Quantum Computing

Lecture $|10\rangle$:

The Quantum Fourier Transform and Phase Estimation - Shor's Algorithm (I)

Paolo Zuliani

Dipartimento di Informatica

Università di Roma "La Sapienza", Rome, Italy

Agenda

- Multiplying two Polynomials
- Discrete Fourier Transform
- Quantum Fourier Transform
- Quantum Algorithm for Phase Estimation

Multiplying Two Polynomials

Given two $(n-1)$-degree polynomials $p(x) = \sum_{i=0}^{n-1} a_i x^i$ and $q(x) = \sum_{j=0}^{n-1} b_j x^j$, we want to compute their product:

$$p(x)q(x) = \left(\sum_{i=0}^{n-1} a_i x^i\right)\left(\sum_{j=0}^{n-1} b_j x^j\right) = \sum_{i,j=0}^{n-1} a_i x^{i+j} b_j \ .$$

We can sum the monomials of the same degree, so:

$$p(x)q(x) = \sum_{k=0}^{2n-2} x^k \left(\sum_{j=0}^{k} a_j b_{k-j}\right) \qquad (a_j = b_j = 0 \quad \text{if } j < 0 \text{ or } j \geqslant n)$$

Multiplying Two Polynomials

By adding an extra 0 term to the sum, we can write:

$$p(x)q(x) = \sum_{k=0}^{2n-1} x^k \left(\sum_{j=0}^{k-1} a_j b_{k-j} \right) = \sum_{k=0}^{2n-1} x^k c_k$$

Definition

The **convolution** $c = a \circledast b$ of vectors $(a_0, \ldots, a_{n-1})$ and $(b_0, \ldots, b_{n-1})$ is the $2n$-element vector defined by:

$$c_k = \sum_{j=0}^{k-1} a_j b_{k-j}$$

for $k = 0, \ldots, 2n-1$ and $a_j = b_j = 0$ if $j < 0$ or $j \geqslant n$.

The Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is defined in general over an arbitrary commutative ring $(R, \cdot, +, 0, 1)$.

Definition

A $w \in R$ is a **principal** ***n*****-th root of unity** if:

1. $w \neq 1$
2. $w^n = 1$
3. $\sum_{j=0}^{n-1} w^{j \cdot p} = 0$ for $p = 1, \ldots, n-1$.

The powers $w^0, w^1, \ldots, w^{n-1}$ are the n-th roots of unity.

Usually, we take $w = e^{\frac{2\pi i}{n}}$ for $n \in \mathbb{N}$.

The Discrete Fourier Transform

Let R be a commutative ring and $w \in R$ a principal n-th root of unity.

Define the $n \times n$ matrix A:

$$A_{ij} = w^{i \cdot j} \qquad \text{for } i, j = 0, \ldots, n-1.$$

Definition

Let $a \in R^n$ be a vector in a commutative ring R.

The vector $F(a) = A \cdot a$ is the **Discrete Fourier Transform** (DFT) of a.

Proposition

The **inverse** *DFT is the matrix A^{-1} given by $A^{-1}_{ij} = \frac{1}{n} w^{-i \cdot j}$, for $i, j = 0, \ldots, n-1$.*

Back to Polynomial Multiplication

The i-th element of $F(a)$ is

$$\sum_{k=0}^{n-1} a_k w^{i \cdot k}$$

the DFT of vector a thus converts a polynomial from its coefficient representation to its value representation at the points $w^0, w^1, \ldots, w^{n-1}$.

The inverse DFT just does the opposite!

Theorem (Convolution Theorem)

Let $a = (a_0, \ldots, a_{n-1}, 0, \ldots, 0) \in R^{2n}$, $b = (b_0, \ldots, b_{n-1}, 0, \ldots, 0) \in R^{2n}$ two $2n$ vectors in R, and let $F(a), F(b)$ be their DFT. We have that:

$$a \circledast b = F^{-1}(F(a) \cdot F(b)).$$

The Discrete Fourier Transform

- The DFT is used for integer multiplication, signal processing (*e.g.*, speech recognition, audio compression), etc. It is sometimes easier to study a signal in a different domain (*e.g.*, frequency instead of time).
- When n is a power of 2, the **Fast Fourier Transform** algorithm computes the DFT in $O(n \log n)$ operations instead of the "naive" $O(n^2)$.
- For a deeper and clear treatment of the DFT see Chapter 7 of "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, and Ullman.

The Quantum Fourier Transform

Definition

The QFT maps each basis state $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ as follows

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \qquad (\text{note } i = \sqrt{-1})$$

Equivalently, for a generic vector:

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle$$

where $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$.

