LECTURE 4 (January 29)

TODAY BQP and its properties

RECAP

- BQP = class of problems that can be solved with bounded error by a P-uniform quantum circuit family

  Remark One can consider the class EQP where quantum circuits solve the problem exactly, but this class depends on the exact gate set used, so it is not very interesting

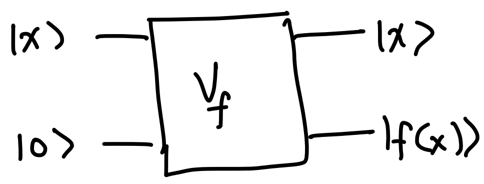
- $BQP^{BQP} = BQP$?

BQP^{BQP} = A BQP algorithm/circuit with a $\underbrace{\text{BQP oracle}}$

E.g. the algorithm can ask if a 2SAT formula is satisfiable

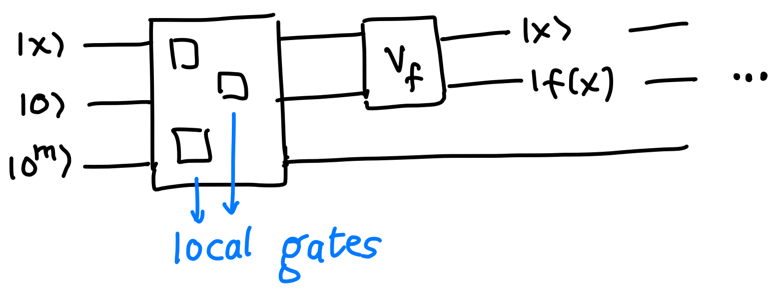
Let $f(x)$ be the answer of the decision problem the BQP oracle solves on input x

BQP^{BQP} circuit can use the following unitary gate in the circuit (apart from the standard universal 2-local gates)



This gate is not local, solves the BQP problem exactly, and can be used in a superposition

For instance,



We know that f can be computed with bounded error by a BQP circuit family

To show that $BQP^{BQP} \subseteq BQP$,

the obvious idea is to use the BQP circuit that computes f instead of V_f

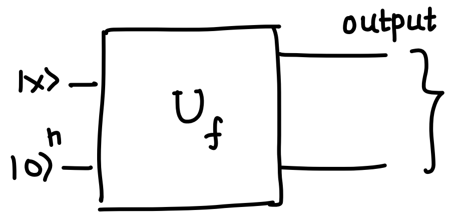
Let us call this circuit U_f

Can we use U_f to simulate the behavior of the ideal oracle V_f ?

Now there are two issues that need to be handled

① <u>Error</u> : U_f only computes f with probability $\geq \frac{2}{3}$ as opposed to V_f

All we know about U_f is that



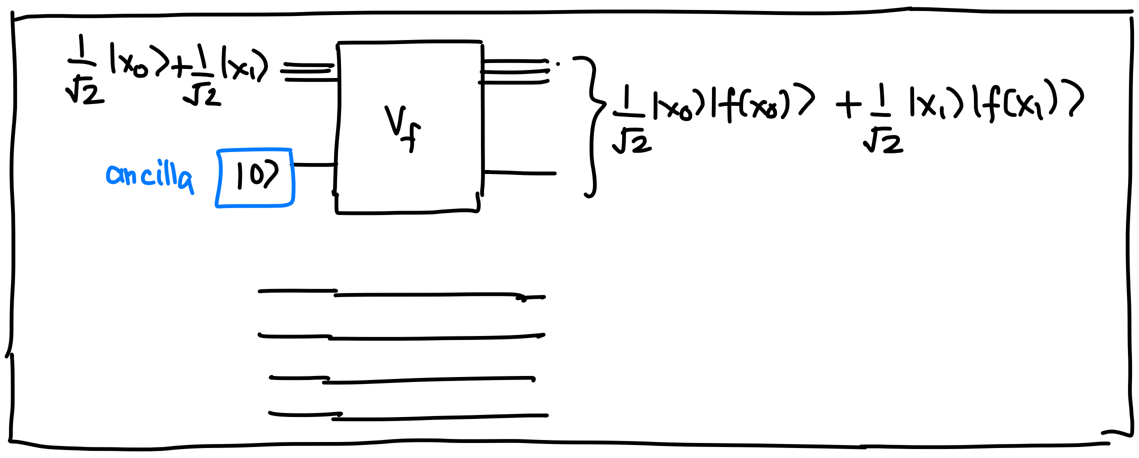
$$= \alpha|f(x)\rangle|\psi_{f(x)}\rangle + \beta|\neg f(x)\rangle|\psi_{\neg f(x)}\rangle$$

