depending on whether n is in some undecidable language).

Note that the class P is independent of which particular gate set we happen to choose.

Quantum computation by its very nature will be probabilistic, so we need to first define a probabilistic version of P.

BPP : Bounded Error Polynomial Time

$L \in BPP$ if $\exists$ polynomials $p(n)$, $q(n)$ & family $\mathcal{F}$ of circuits $\{C_n\}_{n \in \mathbb{N}}$ such that



- circuit C_n has input x ($|x| = n$) and random input r, $|r| \leq q(n)$
- $|C_n| \leq p(n)$ $\forall\ n \in \mathbb{N}$.
- uniformity
- if $x \in L$ and $|x| = n$ $\text{Prob}_r[C(x,r) = 1] \geq 2/3$
  if $x \notin L$ and $|x| = n$ $\text{Prob}_r[C(x,r) = 0] \geq 2/3$

Note that the probability of error can be made arbitrarily small by repeating the computation (with fresh random bits) and taking the majority answer.

Quantum circuits.

The function computed by a quantum circuit must be unitary, so the # input qubits = # output qubits.

A circuit is a sequence of primitive gates, each of which acts on $\leq k$ qubits

Since the set of all unitary operators on n-qubits is a continuous space, it is impossible for a finite gate set to generate all of them.

So we need a notion of approximation:

Operator norm: $\|B\| = \max_{|\,|v\rangle\,| = 1} |B|v\rangle|$

We say that U simulates U' to within ϵ if:

$\|U - U'\| \leq \epsilon$. $\quad \|U-U'\| = \max_{\||v\rangle\| = 1} |(U-U')|v\rangle|$

A set G of quantum gates is Universal if $\forall$ U (unitary operators on n qubits) $\forall \epsilon$

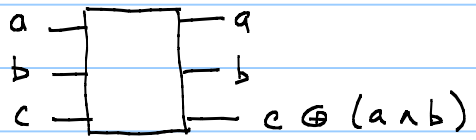
$\exists\, g_1, g_2, \ldots, g_\ell \in G$ such that

$\|U - U_{g_1} U_{g_2} \cdots U_{g_\ell}\| \leq \epsilon$

U_{g_i} is g_i applied to two particular qubits tensored w/ I on rest

Known universal gate sets: (→ without error.)

- CNOT, 1-qubit gates
- CNOT, H, one additional phase (e.g. $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}$)
- Toffoli, H
  - ↳ 3 qubit gate



What is the overhead in approximating U with a finite gate set?

* Any U on n qubits can be computed exactly with 2 qubit gates:

$$U = U_1 U_2 \cdots U_\ell \qquad U_i = G_{g_1(i)\, g_2(i)} \otimes I$$

(unitary on qubits $g_1(i)$ & $g_2(i)$)

[The construction uses exponentially many gates]

Solovay-Kitaev Theorem

Let $G \subseteq SU(d)$ [SU(d) = set of unitary ops on d-dim Hilbert space]

Suppose G is closed under inverse ($g \in G \iff g^{-1} \in G$)

and G generates a dense subset for SU(d)

then $\forall\ U \in SU(d)$, $\epsilon > 0$ $\exists\ g_1, g_2 \cdots g_\ell \in G$

such that $\| U - U_{g_1} \cdots U_{g_\ell} \| \leq \epsilon$ and $\ell = O(\log^2 1/\epsilon)$

(to within any ϵ)

→ this means you can get arbitrarily close to any $U \in SU(d)$ using gates from G.

Can use Solovay-Kitaev to get a universal gate set

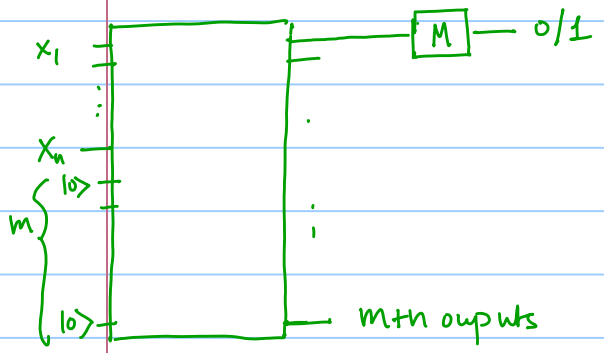
arbitrary U on n qubits —(no error)→ product of ℓ 2-qubit gates ⟶ product of gates from G.

If we want the total error to be ϵ, we need to simulate each gate to within error ϵ/ℓ.
The total number of gates will be $O(\ell \cdot \log^2(\ell/\epsilon))$.

Note that some U will require an exponential number of gates. Even if we only want to match what U does on an input of all 0's (U is ϵ-close to U' if $\| (U-U')|0\rangle \| < \epsilon$) we still need a set of at least $\Omega((1/\epsilon)^{2^n})$ to get within ϵ of any possible U. If our universal gate set has g gates which each act on f qubits, there are $\binom{n}{f}^g$ choices for each gate in a circuit. There are $O(n^{fg})$ possible circuits of size ℓ. $\Rightarrow$ $\ell \geq \Omega\left(\frac{2^n \log(1/\epsilon)}{\log n}\right)$.

