CS294-4: Fourier Transforms & Theoretical Computer Science **Spring 1999**

Lecture 6: April 9, 1999

Lecturer: Umesh Vazirani *Scribe: Rupak Majumdar*

In the last lecture we saw that the state of a single qubit is a unit vector in $\mathbb{C}^2$. The state of n qubits is given by a unit vector in the space $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \ldots \mathbb{C}^2 \equiv \mathbb{C}^{2^n}$. For example, a system with thre qubits may be in a state

$$|\phi\rangle = \tfrac{1}{2}|000\rangle + \tfrac{i}{\sqrt{2}}|110\rangle - \tfrac{1}{2}|011\rangle.$$

A partial measurement of the first bit will then produce the bit 0 with probability $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, and the system goes to the new state $\frac{1}{\sqrt{2}}|000\rangle - \frac{1}{\sqrt{2}}|011\rangle$. Similarly,the bit 1 is seen with probability $\frac{1}{2}$ and the new state is $|110\rangle$.

Another important concept is the evolution of a quantum state over time. Given an initial state $|\phi_i\rangle$, the state an be transformed by a unitary transformation $\mathcal{U}$ to produce the final state $|\phi_f\rangle = \mathcal{U}|\phi_i\rangle$. Further, we require that $\mathcal{U}$ be *efficiently computable* in the sense that $\mathcal{U} = \mathcal{U}_1\mathcal{U}_2 \ldots \mathcal{U}_T$, where each $\mathcal{U}_i$ is *elementary*, and T is a polynomial. Recall that a unitary transformation is elementary if it is the identity on all but a small constant (say 2) number of bits. We give two elementary transformations that can be computed efficiently. These will be used as primitive operations in the quantum algorithms that we shall see.

The first transformation is a *permutation*. This simply causes a change in the labelings of the basis vectors.

$$\textstyle\sum_x \alpha_x|x\rangle|0\rangle \longrightarrow \sum_x \alpha_x|x\rangle|f(x)\rangle$$

where f is polynomial time computable (classically).

The second transformation is the *Fourier transform*, for example a Fourier transform over the group $\mathbb{Z}_2^n$. For a single qubit, the Fourier transform acts as follows:

$$|0\rangle \longrightarrow \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \longrightarrow \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The matrix for the tansformation (ignoring normalization constants) is

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Now since $\mathbb{Z}_2^n = \mathbb{Z}_2 \otimes \ldots \otimes \mathbb{Z}_2 = (\mathbb{Z}_2)^{\otimes n}$, we can write the Fourier transform for n qubits as $H_1 \otimes \ldots \otimes H_1$ (n times). The matrix written explicitly is

$$H_n = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}$$

6.1 Use of Elementary Transforms in Computation

We shall now apply the ideas discussed above to a problem due to Simon[1]. We shall solve an oracle problem that has a simple quantum mechanical algorithm that runs in polynomial time on a quantum computer, but takes exponential time on a classical computer.

We use the finite field $\mathbb{Z}_2^n$. For $x, y \in \mathbb{Z}_2^n$, let $x + y$ denote the bitwise addition and $x \cdot y$ the inner product ($\sum x_i y_i \mod 2$). Simon's problem is the following: Given an oracle for a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, and a promise that either

1. f is one to one; or
2. f is two to one, and there exists an $s \in \mathbb{Z}_2^n$ such that $f(x) = f(x + s)$ for all $x \in \mathbb{Z}_2^n$.

Simon's problem is to determine which of (1) or (2) holds for f and, in the latter case, to find s. Intuitively, this problem is hard to solve classically. Given the function f a (classical) probabilistic algorithm will look for collisions (by guessing x and $x+s$), but the probability of finding a collision is exponentially small. More formally, we can prove:

Theorem 6.1 *Any probabilistic algorithm to solve Simon's problem requires exponential time.*

Proof: We shall apply Yao's lemma to provide a ower bound on the time required to find s in case (2) holds.

Lemma 6.2 Yao's lemma. *Suppose there is a probability distribution $\mathcal{D}$ on all possible inputs such that no deterministic algorithm running in time T can give a correct answer with probability at least p when the input is drawn from the distribution $\mathcal{D}$. Then there is no probabilistic algorithm rnning in time T with a probability of success at least p.*

The hard probability distribution is the following. We choose s uniformly at random from all nonzero elements of $\mathbb{Z}_2^n$. This divides the elements in pairs $(x, x + s)$. Each pair is mapped to a random element of $\mathbb{Z}_2^n$ such that no two pairs are mapped to the same element.