Entangled Junk

$|\alpha|^2 \geq 2/3$

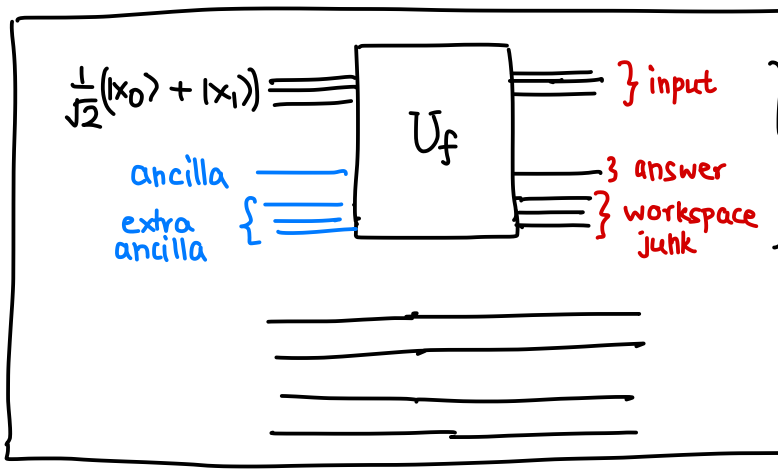
② <u>Entangled Junk</u> is also a problem

For example, suppose in circuit C (which uses V_f gates) at an intermediate step



$$\frac{1}{\sqrt2}|x_0\rangle|f(x_0)\rangle + \frac{1}{\sqrt2}|x_1\rangle|f(x_1)\rangle$$

But if we used U_f instead



Error

$$\frac{1}{\sqrt2}\alpha_0|x_0\rangle|f(x_0)\rangle|\psi_{f(x_0)}\rangle + \frac{1}{\sqrt2}\beta_0|x_1\rangle|\overline{f(x_0)}\rangle|\psi_{\overline{f(x_0)}}\rangle$$
$$+\frac{1}{\sqrt2}\alpha_1|x_1\rangle|f(x_1)\rangle|\chi_{f(x_1)}\rangle + \frac{1}{\sqrt2}\beta_1|x_1\rangle|\overline{f(x_1)}\rangle|\chi_{\overline{f(x_1)}}\rangle$$

This is a mess!
Infact, this state may not be close to the one we want

For instance, if $f(x_0)=f(x_1)=0$
the state on top is a tensor product
$$\left(\frac{1}{\sqrt2}|x_0\rangle+\frac{1}{\sqrt2}|x_1\rangle\right)\otimes|0\rangle$$

but the state in the U_f circuit may be very far from it if $|\psi_0\rangle$ and $|\chi_0\rangle$ are orthogonal

How can we solve these problems ?

