Ph/CS 219A
Quantum Computation

Lecture 14. Period Finding

Last time we discussed query complexity relative to an oracle in a setting where one can query the oracle either classically (with computational basis states, i.e., bit strings) or quantumly (with coherent superpositions of computational basis states).

We showed that Simon's problem for a function f taking n bits to n bits can be solved with $O(n)$ quantum queries, while the number of randomized classical queries required is exponential in n: $BPP^O \neq BQP^O$, where O is Simon's oracle.

Though Simon's problem has no known practical applications, today we will discuss a related oracle problem that does have applications: finding the period of a function f. For this problem, too, an exponential separation between classical and quantum query complexity can be established.

The period finding algorithm is a key ingredient in Shor's algorithm for factoring integers, as we'll see in Lecture 15.

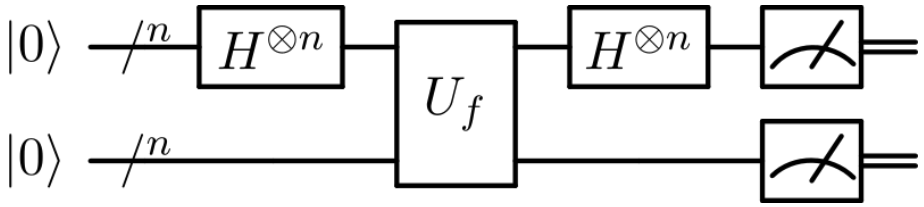
See Chapter 6 of the Lecture Notes. But I'll be organizing the material differently than in those notes.

Period Finding

Simon's Problem:

$f:\{0,1\}^n \to \{0,1\}^n$. We are promised that f is a 2-to-1 function such that $f(x)=f(y)$ iff $x \oplus y \in \{0,a\}$.

The problem is to find a. We can restate this as a decision problem: Is f 1-1, or is it 2-1, with the above structure.



We saw that an exponential number of queries are needed classically, but O(n) quantum queries suffice.

Period Finding is an analogous problem:

$f:\mathbb{Z} \to \{0,1\}^m$. We are promised that $f(x)=f(y)$ iff $x-y = r \times \text{integer}$, where $r \le M$.

f is periodic with period $r \le M$, and it is 1-1 on its period. The problem is to find r. The input size is $\log_2 M$.

Formulated as a decision problem: YES, if there is a period $\le M$. NO, if f is 1-1 for $0 \le x \le M$.

The decision problem is in NP^O, since $r < M$ is a witness; we can easily check $f(x) = f(x+r)$. But it is not in BPP^O. For each output value of f, a fraction r of all input values is mapped to that output. Given two random inputs, the probability both are mapped to the same output is $1/r$, exponentially small in the input size.

For $r^{1/4}$ queries, success probability is $P_{\text{success}} \le \frac{1}{2} r^{1/4}(r^{1/4}-1)/r < r^{-1/2}$ (exp small for exp large no. of queries).

Period Finding

Period Finding:

$f:\mathbb{Z}\to\{0,1\}^m$. We are promised that $f(x)=f(y)$ iff $x-y=r\times\text{integer}$, where $r\le M$.

f is periodic with period $r\le M$, and it is 1-1 on its period. The problem is to find r.

The input size is $\log_2 M$. Formulated as a decision problem:

YES, if there is a period $\le M$. NO, if f is 1-1 for $x\le M$.

Claim: the quantum query complexity of Period Finding is polylog *M* (actually, it's a constant). Furthermore, the quantum post-processing needed to identify the period is also efficient. Thus, a quantum computer can efficiently find the period of any efficiently computable function.

The idea of the Period Finding algorithm is similar to the solution to Simon's problem, except that instead of applying a bitwise Hadamard to the input register after the query, we apply the quantum Fourier transform.

$$\text{QFT}_N:|x\rangle\to\frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}e^{2\pi iyx/N}\,|y\rangle.$$

For *n* qubits, $N=2^n$. We will see that the QFT can be executed by a quantum circuit with size polylog *N* = poly *n* (actually quadratic in n). That the QFT is efficient is not needed for the analysis of the query complexity of Period Finding, but is important for practical applications.

