Quantum algorithms (CO 781/CS 867/QIC 823, Winter 2011)
Andrew Childs, University of Waterloo

LECTURE 2: The abelian QFT and phase estimation

Quantum Fourier transform

Perhaps the most important unitary transformation in quantum computing is the quantum Fourier transform (QFT). Later, we will discuss the QFT over arbitrary finite groups, but for now we will focus on the case of an abelian group $G$. Here the transformation is

$$F_G := \frac{1}{\sqrt{|G|}} \sum_{x \in G} \sum_{y \in \hat{G}} \chi_y(x) |y\rangle \langle x|$$

(1)

where $\hat{G}$ is a complete set of characters of $G$, and $\chi_y(x)$ denotes the $y$th character of $G$ evaluated at $x$. (You can verify that this is a unitary operator using the orthogonality of characters.) Since $G$ and $\hat{G}$ are isomorphic, we can label the elements of $\hat{G}$ using elements of $G$, and it is often useful to do so.

The simplest QFT over a family of groups is the QFT over $G = \mathbb{Z}_2^n$. The characters of this group are $\chi_y(x) = (-1)^{x \cdot y}$, so the QFT is simply

$$F_{\mathbb{Z}_2^n} = \frac{1}{\sqrt{2^n}} \sum_{x,y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} |y\rangle \langle x| = H^\otimes n.$$

(2)

You have presumably seen how this transformation is used in the solution of Simon’s problem.

QFT over $\mathbb{Z}_2^n$

A more complex quantum Fourier transform is the QFT over $G = \mathbb{Z}_2^n$:

$$F_{\mathbb{Z}_2^n} = \frac{1}{\sqrt{2^n}} \sum_{x,y \in \mathbb{Z}_2^n} \omega_{2^n}^{xy} |y\rangle \langle x|$$

(3)

where $\omega_m := \exp(2\pi i/m)$ is a primitive $m$th root of unity. To see how to realize this transformation by a quantum circuit, it is helpful to represent the input $x$ as a string of bits, $x = x_{n-1} \ldots x_1 x_0$, and to consider how an input basis vector is transformed:

$$|x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} \omega_{2^n}^{xy} |y\rangle$$

(4)

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} \omega_{2^n}^{x(y \sum_{k=0}^{n-1} y_k 2^k)} |y_{n-1} \ldots y_1 y_0\rangle$$

(5)

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} \prod_{k=0}^{n-1} \omega_{2^n}^{y_k 2^k} |y_{n-1} \ldots y_1 y_0\rangle$$

(6)

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{k=0}^{n-1} \sum_{y_k \in \mathbb{Z}_2} \omega_{2^n}^{y_k 2^k} |y_k\rangle$$

(7)

$$= \bigotimes_{k=0}^{n-1} |z_k\rangle$$

(8)
where

\begin{equation}
|z_k\rangle := \frac{1}{\sqrt{2}} \sum_{y_k \in \mathbb{Z}_2} \omega_{2^n}^{x_k y_k/2^k} |y_k\rangle
\end{equation}

(9)

\begin{equation}
= \frac{1}{\sqrt{2}} (|0\rangle + \omega_{2^n}^{x_{2^k}} |1\rangle)
\end{equation}

(10)

\begin{equation}
= \frac{1}{\sqrt{2}} (|0\rangle + \omega_{2^n}^{\sum_{j=0}^{n-1} x_j 2^j} |1\rangle) 
\end{equation}

(11)

\begin{equation}
= \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i (x_0 2^{k-n} + x_1 2^{k-n+1} + \cdots + x_{n-1-k} 2^{-1})} |1\rangle) .
\end{equation}

(12)

(A more succinct way to write this is $|z_k\rangle = \frac{1}{\sqrt{2}} (|0\rangle + \omega_{2^n}^x |1\rangle)$, but the above expression is more helpful for understanding the circuit.) In other words, $F|x\rangle$ is a tensor product of single-qubit states, where the $k$th qubit only depends on the $k$ least significant bits of $x$.

This decomposition immediately gives a circuit for the QFT over $\mathbb{Z}_{2^n}$. Let $R_k$ denote the single-qubit unitary operator

\begin{equation}
R_k := \begin{pmatrix} 1 & 0 \\ 0 & \omega_{2^k} \end{pmatrix}
\end{equation}

(13)

Then the circuit can be written as follows:

\begin{align*}
|x_0\rangle & \quad \cdots \quad |x_0\rangle \\
|x_1\rangle & \quad \cdots \quad |x_1\rangle \\
\vdots & \quad \ddots \quad \vdots \\
|x_{n-3}\rangle & \quad \cdots \quad |x_{n-3}\rangle \\
|x_{n-2}\rangle & \quad |R_{n-2}\rangle \cdots \quad |R_{n-2}\rangle \\
|x_{n-1}\rangle & \quad |R_{n}\rangle \cdots \quad |R_{n}\rangle
\end{align*}

This circuit uses $O(n^2)$ gates. However, there are many rotations by small angles that do not affect the final result very much. If we simply omit the gates $R_k$ with $k = \Omega(\log n)$, then we obtain a circuit with $O(n \log n)$ gates that implements the QFT with precision $1/poly(n)$.

