Quantum computing III

Aleksandr Chatrchyan
24/01/23

Previous lecture covered several quantum algorithms: Deutsch-Jozsa, Bernstein-Vazirani, Simon's algorithm.

Today:

Shor's algorithm, for finding the prime factors of an integer N.

- developed by Peter Shor in 1994.
- runs in polynomial time $\sim (\log N)^2$. In contrast, classical algorithms are polynomial in N.
- important for cryptography: could be used to break RSA.
- In 2001 was demonstrated at IBM for $15 = 5 \cdot 3$

The efficiency of Shor's algorithm is, among others, due to the efficiency of the quantum Fourier transform.

- Will be discussed first.

Main reference: Ronald de Wolf lecture notes. (1907.09415) (chapter 4 & 5)

Quantum Fourier transform (QFT)

Definition: classical FT applied to a vector of amplitudes of a quantum state.

classical FT

acts on a vector $(x_0, \dots, x_{N-1}) \in \mathbb{C}^N$ and maps it to $(y_0, \dots, y_{N-1}) \in \mathbb{C}^N$ according to

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \omega_N^{kn} x_n$$

where $k = 0, \dots, N-1$, and $\omega_N = e^{\frac{2\pi i}{N}}$

quantum FT

acts on a quantum state $|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$ and maps it to $|y\rangle = \sum_{i=0}^{N-1} y_i |i\rangle$ according to

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \omega_N^{kn} x_n$$

$k = 0, \dots N-1$, $\omega_N = \frac{2\pi i}{N}$

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} \cdots & \vdots & \cdots \\ \cdots & \omega_N^{ij} & \cdots \\ \cdots & \vdots & \cdots \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1 & \omega_N & \omega_N^2 & \cdots & \\ 1 & \omega_N^2 & \ddots & & \\ \vdots & \vdots & & & \end{pmatrix}$$

Properties of $\hat{F}_N$

- $\hat{F}_N^\dagger \hat{F}_N = I$
- $F_N^T = F_N$

Since QFT is linear, it sufices to implement it correctly an all basis states $|x\rangle = |x_1\rangle |x_2\rangle \dots |x_n\rangle$

$$|x\rangle \to \hat{F}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jx} |j\rangle$$

Useful identity

$$\hat{F}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jx} |j\rangle = \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + \omega_N^{(2^{n-\ell} \cdot x)} |1\rangle \right) =$$

product state, no entanglement.

$$= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \frac{x_n}{2}} |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \left(\frac{x_n}{4} + \frac{x_{n-1}}{2} \right)} |1\rangle \right) \otimes \dots$$

$$\dots \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \left(\frac{x_n}{2^n} + \dots + \frac{x_1}{2} \right)} |1\rangle \right)$$

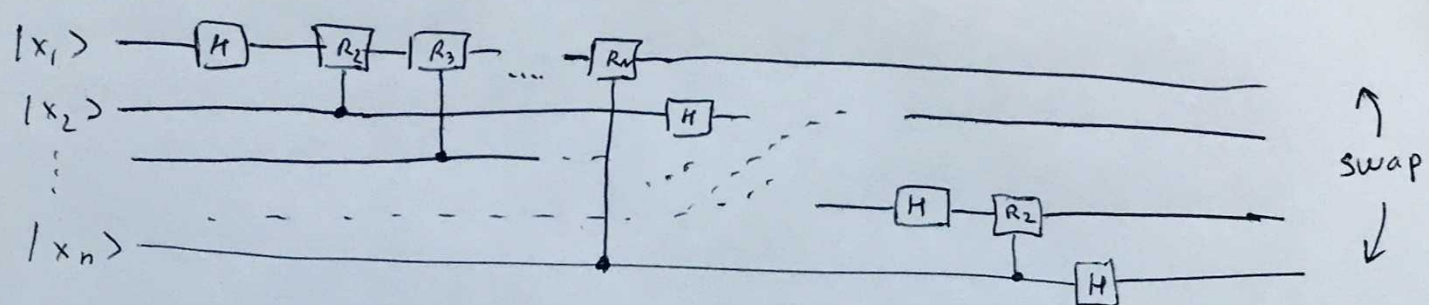
Efficient quantum circuit

We allow for the following elementary gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad R_s = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^s}} \end{pmatrix}$$

Hadamard gate

Implementation in a circuit



$\mathcal{O}(\log N)^2$ elementary operations. In contrast, classical FFT requires $\mathcal{O}(N \log N)$ operations.

Outline of Shor's algorithm

Suppose we want to find factors of the composite number $N > 1$

1) Pick a random integer $a < N$

2) If $\gcd(a, N) > 1$ (greatest common divisor), we are done!

3) If $\gcd(a, N) = 1$ (a is coprime to N)

- Consider the function $f(x) = a^x \pmod N$

  (e.g. $f(0) = 1$, $f(1) = a$, $f(2) = a^2 \pmod N$, ...)

- Use a quantum <u>period</u> finding subroutine to find the period of f.

  $r \leq N$, such that $f(x+r) = f(x)$ or, equivalently, $a^r = 1 \pmod N$

- $\gcd(a^{r/2}+1, N)$ & $\gcd(a^{r/2}-1, N)$ are <u>non-trivial factors of N</u>

- If r is odd, or $a^{r/2} - 1 = 0 \pmod N$, go to step 1)

  ↑ the probability for this is $\leq \frac{1}{2}$.