Consider the action of the best deterministic algoritm on this distribution. After m steps, the algorithm has computed at most m values of the function f. Let these values be $f(x_1), f(x_2), \ldots f(x_m)$. If $f(x_i) = f(x_j)$ for some i, j such that $x_i \neq x_j$, then we know $s = x_i + x_j$. On the other hand, if $f(x_i) \neq f(x_j)$ then it follows that $s \neq x_i + x_j$.

If $f(x_1), f(x_2), \ldots, f(x_k)$ are all distinct, then we can say that s is none of the $\binom{k}{2}$ values $x_i + x_j$. Also, all other values of s are equally likely. Now suppose the detrministic alorithm chooses x_{k+1}. The probability that for some $i \in \{1, \ldots, k\}$ $f(x_{k+1}) = f(x_i)$ (i.e. $s = x_i + x_{k+1}$) is bounded above by

$$\frac{k}{2^n - 1 - \binom{k}{2}}$$

Takin the sum over all $k \in \{1, \ldots, m\}$, we get

$$\sum_{k=1}^{m} \frac{k}{2^n - 1 - \binom{k}{2}} \leq \sum_{k=1}^{m} \frac{k}{2^n - k^2} \leq \frac{m^2}{2^n - m^2}.$$

Now, if $m = 2^{(\frac{1}{2}-\epsilon)n}$ (i.e. there is a subexponential time deterministic algorithm), then

$$\frac{m^2}{2^n - m^2} = \frac{2^{(1-2\epsilon)n}}{2^n - o(2^n)} = 2^{-2\epsilon n} - o(2^{-2\epsilon n}).$$

Thus, the probability of success of the best deterministic algorithm is exponentially small. By Yao's lemma, we get a corresponding lower bound for probabilistic algorithms. ■

Although there is no subexponential time classical algorithm for this problem, we now show a polynomial time quantum algorithm that solves this problem.

We initialize the system to the distibution $\frac{1}{2^{\frac{n}{2}}} \sum_{x \in \{0,1\}^n} |x\rangle|0\rangle$, i.e. there is the same amplitude to be in each of the 2^n states. Notice that we can reach this state from a state $|0\rangle|0\rangle$ by doing a Fourier transform on the first register.

Now we compute f and get the state $\frac{1}{2^{\frac{n}{2}}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle$. Bits of $f(x)$ will not be used beyond this point. Hence, we can apply the principle of safe storage:
The Principle of Safe Storage. If some bits of a quantum circuit are not changed after some moment, then the outcome of the circuit is the same as in the case if these bits are measured immediately.

Let $|f(z)\rangle$ be the result of measuring $|f(x)\rangle$. Consider the case where f is two to one. Since there are exactly two x's such that $f(x) = f(z)$, namely z and $z + s$, the quantum state after measuring $|f(x)\rangle$ is the superposition of exactly two states,

$$\frac{1}{\sqrt{2}}(|z\rangle|f(z)\rangle + |z+s\rangle|f(z)\rangle).$$

If we measure $|x\rangle$ at this point, we get z or $z + s$ with equal probability, but not both. This will not serve our purpose (we get a random $x \in \mathbb{Z}_2^n$ and the value $f(x)$). We do a Fourier transform on this state (on the first register) to get

$$\frac{1}{\sqrt{2}} \frac{1}{2^{\frac{n}{2}}} \sum_{u \in \{0,1\}^n} [(-1)^{z.u} + (-1)^{(z+s).u}]|u\rangle = \frac{1}{2^{\frac{n+1}{2}}} \sum_u (-1)^{x.u}[1 + (-1)^{s.u}]|u\rangle$$

Hence, if $s.u = 1$, then the amplitude of $|u\rangle$ is zero. On the other hand, all $|u\rangle$ such that $s.u = 0$ have the same amplitude $\frac{(-1)^{x.u}}{2^{\frac{n+1}{2}}}$(with different signs). Thus, on measurement, we get a random u such that $s.u = 0$.

This u gives us one bit of information about s: we can conclude that s is such that $\sum s_i u_i = 0 (\text{mod} 2)$. Simon's algorithm runs the above steps $(n-1)$ times and obtains $n-1$ vectors. If these

vectors are linearly independent, we get a system of linear equations that can be solved to obtain s. After that, we check that $f(x) = f(x+s)$ for some x. If yes, we are done. If not, f is one to one. If some vectors are linearly dependent, we run the algorithm again, until $n-1$ linearly independent vectors are obtained.

Notice that if f is one to one, we again get u uniformly distributed over $\mathbb{Z}_2^n$. In that case, the $n-1$ vectors will again be linearly independent with high probability. The system of linear equations in this case will have the solutions 0 and some s'. However, there is no x such that $f(x) = f(x+s')$. Hence, the check at the end of the process is essential.

