CSE 599d - Quantum Computing
Quantum Phase Estimation and Arbitrary Size Quantum Fourier Transforms

Dave Bacon
Department of Computer Science & Engineering, University of Washington

What use is the quantum Fourier transform? Well we've already seen Jordan's algorithm which was like the Bernstein-Vazirani problem. But the most important use for the quantum Fourier transform was discovered by Peter Shor when he showed how to us it to efficiently factor numbers. In these notes we'll develop the quantum machinery necessary to get to the point where we can understand Shor's algorithm.

I. QUANTUM PHASE ESTIMATION ALGORITHM

Let's look at the quantum Fourier transform. It performs the transform

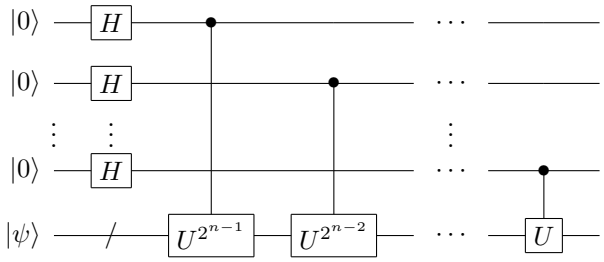
$$|x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{-xy} |y\rangle \tag{1}$$

Or, looking at it in a different manner, transforms the states

$$\frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle \rightarrow |x\rangle \tag{2}$$

Now what we learned from the Bernstein-Vazirani problem was that if we could get the information about the function into the ± 1 phase of each state, then we could use this, for the Hadamard basis states, to find a hidden string s for the function $f(x) = s \cdot x$. What might the analogy be for the quantum Fourier transform? Well the first thing to notice is that there are roots of unity which are not just ± 1 here. Intuitively this means that a phase kickback trick will have to do phase kickback with roots of unity. Recall that the eigenvalues of unitary matrices are roots of unity. This leads us naturally to the quantum phase estimation algorithm.

Suppose that you are in the following situation. You can prepare an eigenstate $|\psi\rangle$ of a unitary operator U. Further, you have the ability to apply a controlled-U^{2^j} operator. How can you use these tools to estimate the eigenvalue of this unitary. First recall that unitary matrices have eigenvalues which are roots of unity. Let's call the eigenvalue corresponding to the eigenvector $|\psi\rangle$, $e^{i2\pi\phi}$, where $0 \leq \phi < 1$. Our goal is to find an estimate of ϕ. The quantum phase estimation algorithm performs exactly this task. How does this algorithm work? It begins by preparing doing a phase kickback trick. Consider the following quantum circuit on n qubits,



(3)

Recall that $U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$, so $U^{2^j}|\psi\rangle = e^{2\pi i\phi 2^j}|\psi\rangle$. Thus if we input $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ into the control of one of these controlled-U^{2^j} gates which itself acts on an the eigenstate $|\psi\rangle$, then this will perform the transform

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^j}|1\rangle)|\psi\rangle \tag{4}$$

Therefore we calculate that the output of the above circuit is

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 2^{n-1}\phi}|1\rangle) \otimes (|0\rangle + e^{2\pi i 2^{n-2}\phi}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 2^0 \phi}|1\rangle) \tag{5}$$

Recall our notation for a binary fraction

$$0.x_l x_{l+1}...x_n = \frac{x_l}{2} + \frac{x_{l+1}}{4} + \cdots + \frac{x_n}{2^{n-l+1}} \tag{6}$$

Suppose that ϕ has such an n bit expansion. Then we can express the output of the circuit as

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{\frac{2\pi i 0.\phi_1}{2^n}}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.\phi_1\phi_2}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.\phi_1\phi_2...\phi_n}|1\rangle) \tag{7}$$

Now it may be that ϕ does not have a n bit binary fraction expansion. We will treat this case in a second. First, to understand what is going on, let's just assume that ϕ does have exactly a n bit binary fraction expansion. How do we determine ϕ? Simple. We simply perform the Fourier transform on the output of this circuit. Notice that the Fourier transform will take the output of the phase kickback state above to the state

$$|\phi_1, \phi_2, \ldots, \phi_n\rangle \tag{8}$$

What happens to the phase estimation algorithm when we are not given a ϕ with an exact n bit binary fraction expansion? In particular, suppose that we wish to obtain ϕ to n bits of accuracy with some probability of failing ϵ. How do we do this?

