LECTURE 10 September 21st, 2023

PART II Fundamental Quantum Algorithms

Today Dealing with Garbage (contd)
Universal Quantum Gate Set
Deutsch's Algorithm

RECAP

↳ Coming soon

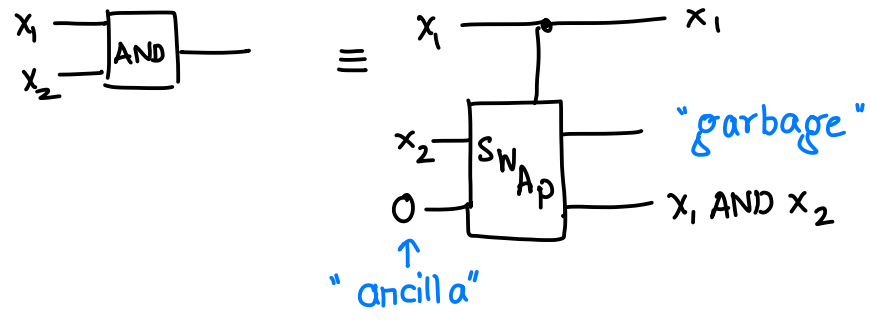
Q1: Does ∃ F which quantum circuits can compute much more efficiently than classical ones?

Last time ← Q2: Can quantum circuits simulate classical circuits?

Today ← Q3: Does the exact quantum gates matter?

Last time we saw that the answer to Q2 was YES but all classical operations need to be done in a reversible manner

E.g. AND gate



Dealing with unwanted garbage bits

Suppose we want to use a classical circuit as a subroutine in a quantum circuit

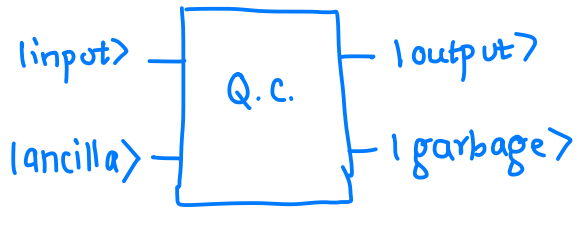
$$F: \{0,1\}^n \longrightarrow \{0,1\}^m$$

If we implement the classical circuit reversibly i.e.

$$|x_1, \ldots x_n\rangle \underbrace{|00\cdots0\rangle}_{\text{ancilla}} \longrightarrow |F(x)\rangle \underbrace{|G(x)\rangle}_{\text{garbage}}$$

Puzzle (in-class) Suppose we have a reversible circuit $(P, Q, 0\cdots0) \longrightarrow (P\cdot Q, \text{garbage})$
Why can't we reverse it & get an efficient algorithm for factoring?

If things are in superposition, garbage is difficult to deal with

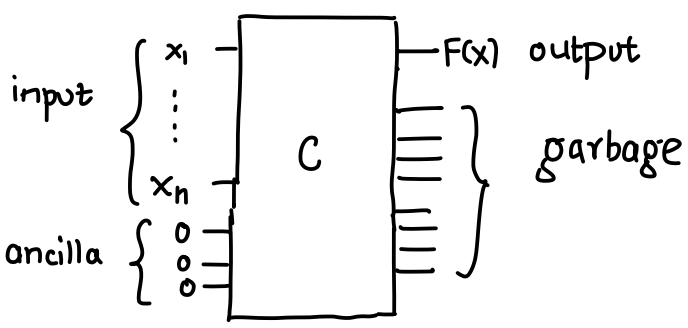


If garbage is entangled with output we can't easily get rid of it
Also, makes it difficult to use it as a subroutine

So, often we want to remove garbage

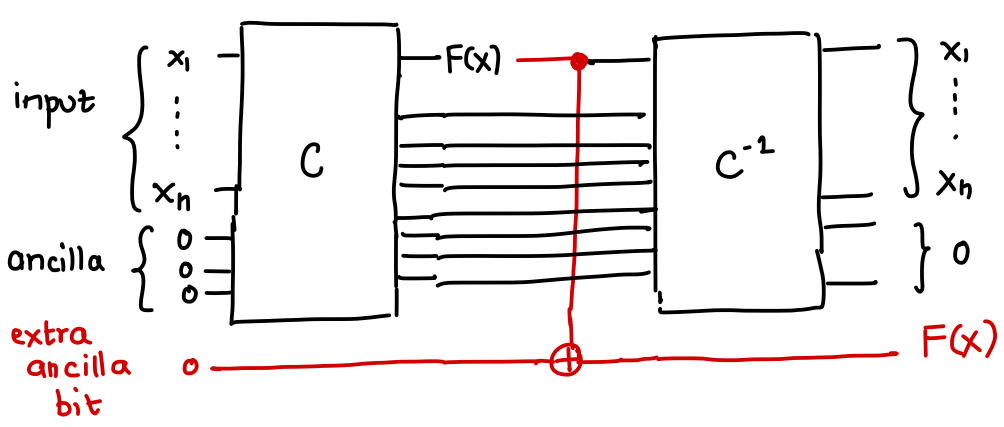
Uncomputing Garbage [Bennett '80s]