Explanation:

$a^r = 1 \pmod N \iff a^r - 1 = \left(a^{r/2}-1\right)\left(a^{r/2}+1\right) = k \cdot N$ for some $k \in \mathbb{N}$

The factoring problem is reduced to finding the period r of $f(x) = a^x \pmod N$.

Example: $N = 15$, $a = 7$

x	0	1	2	3	4	5	...
f(x)	1	7	4	13	1	7	...

$r = 4$

$\gcd\left(7^{4/2}+1, 15\right) = 5$

$\gcd\left(7^{4/2}-1, 15\right) = 3$

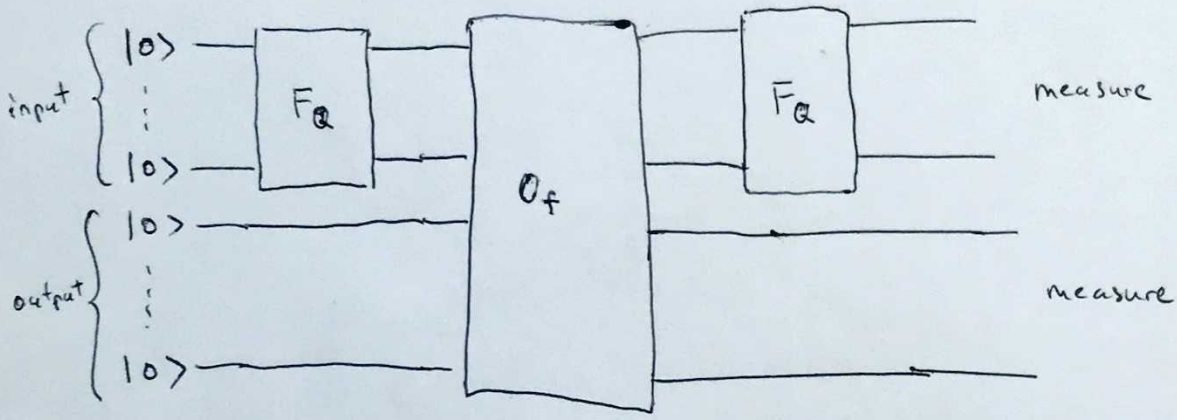
$15 = 5 \times 3$

The quantum period finding subroutine

Consider q "input" qubits & n "output" qubits, such that

$$N^2 < Q \leq 2N^2 \qquad n = \log N \qquad q = \log Q$$

The circuit consists of 2 QFTs and a „black-box" to compute $f(x)$



Applying the first QFT on $|0^q\rangle|0^n\rangle$ leads to

$$\frac{1}{\sqrt{Q}} \sum_{b=0}^{Q-1} |b\rangle|0^n\rangle$$

The „black-box" maps $|x\rangle|0^n\rangle \longrightarrow |x\rangle|f(x)\rangle$. Use the „black-box" to compute $f(x)$.

$$\frac{1}{\sqrt{Q}} \sum_{b=0}^{Q-1} |b\rangle|f(b)\rangle$$

The input and output qubits are now entangled.

Measuring the "output" register gives some value y. This implies that the "input" register collapses to a superposition of

$$|s\rangle,\ |s+r\rangle,\ |s+2r\rangle,\ \ldots,\ |s+mr\rangle \qquad \left(m \sim \frac{Q}{r}\right)$$

where $f(s) = y$, and $s < r$.

Ignoring the second register, we have in the first one

$$\frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} |jr+s\rangle$$

Applying the second QFT ($\omega_Q = e^{\frac{2\pi i}{Q}}$)

$$\hat{F}_Q \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} |jr+s\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} \frac{1}{\sqrt{Q}} \sum_{b=0}^{Q-1} \omega_Q^{b(jr+s)} |b\rangle =$$

$$= \frac{1}{\sqrt{mQ}} \sum_{b=0}^{Q-1} \omega_Q^{bs} \cdot \underbrace{\left(\sum_{j=0}^{m-1} \omega_Q^{j(rb)} \right)}_{\text{geometric sum}} |b\rangle = \frac{1}{\sqrt{mQ}} \sum_{b=0}^{Q-1} \omega_Q^{bs} |b\rangle \times \begin{cases} m, & \text{if } \omega_Q^{rb} = 1 \\ \dfrac{1 - \omega_Q^{mrb}}{1 - \omega_Q^{rb}}, & \text{if } \omega_Q^{rb} \neq 1 \end{cases}$$

States $|b\rangle$, for which $\omega_Q^{rb} \approx 1$, have a large amplitude. ~~Reffa~~

Call them "good" states.

Determining r

Let us consider a simple case, where r divides Q. ($m = Q/r$)

↳ States with $\frac{rb}{Q}$ integer will have $\omega_Q^{rb} = 1$, and their amplitude is $\left(\frac{m}{\sqrt{mQ}}\right)^2 = \frac{m}{Q} = \frac{1}{r}$. There are exactly r such basis states.

Therefore we will observe the „input" register in some state b, with

$$b = \frac{Q}{r} \cdot c \quad , \; c \text{ - uniform random variable in } \{0, \ldots, r-1\}$$

→ There are $\phi(r) \gtrsim \frac{k\,r}{\log\log(r)}$ numbers smaller than r that are coprime to r. Hence, after $O(\log\log N)$ repetitions of the procedure we would likely encounter at least on b, for which c is co-prime to r. For that b, we can obtain r as the denominator, writing b/Q in lowest terms.