BQP:

$L \in BQP$ if $\exists$ polynomials $p(n)$ & $q(n)$ and a family of circuits $\{C_n \in SU(2^n)\}_{n \in \mathbb{N}}$.
↳ made from a finite universal gate set.



- input x on $|x| = n$ qubits
  $m = q(n)$ additional inputs set to 0.
- output: measurement of first qubit (in the 0/1 basis).
- $|C_n| \leq p(n)$ $\forall n$
- uniformity
- if $x \in L$, $|x| = n$ $Pr[\text{output} = 1] \geq 2/3$
  if $x \notin L$, $|x| = n$ $Pr[\text{output} = 0] \geq 2/3$

Note that for every quantum circuit, there is a reverse circuit that computes the inverse function: reverse the order of the gates, and for each gate g, use g^{-1} instead. Since g is unitary $g^{-1} = g^\dagger$

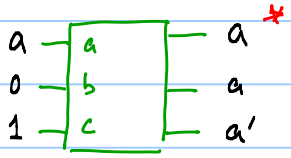
One would expect that $P + BPP \subseteq BQP$, but there are a couple issues to be worked out. Quantum computation must be reversible, whereas classical computation does not have this restriction. Things like fan-out from a gate are easy in the classical setting but problematic in quantum because of the no-cloning theorem.

In order to show $P \subseteq BQP$, we need to show that an arbitrary classical circuit can be transformed into a reversible circuit. We just need to show it for a universal gate set. We will use a Fredkin gate which is a controlled SWAP:

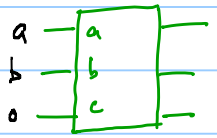
$(0, b, c) \to (0, b, c)$
$(1, b, c) \to (1, c, b)$.

$F^2 = I$ (reversible)

Some of the inputs will be hard-coded to produce certain effects



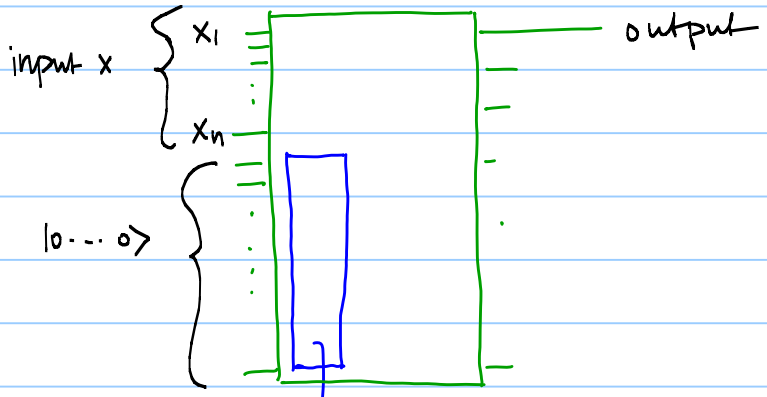
NOT and Fanout



AND

* note this is not cloning. If $|a\rangle$ is a quantum state $|a\rangle = \alpha|0\rangle + \beta|1\rangle$ $|a\rangle|0\,1\rangle$ will produce $\alpha|0\,0\,1\rangle + \beta|1\,1\,0\rangle$ and not $|a\rangle \otimes |a\rangle \otimes X|a\rangle$

For this construction all intermediate states remain classical.



↳ fix auxiliary bits for Fredkin gates. These are dependent on the circuit (i.e $|x|=n$) but not on x.

What about $BPP \subseteq BQP$? For this we need a source of random bits. First make the circuit reversible as before.

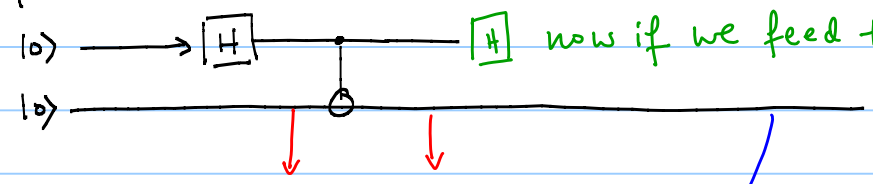
To get a random bit, can do $|0\rangle \to$ [H] $\xrightarrow{|+\rangle}$ [M] $\to$ 0/1 w. prob $1/2$

Measurement can be difficult to implement at intermediate stages. Can it be deferred until the end?