The Quantum Fourier Transform

The QFT maps each basis state $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ as follows

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle$$

so it can be written as

$$QFT = \sum_{j=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \right) \langle j|$$

Proposition

The QFT is a **unitary** *operator*, i.e., $QFT\ QFT^\dagger = QFT^\dagger\ QFT = I$. *Note that*

$$QFT^\dagger = \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi ijk/N} \langle k| \right)$$

The Quantum Fourier Transform: Unitarity

Let us show that the QFT is unitary:

$$
\begin{aligned}
QFT^\dagger \; QFT &= \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} e^{-2\pi i j r/N} \langle r| \right) \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{2\pi i s k/N} |s\rangle \right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{-2\pi i j r/N} \langle r| \right) \left(\sum_{s=0}^{N-1} e^{2\pi i s k/N} |s\rangle \right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r,s=0}^{N-1} e^{2\pi i(-jr+ks)/N} \langle r|s\rangle \right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N} \right) \langle k| \qquad [\text{recall } \langle r|s\rangle = \delta_{rs}]
\end{aligned}
$$

The Quantum Fourier Transform: Unitarity

$$
\begin{aligned}
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N} \right) \langle k| \\
&= \frac{1}{N} \sum_{j=k;j=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} 1 \right) \langle k| + \frac{1}{N} \sum_{j,k=0;j\neq k}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N} \right) \langle k| \\
&= \sum_{j=0}^{N-1} |j\rangle\langle j| + \frac{1}{N} \sum_{j,k=0;j\neq k}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} (e^{2\pi i(k-j)/N})^r \right) \langle k| \\
&= I + \frac{1}{N} \sum_{j,k=0;j\neq k}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} (e^{2\pi i(k-j)/N})^r \right) \langle k|
\end{aligned}
$$

The Quantum Fourier Transform: Unitarity

$$= I + \frac{1}{N} \sum_{j,k=0; j \neq k}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} (e^{2\pi i (k-j)/N})^r \right) \langle k|$$

$$= I + \frac{1}{N} \sum_{j,k=0; j \neq k}^{N-1} |j\rangle \left(\frac{1 - (e^{2\pi i (k-j)/N})^N}{1 - e^{2\pi i (k-j)/N}} \right) \langle k| \qquad \left[\text{recall } \sum_{i=0}^{N-1} \rho^i = \frac{1-\rho^N}{1-\rho} \text{ for } \rho \neq 1\right]$$

$$= I + \frac{1}{N} \sum_{j,k=0; j \neq k}^{N-1} |j\rangle \frac{1 - e^{2\pi i (k-j)}}{1 - e^{2\pi i (k-j)/N}} \langle k| \qquad [\forall j \neq k \in \{0, \ldots, N-1\}, e^{2\pi i (k-j)} = 1]$$

$$= I$$

Exercise: prove $QFT \, QFT^\dagger = I$.

The Quantum Fourier Transform: Quantum Circuit

An **equivalent** QFT definition (assuming $N = 2^n$, hence n qubits):

$$|j_1 \dots j_n\rangle \xrightarrow{QFT} \frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.j_1 \cdots j_n}|1\rangle)}{2^{n/2}}$$

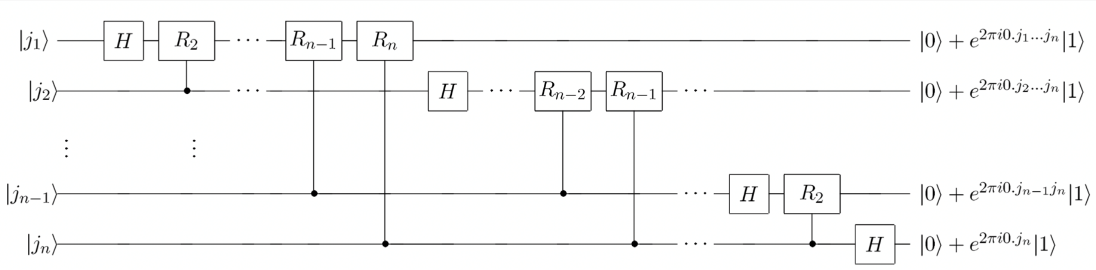
where $j_1, \dots, j_n$ are bits, and the **binary fraction**

$$0.j_l j_{l+1} j_m = \frac{j_l}{2} + \frac{j_{l+1}}{4} + \cdots + \frac{j_m}{2^{m-l+1}}$$