① Error $\longrightarrow$ use amplification to make error very small , i.e. $|\alpha|^2 \geq 1-2^{-n}$
$|\beta|^2 \leq 2^{-n}$

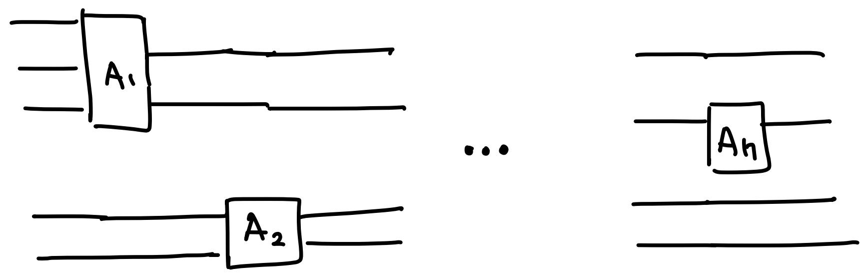
This means that the state in the U_f circuit is exponentially close to

$$\frac{1}{\sqrt{2}}|x_0\rangle|f(x_0)\rangle|\psi_{f(x_0)}\rangle + \frac{1}{\sqrt{2}}|x_1\rangle|f(x_1)\rangle|\chi_{f(x_1)}\rangle$$

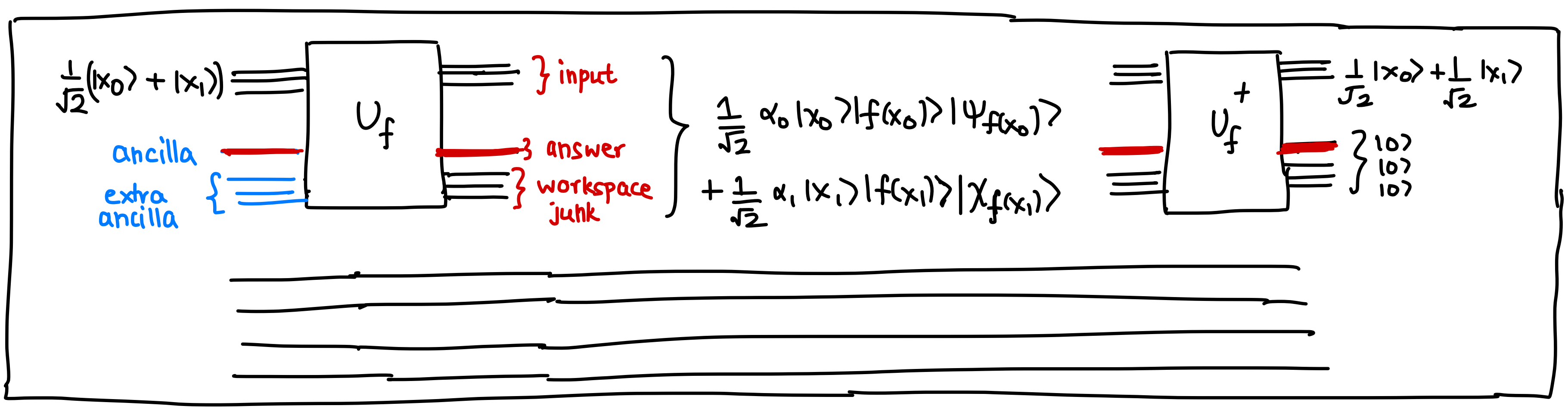
② Junk $\longrightarrow$ use a trick called uncomputation

Recall that any unitary U is reversible , its inverse is $U^\dagger$

What is the inverse of a quantum circuit ?



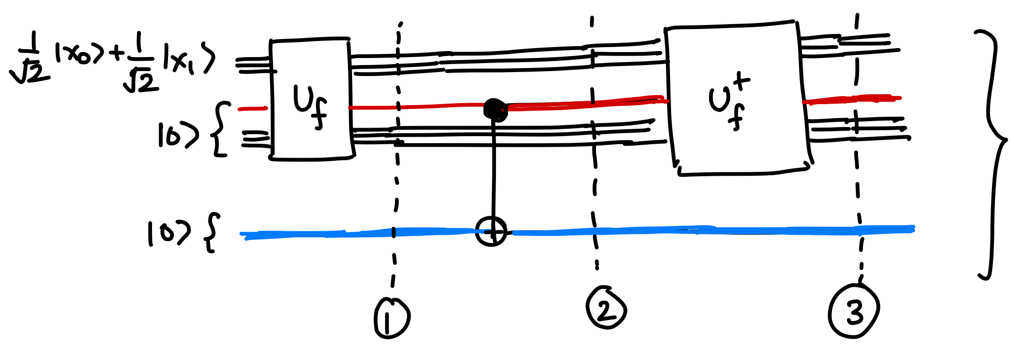
Now if we run U_f followed by $U_f^\dagger$ in our circuit above (ignoring exponentially small error)