Period Finding

$f:\mathbb{Z}\to\{0,1\}^m$. We are promised that $f(x)=f(y)$ iff $x-y=r\times\text{integer}$, where $r\le M$.

f is periodic with period $r\le M$, and it is 1-1 on its period. The problem is to find r.

As in Simon's algorithm, query in superposition, and then measure the output register.

$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\otimes|0\rangle\to\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\otimes|f(x)\rangle$. After measuring the output register, state of input register is

$\frac{1}{\sqrt{A}}\sum_{j=0}^{A-1}|x_0+jr\rangle,\quad 0\le x_0<r,\quad A=\text{least integer}\ge N/r.$

Throw away the output and apply to input register $\text{QFT}_N:|x\rangle\to\frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}e^{2\pi iyx/N}|y\rangle$.

$\Rightarrow\quad\frac{1}{\sqrt{NA}}\sum_{y=0}^{N-1}e^{2\pi ix_0y/N}\sum_{j=0}^{A-1}e^{2\pi ijry/N}|y\rangle$. Then measure in the computational basis:

$\Rightarrow\quad\text{Prob}(y)=\frac{A}{N}\left|\frac{1}{A}\sum_{j=0}^{A-1}e^{2\pi ijry/N}\right|^2$. (Probability distribution does not depend on x_0.)

This distribution is strongly peaked around values of *y* that convey useful information about the period *r*. To see how this works, suppose for the moment that *r* divides *N*.

Period Finding

$$\text{Prob}(y) = \frac{A}{N}\left|\frac{1}{A}\sum_{j=0}^{A-1} e^{2\pi i j r y/N}\right|^2, \quad A = N/r = \text{integer}.$$

If $y = \frac{N}{r} \times \text{integer}$, then $e^{2\pi i j r y/N} = 1 \Rightarrow$

$$\text{Prob}(y) = \frac{1}{r}\left|\frac{1}{A}\sum_{j=0}^{A-1} e^{2\pi i j y/A}\right| = \begin{cases} \frac{1}{r} & y = A\cdot(\text{integer}) \\ 0 & \text{otherwise.} \end{cases}$$

Constructive interference if *y* is a multiple of *N/r*, destructive interference otherwise.

We sample uniformly from values of y such that

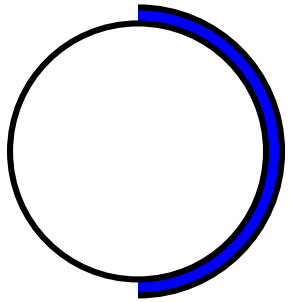
$$\frac{y}{N} = \frac{k}{r}, \quad k \in \mathbb{Z}$$

And even if *N* is not a multiple of *r*, we sample such a value of *y* with O(1) success probability.

There are r values of y such that $\left|\frac{y}{N} - \frac{k}{r}\right| \le \frac{1}{2N}, \quad k \in \mathbb{Z}.$

If $\frac{y}{N} = \frac{k}{r} + \delta, \quad |\delta| \le \frac{1}{2N}$, then $\exp\left(2\pi i\left(\frac{k}{r} + \delta\right) r j\right) = e^{2\pi i \delta r j}$.

$rj < N, \quad |\delta| \le 1/(2N) \Rightarrow$ phase is on right side of unit circle.



Therefore, for these values of *y,* constructive interference occurs.

Period Finding

Sum the geometric series: $\sum_{j=0}^{A-1} e^{2\pi ijry/N} = \sum_{j=0}^{A-1} \left(\omega^{ry}\right)^j = \frac{\left(\omega^{ry}\right)^A - 1}{\omega^{ry} - 1}, \quad \omega = e^{2\pi i/N}$

$$\Rightarrow \text{Prob}(y) = \frac{A}{N}\left|\frac{1}{A}\sum_{j=0}^{A-1} e^{2\pi ijry/N}\right|^2 = \frac{1}{NA}\left|\frac{e^{2\pi iAry/N}-1}{e^{2\pi iry/N}-1}\right|^2 = \frac{1}{NA}\frac{\sin^2\left(\pi Ary/N\right)}{\sin^2\left(\pi ry/N\right)} = \frac{1}{NA}\frac{\sin^2\left(\pi Ar\delta\right)}{\sin^2\left(\pi r\delta\right)}.$$

