Circuit model of quantum computing

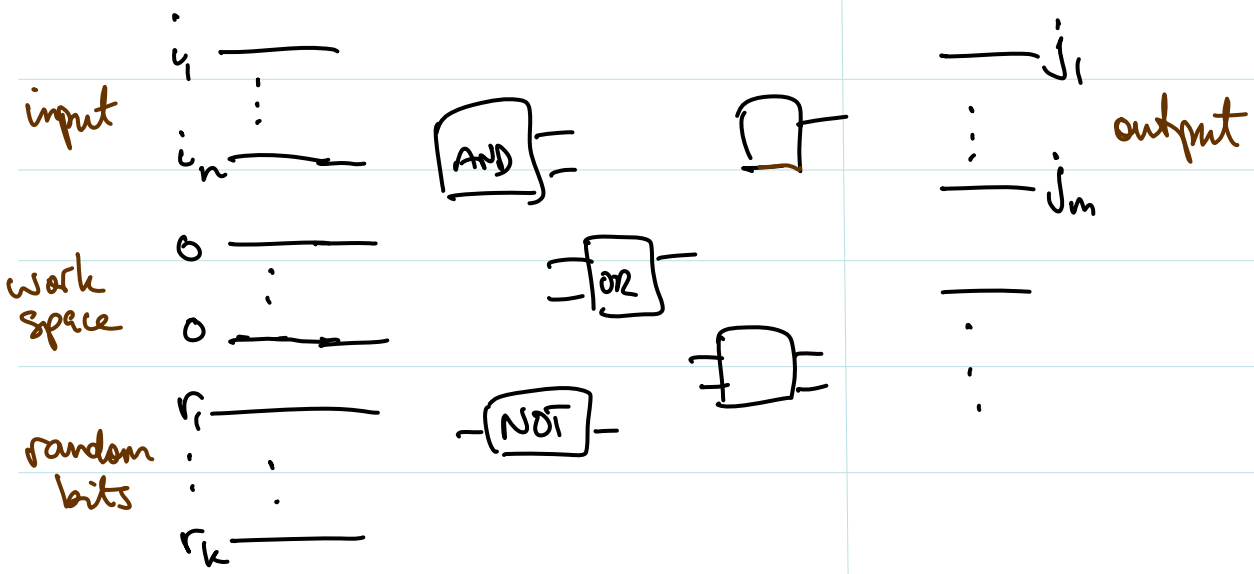
First: some classical notions –

Computational task

Input: bit string $i_1 \ldots i_n$ (input size n)

Output: 1 or more bits j_1 or $j_1 \ldots j_m$

For each n have prescribed circuit C_n of classical gates acting on n bit-lines + extra workspace + extra random bits



Further features

- uniformity condition $n \longrightarrow$ description of C_n by suitable (log-space) classical computation (not a problem in practice--)
- gates chosen from some (finite) universal set of "small" gates to enable arbitrary Boolean $j_1 \cdots j_m = f(i_1 \cdots i_n)$ to be implemented.

Basic class of interest "feasible classical computations"

BPP : Bounded error Probabilistic Polynomial time —

- size of C_n (= #gates) is bounded by poly(n) (so $k = \text{poly}(n)$ too)
- for each input $i_1 \cdots i_n$, answer is correct with prob $> 2/3$ (or $> 1-\epsilon$ for any fixed $\epsilon > 0$)

(Probabilistic here since quantum measurements probabilistic & exact algorithms (deterministic) over-idealised --)

Quantum circuit model

very similar to the above!

- Input – $i_1 \ldots i_n$ represented as basis state $|i_1 \ldots i_n\rangle$
- $n \to C_n$ circuit of quantum gates chosen from some "universal" set.
- C_n description generated by classical poly(n)-time computation on n.
- Don't need random bits (can make them! --) but may use extra "ancilla" qubits in some fiducial state eg $|0\rangle \ldots |0\rangle$.
- Output – measure some specified qubit(s) in standard basis.

Basic class of interest "feasible quantum computations"

- Size of C_n bounded by poly(n).
- for all inputs prob(correct answer) $> 2/3$ as before.

Is $BQP \supseteq \neq BPP$?? No proof (yet?--)! ← would imply P ≠ PSPACE! etc...

But expected to be true by most people...

Universal sets of basic quantum gates

For any unitary U on n qubits (cf discrete circuits vs continuous gates)

- make U exactly

or

- approximate U to arbitrary accuracy ϵ

V approx's U to accuracy ϵ if $\|(U-V)|\psi\rangle\| < \epsilon$ for all $|\psi\rangle$

Examples:

Exact set - $\{$CNOT, all 1-qubit gates$\}$ (no finite exact set possible)

Approx sets - $\{$Toffoli gate, $H\}$

$\{\text{CNOT}, S = Z^{1/4} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, H\}$

$\{$generically any 2-qubit gate by itself$\}$ etc.