Say $F: \{0,1\}^n \to \{0,1\}$ computed reversibly



$$(x_1 \dots x_n, 0 \dots 0) \longrightarrow (F(x), g(x))$$

Trick: copy the answer somewhere and undo the computation



$$(x_1 \dots x_n, 0 \dots 0, 0) \longrightarrow (x_1 \dots x_n, 0 \dots 0, F(x))$$

If we initialize the extra ancilla bit to 1

$$(x_1 \dots x_n, 0 \dots 0, 1) \longrightarrow (x_1 \dots x_n, 0 \dots 0, 1 \oplus F(x))$$

So, overall we get a quantum circuit implementing

$$|x_1 \dots x_n\rangle |0 \dots 0\rangle |y\rangle \longrightarrow |x_1 \dots x_n\rangle |0 \dots 0\rangle |y \oplus F(x)\rangle$$

We can even throw these away now since they are unentangled!

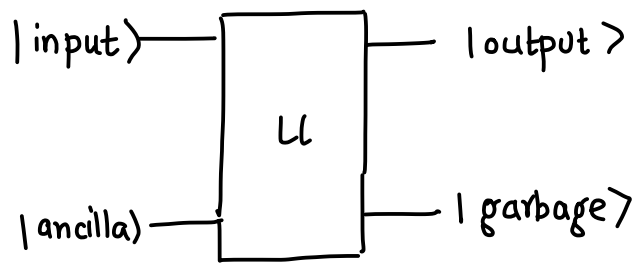
Definition A quantum circuit implements $F: \{0,1\}^n \to \{0,1\}^m$ if it computes it as

$$|x_1 \dots x_n\rangle |y\rangle \longrightarrow |x_1 \dots x_n\rangle |y \oplus F(x)\rangle$$

($y = y_1 \dots y_m$)

Let's try to answer Q3 now! What kind of gates do we need for a quantum circuit?

Consider a general quantum circuit which implements a unitary on $n+a$ qubits



$2^{n+a} \times 2^{n+a}$ unitary

Extremely big unitary & difficult to build in general

Are one or two qubit gates enough to implement U?

YES!

Universal Gate Set

$\{H, S, CNOT, T\}$ is universal for quantum computing → similar to how AND, OR, NOT is universal for classical circuits

where $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ Phase gate

$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

Since ∃ uncountably many unitaries, here we can only get ε-approximation meaning the output state of the quantum circuit using $\{H, S, CNOT, T\}$ gate is ε-close on any input state to the real output state:

$$\| U|\psi\rangle - C|\psi\rangle \| \leq \varepsilon$$

↑ universal quantum circuit

→ All measurement outcomes will only differ by ε

One wants to choose $\varepsilon = \frac{1}{poly(n)}$ or even 2^{-n} where $n = \#$ qubits

So, dependence on ε matters a lot! Does the gate set matter for this?

Solovay-Kitaev theorem says that using a different universal gate set can only reduce the number of gates by a factor of $\log^{1}/_{\varepsilon}$ i.e.

$$\#\text{ Gates with universal gate set } G \lesssim \#\text{ Gates with } \{H, CNOT, S, T\} \cdot \log \frac{1}{\varepsilon}$$

Analog of Shannon's Theorem for Quantum Setting

Any unitary U on n qubits can be approximated, with any universal gate set, to ε-precision using

$$O(4^n \cdot \log^{c} 1/\varepsilon) \text{ gates}$$

Most unitaries also require (roughly) this many gates.

Now we can start with answering Q1 — can quantum circuits be more efficient in computing a function?

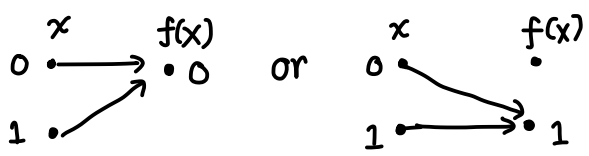
Deutsch's Algorithm

This was arguably the first ever quantum algorithm!

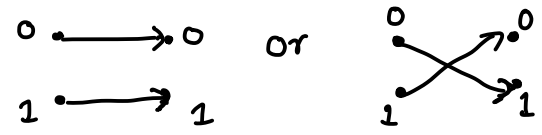
What is the problem?

Given a function $f: \{0,1\} \to \{0,1\}$, determine if it is constant or balanced

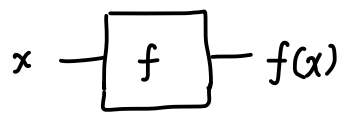
constant : $f(0) = f(1)$



balanced : $f(0) \neq f(1)$



How's the function given to you? Only "black-box" or "query" access



One is given an API which allows you to get the value of f on any input. This is the only way to access f. In particular, you can't see the code of how f is implemented.