Example:

$$0.1101 = \frac{1}{2} + \frac{1}{4} + \frac{1}{2^4}$$

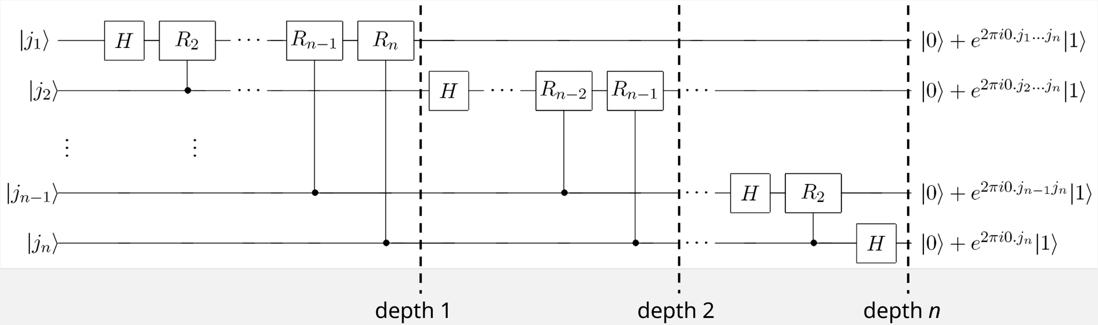
The Quantum Fourier Transform: Quantum Circuit



where H is the usual Hadamard and the controlled-R_k gates are defined on

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

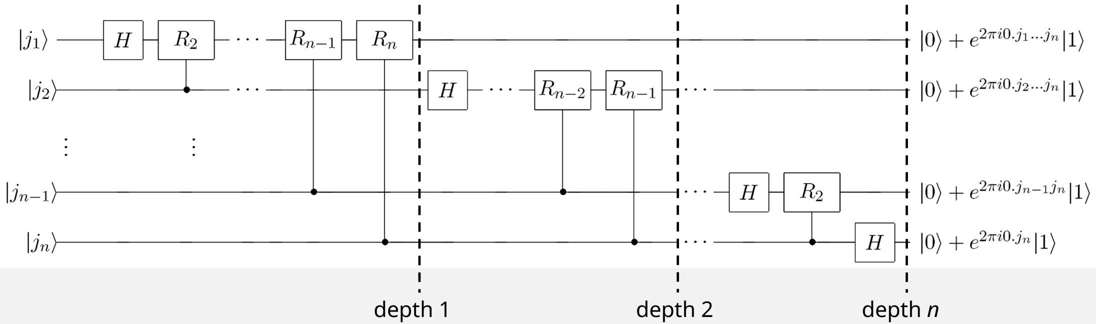
The Quantum Fourier Transform: Quantum Circuit



State at depth 1: $\frac{1}{2^{1/2}}(|0\rangle + e^{2\pi i 0.j_1 \cdots j_n}|1\rangle)|j_2 \dots j_n\rangle$

State at depth 2: $\frac{1}{2^{2/2}}(|0\rangle + e^{2\pi i 0.j_1 \cdots j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_2 \cdots j_n}|1\rangle)|j_3 \dots j_n\rangle$

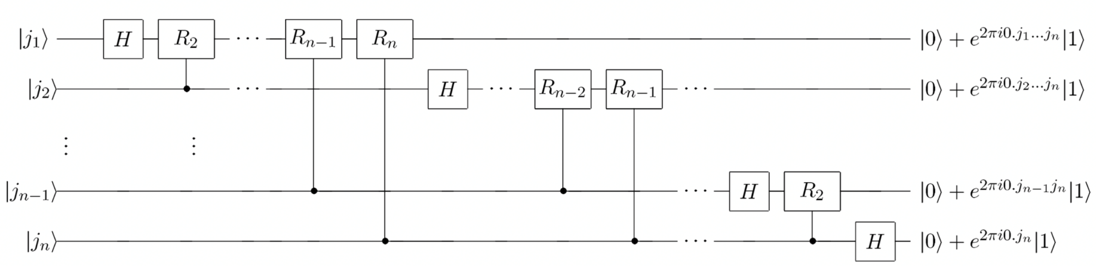
The Quantum Fourier Transform: Quantum Circuit



The state at depth n has the correct terms, but in the wrong order! (Remember the tensor product is NOT commutative.)

We need to **swap** the qubits, which can be done unitarily, of course.

The Quantum Fourier Transform: Complexity



- Quantum circuit has $O(n^2)$ gates. Best classical circuit needs $O(n2^n)$ gates.
- Looks great! Is it?

$$\sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{QFT} \sum_{k=0}^{N-1} y_k |k\rangle \qquad \left(\text{where } y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i jk/N}\right)$$

- We want the DFT coefficients y_k's, but they are encoded in the amplitudes!