For these sets

Accuracy ϵ achieved by poly$(\log \frac{1}{\epsilon})$ o'head in size.

Fact: level of approx grows only linearly with number of concatenated gates

i.e. if $U_i \sim V_i$ accuracy ϵ then

$U = U_1 U_2 \cdots U_k \sim V = V_1 V_2 \cdots V_k$ accuracy $k\epsilon$

Fact if $\| |\psi_1\rangle - |\psi_2\rangle \| < \epsilon$ then mmt probabilities differ by $O(\epsilon)$ too.

Thus BQP conditions insensitive to choice of gate set used:

BQP circuit has size poly(n). If individual gate approx is ϵ, final level is poly(n)ϵ.

To preserve bounded error condition on probabilities

want poly(n)ϵ = const i.e. $\epsilon = \frac{1}{\text{poly}(n)}$

- achieved by $\text{poly}\left(\log \frac{1}{\epsilon}\right) = \text{poly}\log(n)$ o'head in circuit size.

So circuit still poly(n) sized.

Note: All quantum gates are unitary hence reversible.

Given any Boolean f: n-bits $\to$ m-bits $\quad y = f(x)$

Have associated $\tilde{f}$: (n+m)-bits $\to$ (n+m)-bits

$$\tilde{f}(x, y) = (x,\ y \oplus f(x))$$

addition of n-bit strings mod 2
so $y \oplus y = 0 \ldots 0$ for all y.

$\tilde{f}$ always reversible i.e. permutation of (n+m)-bit strings

$$(x,y) \xrightarrow{\tilde{f}} (x, y \oplus f(x)) \xrightarrow{\tilde{f}} (x, y \oplus f(x) \oplus f(x)) = (x, y)$$

So $\quad \tilde{f} = \tilde{f}^{-1}$

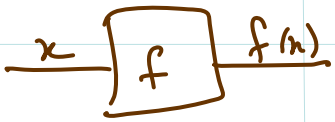
Associated quantum gate U_f on (n+m) qubits

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$$

always unitary (as any permutation of basis states is unitary as linear extension.)

Query complexity and promise problems

Instead of inputs $i_1 \ldots i_n$ we are given "black box" or "oracle" for f : n-bits → m-bits

classical 

quantum 

f maybe a promise on the form of f.

Want to determine some property of the function f with least number of queries to oracle.

Query complexity : how number of queries grows with n

May also be interested in how much further computation is used to process answers to queries.

Examples.

"Balanced vs. constant" problem.

Input: oracle for $f: n\text{-bits} \to 1\text{-bit}$

Promise: f is either (a) a constant function or

(b) a "balanced" function i.e. $f(x) = 0$ resp. 1 for exactly half of all 2^n inputs.

Problem: Determine whether f is balanced or constant.

(can ask for answer to be correct with certainty or merely some prob. say 0.99)

Boolean satisfiability

Input: oracle for $f: n\text{-bits} \to 1\text{-bit}$

Promise: no restriction on form of f

Problem: determine whether there is an x with $f(x)=1$, or not.

Search

Input: oracle for $f: n\text{-bits} \to 1\text{-bit}$

Promise: there is a unique x with $f(x) = 1$

Problem: find that unique x.

Periodicity

Input: oracle for $f: \mathbb{Z}_n \to \mathbb{Z}_m$ (integers mod n)

Promise: f is periodic i.e. there is a least r such that
$f(x+r) = f(x)$ for all x ($+$ is addition mod n)

Problem: find the period r.

A very useful quantum feature –

computing values in superposition. e.g. for n qubits

$$|0\rangle \quad |0\rangle \quad \cdots \quad |0\rangle$$

$$\downarrow H \quad \downarrow H \quad \cdots \quad \downarrow H$$

$$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

Get uniform superposition of all 2^n n-bit strings with only $O(n)$ computational effort!

Then

$$\frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle|f(x)\rangle \equiv |f\rangle$$

Computes <u>all</u> values of f in superposition with just a <u>single</u> query.

recall "interference" ...

Mathematical identity of $|f\rangle$ encodes all f values

But physically: cannot read out all this information!

can get only small amount of information about identity of $|f\rangle$

quantum measurement destructive!

Further unitary processing of $|f\rangle$ before measurement:
can get global property of all values, small amount of info,
but maybe still hard to get classically!

What kind of information is available from $|f\rangle$?

"bal vs const" ✓ "periodicity" ✓

"SAT & search" ✗!

Intuition: "can see pattern structure, stable under small changes"

c.f. $|f\rangle$ for uniq-sat f vs un-sat f ....

Deutsch-Jozsa (DJ) algorithm (Balanced vs. const problem)

Input: oracle for f: n-bits → 1-bit

Promise: f is either (a) constant
or (b) balanced

Problem: determine (a) vs. (b) with certainty!

Classically: $2^{n/2}+1$ queries necess & sufficient

Quantumly: can solve with single query (+O(n) o'head processing)

Recall $|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$

Set y-register to

$$|A\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|1\rangle = HX|0\rangle$$

then ...