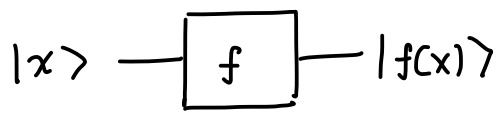
How many queries are needed classically to solve the problem (with no error)?

- 2 queries are necessary and sufficient

Deutsch's algorithm does it in a single query ← 2× speedup over classical algorithms

Quantum Queries

What does it mean to query a function as a black-box?



is not reversible and hence not unitary in general

e.g. f = AND

We need to implement it reversibly as we have seen:

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

input output

Let us call this unitary U_f

A quantum algorithm can query in superposition

It is easy to see that 2 queries are still required if we only use a classical reversible circuit

Suppose we put the first qubit in $|+\rangle$ state

$|+\rangle|0\rangle$ — U_f — $? = \frac{1}{\sqrt{2}} U_f|00\rangle + \frac{1}{\sqrt{2}} U_f|10\rangle$

$= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$

$= \frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$

↑ Contains information about both f(0) and f(1)!

How do we extract it though?

Measuring 1st qubit collapses to

$|0\rangle|f(0)\rangle$ or $|0\rangle|f(1)\rangle$ w\prob $\frac{1}{2}$ each

We could have done this classically as well by querying on a random bit

Let us try putting the 2nd qubit in superposition:

$|x\rangle|-\rangle$ — U_f — $? = \frac{1}{\sqrt{2}} U_f|x\rangle|0\rangle - \frac{1}{\sqrt{2}} U_f|x\rangle|1\rangle$

$= \frac{1}{\sqrt{2}}|x0\rangle - \frac{1}{\sqrt{2}}|x1\rangle$

$= \frac{1}{\sqrt{2}}|x\rangle|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|x\rangle|1 \oplus f(x)\rangle$

$= |x\rangle\left(\frac{1}{\sqrt{2}}|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|1 \oplus f(x)\rangle\right)$

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

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

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

f is implemented in the global phase, but still measurement will not help with a global phase

In summary: First qubit in superposition $|+\rangle|0\rangle$ — U_f — $\frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$

Second qubit in superposition $|x\rangle|-\rangle$ — U_f — $(-1)^{f(x)}|x\rangle|-\rangle$

Let's put both qubits in superposition :

$$|+\rangle|-\rangle \;\text{—}\boxed{U_f}\text{—}\; \frac{1}{\sqrt{2}}U_f|0\rangle|-\rangle + \frac{1}{\sqrt{2}}U_f|1\rangle|-\rangle$$

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

Let's focus on the first qubit and ignore the second

What we have is a qubit $\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle$

$$= \begin{cases} |+\rangle & \text{if } f(0)=f(1) \\ |-\rangle & \text{if } f(0)\neq f(1) \end{cases}$$

up to global phase

These two states can be distinguished perfectly! Interference at work!

Final Algorithm

(i) Create the state $|+\rangle|-\rangle$

(ii) Apply U_f

(iii) Measure the first qubit in $|\pm\rangle$ basis

If outcome is "+", output "f is constant"

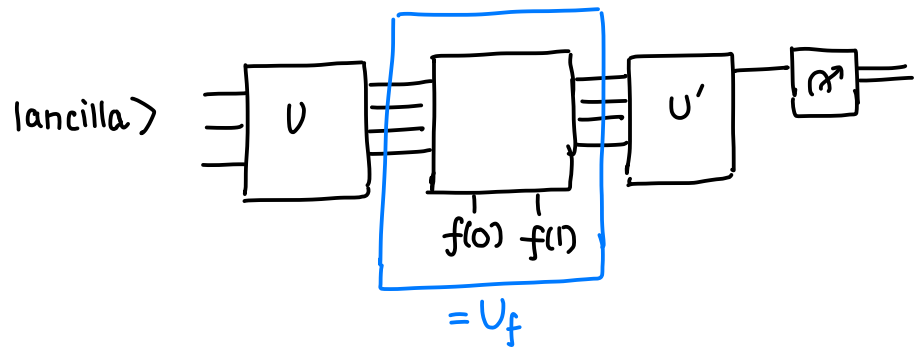
If outcome is "−", output "f is balanced"

Exercise Draw a circuit diagram for this algorithm

Warning You may wonder that this algorithm has no input qubits

The input is the truth table of the function f
but it can only be accessed as a "black-box" or "oracle"

Basically, the circuit can be drawn as



In summary, we take our input ≡ the truth table of f and the ancillas and write a subroutine to compute f reversibly
The quantum algorithm is then allowed to use U_f as a gate in the circuit

The subroutine is the only way to access the input but it may be very complex to implement

So, number of queries is not the final word for efficiency

so, why study this model ?

① Theoretically interesting and one can rigorously compare quantum vs classical

② For most practical quantum algorithms, # queries ≅ # gates

③ We don't know how to prove that there are no efficient classical circuits in terms of # gates

This model is called query complexity model or "black-box model" or "oracle model"

NEXT TIME Quantum algorithms can give exponential advantage!