Well the output of the phase kickback circuit is

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i \phi x}|x\rangle \tag{9}$$

Evaluating the QFT on this state produces the state

$$\frac{1}{2^n} \sum_{y,x=0}^{2^n-1} e^{-2\pi i \frac{xy}{2^n} + 2\pi i \phi x}|y\rangle \tag{10}$$

The probability that we observe y is therefore

$$Pr(y) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} e^{-2\pi i \frac{xy}{2^n} + 2\pi i \phi x} \right|^2 = \frac{1}{2^{2n}} \left| \sum_{x=0}^{2^n-1} e^{2\pi i x(-\frac{y}{2^n} + \phi)} \right|^2 \tag{11}$$

We can evaluate this sum exactly:

$$Pr(y) = \frac{1}{2^{2n}} \left| \frac{1 - r^{2^n}}{1 - r} \right|^2 \tag{12}$$

where $r = e^{2\pi i(\phi - \frac{y}{2^n})}$. Note that if $\phi = \frac{y}{2^n}$, then this equation has a division by zero, and in this we find that only the correct y is measured. We will assume this is not the case. Suppose that the closest n bit approximation to ϕ is $\nu = 0.\nu_1\nu_2 \ldots \nu_n$. This differs from ϕ by an error $0 < |\delta| \leq 2^{n+1}$. Then we see that $r = e^{2\pi i \delta}$.

Let's bound the probability that we get the correct probability. First notice that $|1 - r^{2^n}|$ is the length of a chord from 1 to r^{2^n} in the complex plane. The ratio of the length of the minor arc to the length of this chord is most $\frac{\pi}{2}$. This yields,

$$\frac{2\pi\delta 2^n}{|1 - r^{2^n}|} \leq \frac{\pi}{2} \tag{13}$$

or

$$|1 - r^{2^n}| \geq 4\delta 2^n \tag{14}$$

What about $|1 - r|$ Again we can consider the chord from 1 to r in the complex plane. Then this ratio is at least 1,

$$\frac{2\pi\delta}{|1 - r|} \geq 1 \tag{15}$$

Putting these together we obtain the probability of succeeding of

$$Pr \geq \frac{1}{2^{2n}} \frac{(4\delta 2^n)^2}{(2\pi)^2} = \frac{4}{\pi^2} > 0.4 \tag{16}$$

This implies that we will get the correct answer with greater than a constant probability.

Actually we can do even better than this because the probability that we will obtain values which are far from the correct value are very small. To show this, we note that the probability of our getting the incorrect outcome can be calculated using $|\delta| > \frac{1}{2^{n+1}}$ Then using $|1 - r^{2^n}| \leq 2$ and $|1 - r| \geq 4\delta$ (like the r^{2^n} before but now for r instead), we see that

$$Pr(\delta) \leq \frac{1}{2^{2m}} \frac{2^2}{(4\delta)^2} \tag{17}$$

If $\delta = \frac{c}{2^n}$, this becomes

$$Pr(c) \leq \frac{1}{4c^2} \tag{18}$$

Thus we see that the probability of getting worse approximations falls off as we move further and further away from the correct value.

II. KITAEV'S METHOD FOR GENERAL QUANTUM FOURIER TRANSFORMS

So far I have showed you a circuit for computing the quantum Fourier transform over n qubits. But what if instead of working with these qubits we want to treat n qubits as a register holding a number between 0 and $N - 1 < 2^n$ and then perform the quantum Fourier transform over this vector of N amplitudes? Kitaev has given a very nice method for approximating just such a quantum Fourier transform.

Recall that we are working on n qubits. Define the state

$$|\Phi_x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{-xy} |y\rangle \tag{19}$$

Then the QFT we wish to enact will take

$$|x\rangle \rightarrow |\Phi_x\rangle \tag{20}$$

Kitaev's procedure for this proceeds in two steps. First an ancilla register is prepared as $|0\rangle$ and then the transform

$$|x\rangle \otimes |0\rangle \rightarrow |x\rangle \otimes |\Phi_x\rangle \tag{21}$$

is enacted followed by

$$|x\rangle \otimes |\Phi_x\rangle \rightarrow |0\rangle \otimes |\Phi_x\rangle. \tag{22}$$

Notice that the inverse of the last step is the phase estimation algorithm! In other words all we need to do to approximately produce the QFT is the first step in the transform.

How do we do this efficiently? Well first we prepare the state $|\Phi_0\rangle$. Then conditional on the first register we will transform $|x\rangle \otimes |\Phi_0\rangle = |x\rangle \otimes |\Phi_x\rangle$.