Then note

$$|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{U_f} \frac{1}{\sqrt{2}}|x\rangle\left[\,|f(x)\rangle - |1 \oplus f(x)\rangle\,\right]$$

$$= \begin{cases} \frac{1}{\sqrt{2}}|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x)=0 \\ -\frac{1}{\sqrt{2}}|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x)=1 \end{cases}$$

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

Now do this in superposition for all x-values:

$$\frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle|A\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum (-1)^{f(x)}|x\rangle|A\rangle$$

Discard $|A\rangle$ to get

$$|f\rangle \equiv \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} (-1)^{f(x)}|x\rangle$$

← f-values now encoded as $\pm$ signs.

If f is constant: all values/signs are same so

$$|f\rangle = \pm \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle = \pm H \otimes \cdots \otimes H |0 \ldots 0\rangle$$

Also $HH = I$

So applying $H \otimes \ldots \otimes H$ we get $\pm |0 \ldots 0\rangle$

If f is balanced exactly half of signs are $+$ & $-$ in $|f\rangle$

Hence $|f_{bal}\rangle$ is orthogonal to $|f_{const}\rangle = \pm \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle$

Since $\langle f_{const} | f_{bal} \rangle$ = sum of all $(-1)^{f(x)}$'s $= 0$

regardless of where the $\pm$ signs are!

$H^{\otimes} \cdots \otimes H \, |f_{bal}\rangle$ orthogonal to $|0\ldots0\rangle$

So $= \sum_{\text{all } x \neq 0\ldots0} a_x |x\rangle$ for some a_x's

Thus

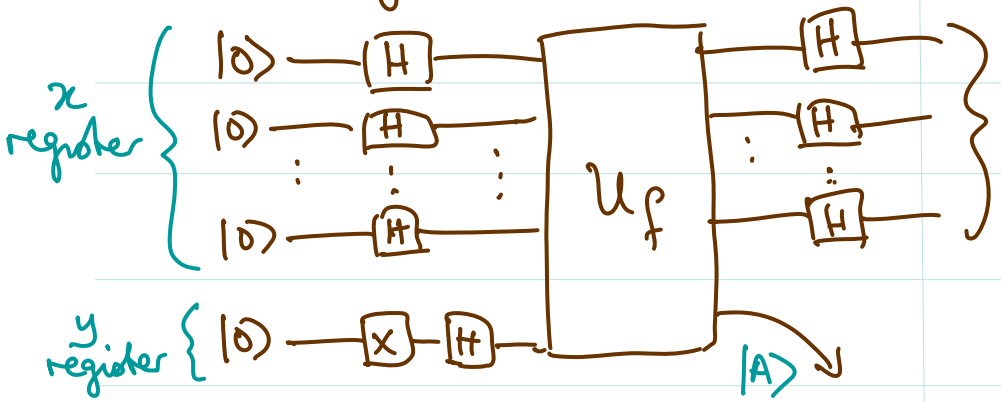
after applying $H^{\otimes} \cdots \otimes H$ to $|f\rangle$ we measure all qubits.

f const : get $0\ldots0$ with certainty

f bal : get some $i_1 \ldots i_n$ which is never $0\ldots0$.

} hence distinguish bal vs const with certainty.

Circuit diagram



if all 0 output "const"

if not all 0 output "balanced"

Exercise:

Bernstein-Vazirani problem:

For any n-bit string $a = a_1 \ldots a_n \neq 0\ldots0$

$f_a(x) = a.x = a_1x_1 \oplus \cdots \oplus a_nx_n$

is balanced & outputs a with certainty.

Remark 1.

If we tolerate some (arb. small) error ϵ then there's a classical algorithm with constant number of queries!

Select K x's uniformly@random. Query $f(x_1), \ldots f(x_K)$.

If all same – output "const"

If not all same – output "balanced"

[If f really const – output always correct.

If f really balanced - each $f(x_i)$ is 0/1 with probs $\frac{1}{2}$

So Prob(answer wrong) = Prob(get all 0's or all 1's) = $\frac{2}{2^K}$

Then $\frac{2}{2^K} <$ any chosen ϵ for suitably large const K]

Not really a problem! – there are more complicated oracle problems that maintain exponential separation between classical & quantum query complexity, even if we tolerate error $\epsilon = 1/3$.

(e.g. Simon's algorithm...)

Remark 2.

Does DJ algorithm imply q. comp. better than cl. comp?

– need oracle model here ... maybe if had formula or source code for f, could decide bal vs const. classically easily too, by examining structure of f? ... unknown ...

Is there a "standard" computational task (with input = bit string etc) with provable exponential quantum speed-up?

i.e. in BQP but not in BPP ?

None known proved! Difficulty with classical complexity theory

e.g. unproven that NP or PSPACE $\supsetneq$ P!

But have factoring – known in BQP

not believed (?) to be in BPP.