It remains to prove that with high probability, $n-1$ vectors such that $u.s = 0$ are linearly independent. Let the vectors be $u_1, u_2, \ldots, u_{n-1}$. The number of linear combinations of $u-1, \ldots, u_{i-1}$ is at most 2^{i-1}. Therefore, the probability that u_i is linearly independent from $u_1, \ldots, u_{i-1}$ is at least

$$\frac{2^{n-1}-2^{i-1}}{2^{n-1}} = 1 - \frac{1}{2^{n-i}}.$$

The probability that $u_1, u_2, \ldots, u_{n-1}$ are all linearly independent is the product of the above probabilities:

$$(1 - \tfrac{1}{2^{n-1}})(1 - \tfrac{1}{2^{n-2}})(1 - \tfrac{1}{2}).$$

Also, by the sum bound, the probability that $u_1, \ldots, u_{n-2}$ are linearly dependent is at most $\frac{1}{2^{n-1}} + \frac{1}{2^{n-2}} + \ldots + \frac{1}{4} = \frac{1}{2}$. Thus, $u_1, \ldots, u_{i-2}$ are linearly independent with probability at least half, and u_{i-1} is linearly independent of all of them with probability at least half. So, the probability that all vectors are independent is at least $\frac{1}{4}$.

Note that we did not use the range of f in the proof, so f can be a function from $\mathbb{Z}_2^n$ to any range.

6.2 Hidden Subgroup Problem

Now we generalize Simon's problem to any Abelian group. Let G be an Abelian group and $f : G \to G$ be a function that is constant on the cosets of some (hidden) subgroup H of G, and distinct on diferent cosets. The *hidden subgroup problem* asks to find H, given f as an oracle.

Our goal is to prove the following:

Theorem 6.3 *If we can do a Fourier transform over G, then there is a quantum mechanical algorithm to solve the Hidden Subgroup problem.*

By a result of Kitaev, we can compute Fourier transforms over arbitrary Abelian groups, and so we have a polynomial time quantum algorithm for the hidden subgroup problem. The algorithm uses some properties of the Fourier transform. We first state the properties.

Property 6.4 *The image of an equal superposition over a subgroup is an equal superposition over the quotient group.*

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle \xrightarrow{F.T.} \sqrt{\frac{|H|}{|G|}} \sum_{k \in [G:H]} |k\rangle$$

Actually this is a slight mis-statement of the result, since the elements of $[G : H]$ are cosets of H. We shall correct the result at the end of the proof.

Property 6.5 *The Fourier transform treats all cosets similarly. The image of uniform superpositions on the coset $g'H$ for all cosets of H have components differing only by a phase factor. The magnitudes are independent of the specific coset.*

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle \xrightarrow{F.T.} \sqrt{\frac{|H|}{|G|}} \sum_{k \in [G:H]} e^{i\theta_k} |k\rangle$$

Proof: The proof follows from the convolution-multiplication property of Fourier transforms. Let $A(x)$ be a function that is constant over the subgroup H.

$$A(x) = \begin{cases} \frac{1}{\sqrt{|H|}} & \text{if } x \in H \\ 0 & \text{otherwise} \end{cases}$$

Let

$$\delta_g(x) = \begin{cases} 1 & \text{if } x = g \\ 0 & \text{otherwise} \end{cases}$$

We want to construct $B(x) = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle$

$$B(x) = \begin{cases} \frac{1}{\sqrt{|H|}} & \text{if } x \in Hg \\ 0 & \text{otherwise} \end{cases}$$

Claim. $B(x) = A(x) * \delta_g(x)$.

Assuming the claim, $\hat{B}(x) = \sum \hat{A}(x)\hat{\delta}_g(x)$. Note that the $\hat{\delta}_g(x)$ term gives rise to the phases.

Proof of Claim. The right hand side can be written as $\sum_y A(xy^{-1})\delta_g(y) = A(xg^{-1})$. Now, $B(x) \neq 0$ only if $xg^{-1} \in H$, i.e. $x \in Hg$. ■

Using the properties above, the figure shows an algorithm to solve the Hidden subgroup problem. Notice that this is identical to the algorithm for the particular case $G = \mathbb{Z}_2^n$ with the Hadamard transforms replaced by general Fourier transforms.