Okay, so how do we prepare $|\Phi_x\rangle$? Here is a procedure for producing any state

$$|\Phi\rangle = \sum_{x_1,\ldots,x_n \in \{0,1\}} \alpha_{x_1,\ldots,x_n} |x_1, \ldots, x_n\rangle \tag{23}$$

where the $\alpha_{x_1,\ldots,x_n}$ are positive real coefficients. Suppose that you measure this state in the computation basis. The probability that you obtain 0 for the first qubit is

$$\sum_{x_2,\ldots,x_n \in \{0,1\}} \alpha^2_{0,x_2,\ldots,x_n} \tag{24}$$

Call this α_0^2 and similarly define α_1^2 for the probability that we obtain 1 for the first qubit. Then the first rotation to perform is on the first qubit and is

$$|0\rangle \to \alpha_0|0\rangle + \alpha_1|1\rangle \tag{25}$$

What I've done here is only defined the rotation on the $|0\rangle$ basis state. Fortunately it has amplitudes whose square sum to unity, so we can choose its action arbitrarily on $|1\rangle$ to make it unitary. I will use this observation repeatedly for here on out. Now what do we do for the next qubit? The probability that the second qubit is 0 given that the first qubit is 0 is

$$\sum_{x_3,\ldots,x_n \in \{0,1\}} \alpha^2_{0,0,x_2,\ldots,x_n} \tag{26}$$

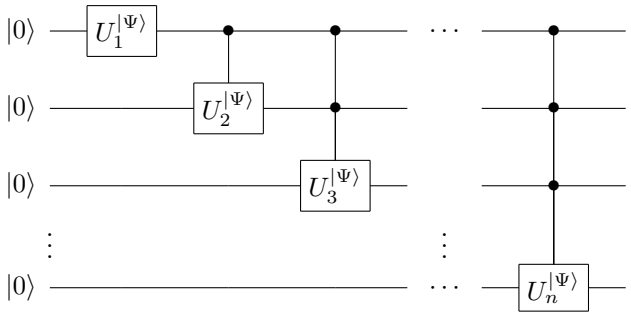
We can use this to define $\alpha^2_{0|0}$, which stands for the probability that the second qubit is measured 0 given that the first qubit was measured 0. In a similar fashion we can define $\alpha^2_{x_2|x_1}$ which is the probability that the second qubit was measured as x_2 given that the first qubit was measured x_1. Then the next operation we need to apply is the conditional rotation:

$$|x_1, 0\rangle \to |x_1\rangle \otimes (\alpha_{0|x_1}|0\rangle + \alpha_{1|x_1}|1\rangle \tag{27}$$

We can carry on in a similar fashion for all n qubits. First applying the single qubit rotation, then performing a controlled single qubit rotation, then perform a doubly controlled-single qubit rotation, etc. If we define $\alpha^2_{x_k|x_1,\ldots,x_{k-1}}$ as the probability of obtaining outcome x_k given that we have measured $x_1, \ldots, x_{k-1}$, then this allows us to define the $k-1$ qubit controlled single qubit rotation

$$|x_1, \ldots, x_{k-1}, 0\rangle \to |x_1, \ldots, x_{k-1}\rangle \otimes (\alpha_{0|x_1,\ldots,x_{k-1}}|0\rangle + \alpha_{1|x_1,\ldots,x_{k-1}}|1\rangle \tag{28}$$

Why does this work? One way to see why this works is that it is guaranteed to obtain the correct probabilities for all of the conditional probabilities for obtaining measurement outcome x_k given measurement outcomes $x_1, \ldots, x_{k-1}$. Since the rotation only transform into positive real coefficients, this implies that the amplitudes for the state so produced must match those of $|\Phi\rangle$. A circuit for this transform will be of the form

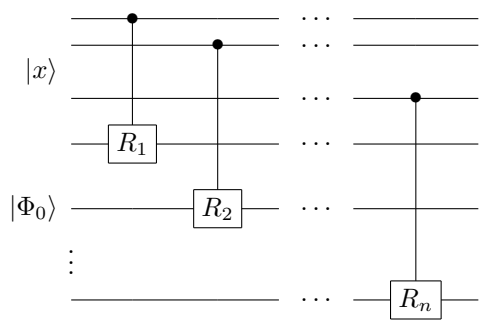


(29)

We can use the above construction to prepare the state $|\Phi_0\rangle$. How do we then perform the transform $|x\rangle|\Phi_0\rangle \to |x\rangle|\Phi_x\rangle$? This part is easy. We just perform a conditional phase rotation between the qubits of the two registers. Recalling our definition of R_k,

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix} \tag{30}$$

then the circuit to do this is



(31)

Thus we see that we can approximately implement the QFT for any N.

Acknowledgments

The diagrams in these notes were typeset using Q-circuit a latex utility written by graduate student extraordinaires Steve Flammia and Bryan Eastin.