This resets the "junk" to $|0\rangle\otimes|0\rangle\cdots\otimes|0\rangle$ which is unentangled with the input $\frac{1}{\sqrt{2}}(|x_0\rangle+|x_1\rangle)$
so we can now throw them away

But we also reset the "output" qubit to 0 and we want to keep it in order to continue the computation

This has a simple solution : CNOT to "copy" the output qubit to an extra ancilla



① $\approx \left(\frac{1}{\sqrt{2}} |x_0\rangle \overbrace{|f(x_0)\rangle}^{\text{control}} |\psi_{f(x_0)}\rangle \overbrace{|0\rangle}^{\text{target}} + \frac{1}{\sqrt{2}} |x_1\rangle \overbrace{|f(x_1)\rangle}^{\text{control}} |\chi_{f(x_1)}\rangle \overbrace{|0\rangle}^{\text{target}} \right)$

② $\approx \frac{1}{\sqrt{2}} |x_0\rangle |f(x_0)\rangle |\psi_{f(x_0)}\rangle |f(x_0)\rangle + \frac{1}{\sqrt{2}} |x_1\rangle |f(x_1)\rangle |\chi_{f(x_1)}\rangle |f(x_1)\rangle$

③ $\approx \frac{1}{\sqrt{2}} |x_0\rangle |0\rangle |0 \cdots 0\rangle |f(x_0)\rangle + \frac{1}{\sqrt{2}} |x_1\rangle |0\rangle |0 \cdots 0\rangle |f(x_1)\rangle$

Final state $\approx \frac{1}{\sqrt{2}} |x_0\rangle |0 \cdots 0\rangle |f(x_0)\rangle + \frac{1}{\sqrt{2}} |x_1\rangle |0 \cdots 0\rangle |f(x_1)\rangle$

Now we can throw away the extra 0 ancillas (since they are unentangled and can carry on with rest of the computation

The exponentially small error does not create a problem (exercise)

Thus, $\boxed{BQP^{BQP} = BQP}$

Remark Above we have assumed that all measurements happen at the end otherwise the scheme above runs into problems

This can always be assumed by the principle of deferred measurement which you will be asked to prove in an optional homework exercise

BQP and classical complexity classes

What is the smallest classical complexity class that contains BQP ?

In other words, how efficiently can quantum computation be simulated classically ?

Let us start as crudely as possible and iteratively refine our upper bound to smaller complexity classes

Theorem 1 $BQP \subseteq EXP \longrightarrow$ problems that can be solved deterministically in exp(poly(n))-time

Proof The quantum circuit applies unitaries that live in $\mathbb{C}^{2^L}$ where $L = \text{poly}(n)$

Just write down these matrices and multiply them with the input state □

Can we do better ?

Theorem 2 BQP $\subseteq$ PSPACE $\subseteq$ EXP

Proof   Exercise

Theorem 3 BQP $\subseteq$ PP $\subseteq$ PSPACE $\subseteq$ EXP   where PP = class of problems that can be solved by a probabilistic algorithm with error $< \frac{1}{2}$ $\left(\text{e.g. } \frac{1}{2} - 2^{-n} \right)$ in poly-time

Note that amplification is not possible in PP and NP $\subseteq$ PP Why?