Here $|\delta| \le \frac{1}{2N}, \quad Ar \le N$. Note that for $|Cx| \le \frac{\pi}{2}$, we have $C\left|\frac{\sin Cx}{Cx}\right| \ge \frac{2}{\pi}C$

$$\Rightarrow \quad \text{Prob}(y) = \frac{1}{NA}\frac{\sin^2\left(\pi Ar\delta\right)}{\sin^2\left(\pi r\delta\right)} \ge \frac{1}{NA}\left(\frac{2A}{\pi}\right)^2 = \left(\frac{4}{\pi^2}\right)\frac{A}{N}.$$

There are r such y with $\left|\frac{y}{N} - \frac{k}{r}\right| \le \frac{1}{2N}$. Hence we sample such y with total prob $\ge 4/\pi^2 = .405$

The probability distribution is not precisely uniform in *k*, since the value of δ depends on *k*, but it is not highly biased against any value of *k*.

Recall we have been promised that $r \le M$. If we chose *N* large enough, we'll be able to infer the value of *k/r*. It suffices to choose $N \ge M^2$.

Period Finding

There are r such y with $\left|\frac{y}{N}-\frac{k}{r}\right|<\frac{1}{2N}$. Hence we sample such y with total prob $\geq 4/\pi^2 = .405$

Recall we have been promised that $r \leq M$. If we chose N large enough, we'll be able to infer the value of k/r. It suffices to choose $N \geq M^2$. The closest spacing between rational numbers with denominator $\leq M$ is $1/M^2$. Furthermore, there is an efficient classical continued-fraction algorithm for finding the closest rational number to y/N with denominator less than $\sqrt{N}$.

$$\frac{a}{s}-\frac{b}{t}=\frac{at-bs}{st}\geq\frac{1}{M^2}, \quad \text{if} \quad s,t\leq\frac{1}{M}.$$

From k/r, either we know the value of r (if k and r have no common factor), or we know a factor of r (if k and r have a common factor). In the former case we are done --- we can easily verify that r really is the period. In the latter case, we repeat the procedure, finding a new value k'/r.

If we find rational numbers (reduced to lowest terms) $k/r = k_1/r_1$, $k'/r = k_2/r_2$, $k''/r = k_3/r_3$, …, then if GCD(k , k' , k'', …) = 1 (no common factor), we have LCM(r_1 , r_2 , r_3, …) = r.
Thus we solve Period Finding if the sampled values of k have no common factor, which is true with high probability after a constant number of trials. Hence the quantum query complexity of Period Finding is constant (even better than for Simon's problem).

Quantum Fourier Transform.

$$\mathrm{QFT}_N : |x\rangle \to \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y / N} \, |y\rangle.$$

We can do this unitary transformation on a quantum computer with circuit size polylog *N*. It is convenient to choose *N* to be a power of 2, $N = 2^n$.

Using binary notation:

$$x = x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \ldots + x_1 \cdot 2 + x_0, \quad y = y_{n-1} \cdot 2^{n-1} + y_{n-2} \cdot 2^{n-2} + \ldots + y_1 \cdot 2 + y_0.$$

$$\Rightarrow \quad \frac{xy}{2^n} \equiv y_{n-1}(.x_0) + y_{n-2}(.x_1 x_0) + y_{n-3}(.x_2 x_1 x_0) + \ldots + y_1(.x_{n-2} x_{n-3} \ldots x_0) + y_0(.x_{n-1} x_{n-2} \ldots x_0)$$ (dropping integer part).

Here, e.g., $.x_2 x_1 x_0 = \frac{x_2}{2} + \frac{x_1}{2^2} + \frac{x_0}{2^3}$.