### Phase estimation

Aside from being directly useful in quantum algorithms, such as Shor’s algorithm, The QFT over $\mathbb{Z}_{2^n}$ provides a useful quantum computing primitive called phase estimation. In the phase estimation problem, we are given a unitary operator $U$ (either as an explicit circuit, or as a black box that lets us apply a controlled-$U^j$ operation for integer values of $j$). We are also given a state $|\phi\rangle$ that is promised to be an eigenvector of $U$, namely $U|\phi\rangle = e^{i\phi}|\phi\rangle$ for some $\phi \in \mathbb{R}$. The goal is to output an estimate of $\phi$ to some desired precision.

The procedure for phase estimation is straightforward. To get an $n$-bit estimate of $\phi$, prepare the quantum computer in the state

\begin{equation}
\frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_{2^n}} |x, \phi\rangle
\end{equation}

(14)
apply the operator
\[ \sum_{x \in \mathbb{Z}_{2^n}} |x\rangle \langle x| \otimes U^x \] (15)
to give the state
\[ \frac{1}{\sqrt{2^m}} \sum_{x \in \mathbb{Z}_{2^n}} e^{i\phi x} |x, \phi\rangle, \] (16)
apply an inverse Fourier transform on the first register, and measure. If the binary expansion of \( \phi/2\pi \) terminates after at most \( n \) bits (i.e., if \( \phi = 2\pi y/2^n \) for some \( y \in \mathbb{Z}_{2^n} \)), then the state (16) is \( F_{2^n} |y\rangle \otimes |\phi\rangle \), so the result is guaranteed to be the binary expansion of \( \phi/2\pi \). In general, we obtain a good approximation with high probability. (You have probably seen this kind of calculation before, and we will see the details of a similar calculation when we discuss period finding.)

**QFT over \( \mathbb{Z}_N \) and over a general finite abelian group**

One useful application of phase estimation is to implement the QFT over an arbitrary cyclic group \( \mathbb{Z}_N \):
\[ F_{\mathbb{Z}_N} = \frac{1}{\sqrt{N}} \sum_{x,y \in \mathbb{Z}_N} \omega_{\mathbb{Z}_N}^{xy} |y\rangle \langle x|. \] (17)
The circuit we derived using the binary representation of the input and output only works when \( N \) is a power of two (or, with a slight generalization, some other small integer). But there is a simple way to realize \( F_{\mathbb{Z}_N} \) (approximately) using phase estimation.

We would like to perform the transformation that maps \( |x\rangle \mapsto |\tilde{x}\rangle \), where \( |\tilde{x}\rangle := F_{\mathbb{Z}_N} |x\rangle \) denotes a Fourier basis state. (By linearity, if the transformation acts correctly on a basis, it acts correctly on all states.) It is straightforward to perform the transformation \( |x, 0\rangle \mapsto |x, \tilde{x}\rangle \); then it remains to erase the register \( |x\rangle \) from such a state.

Consider the unitary operator that adds 1 modulo \( N \):
\[ U := \sum_{x \in \mathbb{Z}_N} |x + 1\rangle \langle x|. \] (18)
The eigenstates of this operator are precisely the Fourier basis states \( |\tilde{x}\rangle := F_{\mathbb{Z}_N} |x\rangle \), since (as a simple calculation shows)
\[ F_{\mathbb{Z}_N}^\dagger U F_{\mathbb{Z}_N} = \sum_{x \in \mathbb{Z}_N} \omega_{\mathbb{Z}_N}^x |x\rangle \langle x|. \] (19)
Thus, using phase estimation on \( U \) (with \( n \) bits of precision where \( n = O(\log N) \)), we can perform the transformation
\[ |\tilde{x}, 0\rangle \mapsto |\tilde{x}, x\rangle \] (20)
(actually, phase estimation only gives an approximation of \( x \), so we implement this transformation only approximately). By running this operation in reverse, we can erase \( |x\rangle \), and thereby produce the desired QFT.

Given the Fourier transform over \( \mathbb{Z}_N \), it is straightforward to implement the QFT over an arbitrary finite abelian group: any finite abelian group can be written as a direct product of cyclic factors, and the QFT over a direct product of groups is simply the tensor product of QFTs over the individual groups.