Phase Estimation

Let's see an application of the QFT.

A previous exercise: the eigenvalues of a unitary operator are complex numbers of **modulus 1**.

This means that any eigenvalue of a unitary operator can be written as $e^{2\pi i\varphi}$ for some real $\varphi \in [0, 1]$.

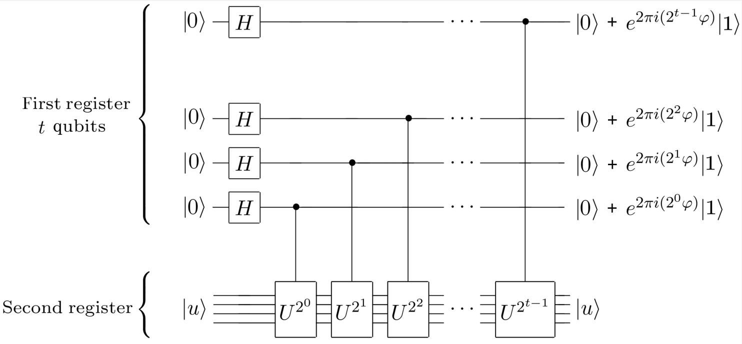
Definition (Phase Estimation Problem)

Let $\lambda = e^{2\pi i\varphi}$ be an eigenvalue of a unitary operator U. Find φ.

This problem can be solved quite easily with the QFT.

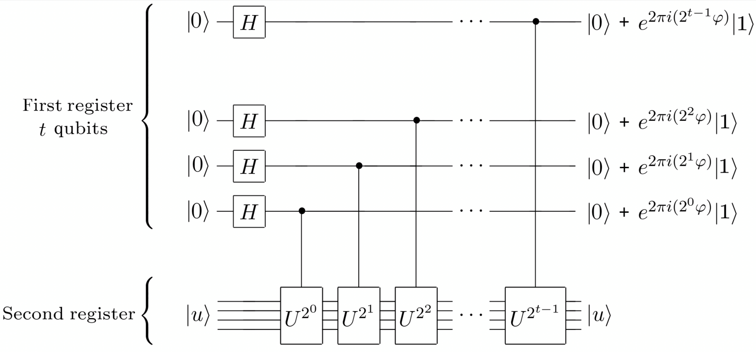
Quantum Phase Estimation Algorithm

Let u be an eigenvector associated to the unknown eigenvalue $e^{2\pi i\varphi}$ of a unitary operator U, *i.e.*, $U\,|u\rangle = e^{2\pi i\varphi}\,|u\rangle$. Consider the circuit below for some natural $t > 0$:



A control-U^{2^k} gate conditionally applies $U^{2^k} = \underbrace{U \cdots U}_{2^k \, times}$ to the second qubit register.

Quantum Phase Estimation Algorithm



The state of the t qubits at the end of the QPE circuit is:

$$\frac{1}{2^{t/2}}(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle) \otimes (|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle) = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i k\varphi}|k\rangle$$

Quantum Phase Estimation Algorithm

Suppose now that φ can be written **exactly** with t bits:

$$\boxed{\varphi = 0.\varphi_1 \ldots \varphi_t}$$

Then, the state at the end of the QPE circuit is:

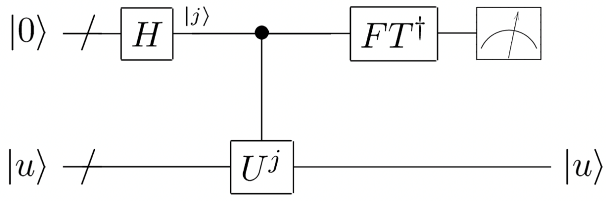
$$\frac{1}{2^{t/2}}(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.\varphi_{t-1}\varphi_t}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.\varphi_1\varphi_2\ldots\varphi_t}|1\rangle)$$

which is **precisely** the final state of the QFT circuit (after the swap)!

Therefore, we apply the **inverse** QFT circuit at the end of the QPE circuit and then measure to obtain the sought phase $|\varphi_1 \ldots \varphi_t\rangle$ with probability 1!

Quantum Phase Estimation Algorithm

The final quantum circuit for solving phase estimation is thus:



What if φ is not expressible in exactly t bits?

Proposition

To estimate φ with n **bits of precision** *and* **success probability** *at least $1 - \epsilon$, it is sufficient to use the QPE circuit with $r = n + \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right) \right\rceil$ qubits.*