Exponential factorizes: $e^{2\pi i x y / N} = e^{2\pi i y_{n-1}(.x_0)} e^{2\pi i y_{n-2}(.x_1 x_0)} \quad \ldots \quad e^{2\pi i y_0 (.x_{n-1} x_{n-2} \ldots x_0)}$

Computational basis state maps to product state.

$$\mathrm{QFT}_N : |x\rangle \to \frac{1}{\sqrt{N}} \sum_y e^{2\pi i x y / N} \, |y\rangle = \frac{1}{\sqrt{2^n}} \underbrace{\left(|0\rangle + e^{2\pi i (.x_0)} |1\rangle\right)}_{y_{n-1}} \otimes \underbrace{\left(|0\rangle + e^{2\pi i (.x_1 x_0)} |1\rangle\right)}_{y_{n-2}} \otimes \quad \ldots \quad \otimes \underbrace{\left(|0\rangle + e^{2\pi i (.x_{n-1} x_{n-2} \ldots x_0)} |1\rangle\right)}_{y_0}.$$

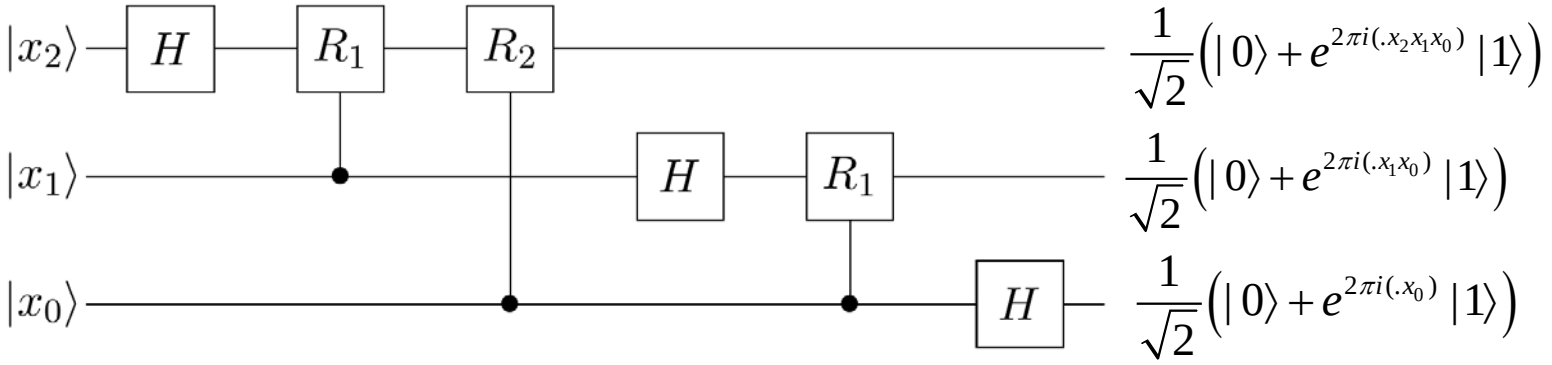
We can construct a simple circuit that performs this transformation.

Quantum Fourier Transform.

$$\mathrm{QFT}_N :| x\rangle \to \frac{1}{\sqrt{N}} \sum_y e^{2\pi i x y / N} \,| y\rangle = \frac{1}{\sqrt{2^n}} \Big(| 0\rangle + e^{2\pi i (.x_0)} \,| 1\rangle \Big) \otimes \Big(| 0\rangle + e^{2\pi i (.x_1 x_0)} \,| 1\rangle \Big) \otimes \quad \ldots \quad \otimes \Big(| 0\rangle + e^{2\pi i (.x_{n-1} x_{n-2} \ldots x_0)} \,| 1\rangle \Big).$$

y_{n-1} y_{n-2} y_0

For n = 3:



Order of bits is reversed.

Action of Hadamard: $H :| x_k\rangle \to \frac{1}{\sqrt{2}} \Big(| 0\rangle + e^{2\pi i (.x_k)} \,| 1\rangle \Big).$

Phase rotation: $R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^d} \end{pmatrix}.$

For n qubits, we use n H gates, and $n(n\text{-}1)/2$ controlled-R_d gates, $n(n\text{+}1)/2$ gates in all.

If we measure immediately after the QFT, no two-qubit gates are needed at all, just single-qubit gates conditioned on measurement outcomes. The controlled rotations are symmetric between control and target qubits.

After H gate, measure y_0 register. Then apply $(R_1)^{y_0}$ to y_1 register, $(R_2)^{y_0}$ to y_2 register, etc.

Then apply H gate, to y_1 register, measure y_1 register, apply $(R_1)^{y_1}$ to y_2 register, $(R_2)^{y_1}$ to y_3 register, etc.