Proof   Let us use $\{H, \text{CNOT}, \text{CCNOT}\}$ gate set

Key idea   Write the final state as sum of paths on a tree

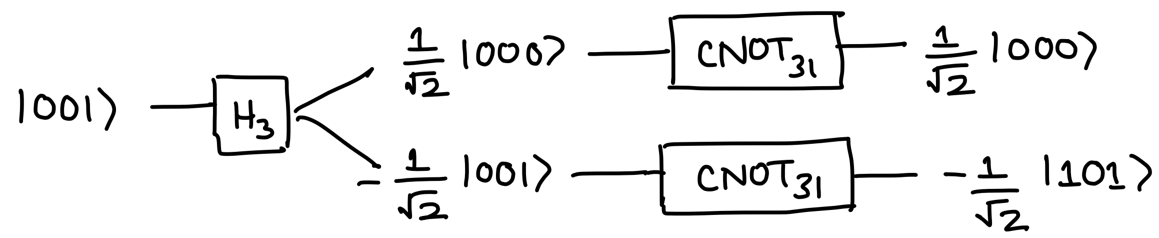
CNOT and CCNOT only flip one qubit $(|x\rangle \to |y\rangle)$

H splits the state into a superposition with equal magnitude

$$|x\rangle \to \frac{|y_1\rangle \pm |y_2\rangle}{\sqrt{2}}$$

Circuit with h Hadamard gates $\Rightarrow$ Tree with 2^h leaves

E.g.



At the leaf of the tree is $\pm \frac{1}{\sqrt{2^h}} |y\rangle$ for some $y \in \{0,1\}^{\text{poly}(n)}$

label of this root-leaf path, label (P)

sign of this root-leaf path, sign (P)

Then, the final state $|\psi\rangle = \frac{1}{\sqrt{2^h}} \sum_P \text{sign}(P) |\text{label}(P)\rangle$

$$= \frac{1}{\sqrt{2^h}} \sum_{y \in \{0,1\}^{\text{poly}(n)}} \underbrace{\left(\sum_{\substack{\text{Paths } P \\ \text{with label } y}} \text{sign}(P) \right)}_{:= \alpha_y} |y\rangle$$

Then, $\alpha_y^2 = \frac{1}{2^h} \sum_{\substack{p,p' \\ \text{label}(p)=\text{label}(p')=y}} \text{sign}(p) \cdot \text{sign}(p')$

$\mathbb{P}[\text{ BQP algorithm outputs } 1] - \mathbb{P}[\text{BQP algorithm outputs } 0]$

$$= \sum_y |\alpha_y|^2 (-1)^{\mathbb{1}[y_1=0]}$$

$$= \frac{1}{2^h} \sum_{\substack{p,p' \\ \text{w/ same} \\ \text{labels}}} \underbrace{\text{sign}(p) \cdot \text{sign}(p') (-1)^{\mathbb{1}[(\text{label}(p))_1 = 0]}}_{:= \beta(p,p')} = \begin{cases} > \frac{1}{3} & \text{if correct answer is } 1 \\ < -\frac{1}{3} & \text{if correct answer is } 0 \end{cases}$$

<u>PP algorithm</u>

Randomly select two paths p, p' in the tree

- If labels are different, just accept / reject w.p. $1/2$
- If labels are same,

accept iff $\underbrace{\text{sign}(p) \cdot \text{sign}(p') (-1)^{\mathbb{1}[(\text{label}(p))_1 = 0]}}_{:= \beta(p,p')} > 0$

Time = poly(n)

$$\mathbb{P}[\text{Accept}] - \mathbb{P}[\text{Reject}] = \frac{1}{2^{2h}} \sum_{\substack{p,p' \\ \text{label}(p)=\text{label}(p')}} \beta(p,p') = \begin{cases} \geq \frac{1}{2^h \cdot 3} & \text{if correct answer is } 1 \\ \leq -\frac{1}{2^h 3} & \text{if correct answer is } 0 \end{cases}$$

$\square$