The problem is that interference can reduce the randomness. For example, if the random bit were followed by another H gate, that would completely unrandomize the qubit:

$|0\rangle \to$ [H] $\to \cdots \to$ [H] $\to |0\rangle$

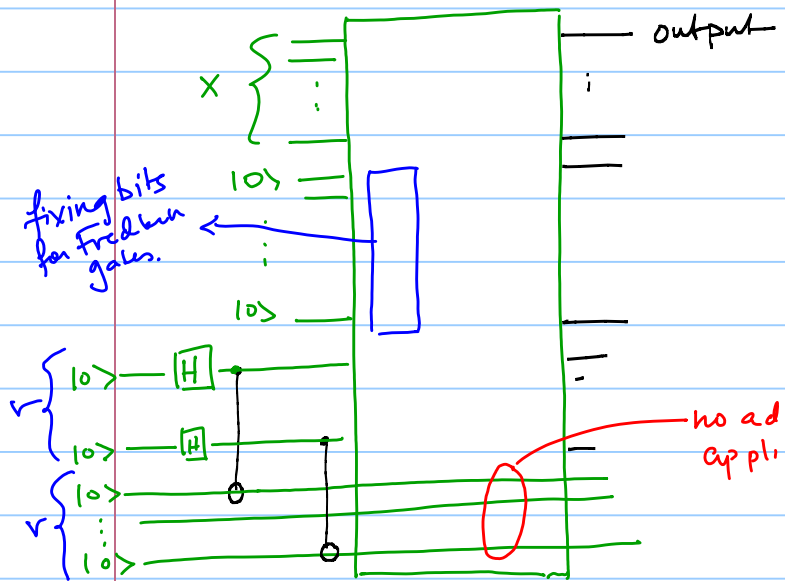
Instead, we can entangle the $|+\rangle$ qubit with another qubit which will not be used again:



now if we feed this qubit into an H gate it the randomness is maintained in the other qubit

$|+\rangle|0\rangle$ $\quad \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ $\quad \frac{1}{\sqrt{2}}(|+0\rangle + |-1\rangle)$

we can measure this qubit at the very end of the computation ... or not at all.



Classical computations that use polynomial space with no limits on time

We have shown that $P \subseteq BPP \subseteq BQP$.
What about limits on the power of BQP?
We will sketch a proof that $BQP \subseteq PSPACE$

Is it possible to prove that quantum computers are strictly more powerful than classical ones? (i.e. $BQP \neq P$ or BPP)? Because $BQP \subseteq PSPACE$ such a proof would also imply that $P \neq PSPACE$. Whether or not $P = PSPACE$ is a long-standing open question in (classical) complexity theory.

$BQP \subseteq PSPACE$ (proof sketch)

We will show how to estimate the amplitude of ending in some standard basis state y:

$$\langle y | U_{p(n)} \cdots U_2 U_1 | x_0 \rangle = \alpha_y$$

$p(n)$ = size of the circuit. gates. x_0 = input

We can then sum over all y s.t. the output bit is 1 $|\alpha_y|^2$. There are an exponential number of y's in this sum but we don't need to keep all the α_y around at one time. We just cycle through all the y's and keep a running tally of the acceptance probability:

current y	current sum $\sum_{y' < y} |\alpha_{y'}|^2$	workspace to compute α_y

Now given y, how much space does it take to compute α_y?

Insert $\sum_z |z\rangle\langle z| = I$ in between each $U_{j+1}\ U_j$:

$$\langle y | U_{p(n)} \cdots U_2 U_1 | x_0 \rangle = \sum_{z_1 z_2 \cdots z_{p(n)-1}} \langle y | U_{p(n)} | z_{p(n)-1} \rangle \cdots \langle z_1 | U_1 | x_0 \rangle$$

Generally $\langle z_j | U_j | z_{j-1} \rangle$ can be determined given z_{j-1} & z_j since U_j is a single gate

It may be an irrational number but it can be approximated to within high accuracy.

Again we have a sum with an exponential number of terms but we don't have to keep them all around simultaneously. Need to store current $z_1 \cdots z_{p(n)}$, and compute the product of the corresponding amplitudes, then move on to the next choice for $z_1 \cdots z_{p(n)-1}$

A word about precision. We need to calculate the probability of acceptance to within some constant ϵ ($= 1/3$). How much precision do we need in our approximations?

The number of terms we are adding is $2^{q(n)}$ for some polynomial q. So we need to estimate each product to within $\left[\frac{\epsilon}{2^{q(n)}}\right]$

If each term in the product is estimated to within $\pm\delta$, and the terms themselves are $O(1)$ then the total error in the product is $O(p(n)\,\delta)$ [$p(n)$ is the # terms in the product]

We need: $\text{const.}\ p(n)\ \delta \le \frac{\epsilon}{2^{q(n)}} \Rightarrow \delta \le \frac{\text{const.}\ \epsilon}{p(n)\ 2^{q(n)}}$

It takes $\log(1/\delta)$ bits of precision to estimate a number to within δ. So we need $O(\log(1/\epsilon) + \log p(n) + q(n))$ bits to specify each number in our calculation.