We assume each register is big enough to hold a group element. We start in the state $|0\rangle|0\rangle$, where 0 is the group identity. We do a Fourier transform over G on the first register to get a superposition

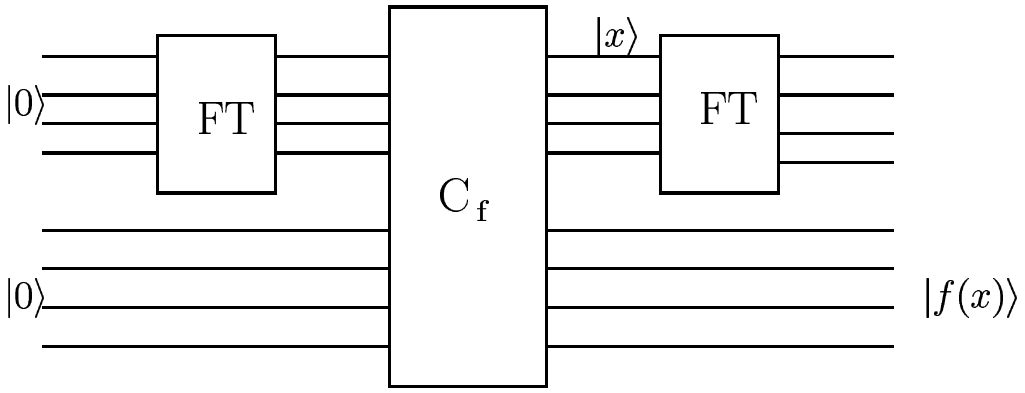

Figure 6.1: Circuit for the Hidden Subgroup Problem

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |0\rangle.$$

Now we compute f in the second register to get $\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle$. We measure the second register to get the value of f on some random coset. At this point, the first register is an equal superposition over this coset. As in the first case, if we measure the value of the first register at this point, we shall only get a random element in the coset. This does not serve our purposes. So we do the same trick as in section 6.1 and take the Fourier transform on the first register and measure.

By property 2 above, we know that the Fourier transform gives magnitudes that are equal for all elements, and it does not matter which particular coset we came from. The measurement produces a random element of the quotient group $[G : H]$. By a result of Babai *et al*, we can reconstruct a group, given the ability to randomly sample from it. Thus we can reconstruct $[G : H]$ and hence, H.

Now we fix property 6.4. Assume $G = \mathbb{Z}_p^n$ for a prime p. An element of G is a n-tuple $(a_1, a_2, \ldots, a_n)$. We shall write such a tuple as $\vec{a}$. The image of $|\vec{a}\rangle$ under the Fourier transform is

$$\sum_{\vec{b}} \omega_p^{\sum a_i b_i \pmod p} |\vec{b}\rangle = \sum_{\vec{b}} \omega_p^{\vec{a}.\vec{b}} |\vec{b}\rangle.$$

What happens when we start in a superposition over a subgroup? If we ignore normalization factors,

$$\begin{aligned} \sum_{\vec{a} \in H} |\vec{a}\rangle \quad &\longrightarrow \\ & \sum_{\vec{a}} \sum_{\vec{b}} \omega_p^{\vec{a}.\vec{b}} |\vec{b}\rangle \\ &= \sum_{\vec{b}} \sum_{\vec{a}} \omega_p^{\vec{a}.\vec{b}} |\vec{b}\rangle \end{aligned}$$

There are two cases. Either $\vec{a}.\vec{b} = 0$ for all $\vec{a} \in H$. In this case we have constructive interference, and for all $\vec{b} \in H^{\perp}$ we get a uniform superposition over $|\vec{b}\rangle$.

On the other hand, if there exists an $\vec{a}$ such that $\vec{a}.\vec{b} \neq 0 (\text{mod } p)$, we can partition the $\vec{a}$'s according to the values of $\vec{a}.\vec{b}$. The number of elements in each partition is the same. Thus, we get the amplitude $\sum_{\vec{a}} \omega_p^{\vec{a}.\vec{b}} = 0$.

So the "correct" statement of property 6.4 is:

Property 6.4′. $\frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle \xrightarrow{F.T.} \sqrt{\frac{|H|}{|G|}} \sum_{k \in H^{\perp}} |k\rangle$.

In general, for any Abelian group $G = \mathbb{Z}_{p_1{}^{e_1}} \times \mathbb{Z}_{p_2{}^{e_2}} \times \ldots \times \mathbb{Z}_{p_n{}^{e_n}}$, define $\vec{a}.\vec{b} = \sum_i a_i b_i Q_i$, where $Q_i = \prod_{j \neq i} p_j^{e_j}$. Let $N = \prod_{j=1}^{n} p_j^{e_j}$. Then

$$|\vec{a}\rangle \longrightarrow \sum_{\vec{b}} \omega_N^{\vec{a}.\vec{b}} |\vec{b}\rangle$$

We can check that this is the correct definition of the Fourier transform by showing that the unit vectors transform correctly. Consider the unit vector $(0, \ldots, 0, a_i, 0, \ldots, 0)$. The transform of this vector can be viewed in two different ways: according to the definition above, the Fourier transform of this vector is

$$\sum_{\vec{b}} \omega_N^{a_i b_i Q_i} |\vec{b}\rangle.$$

On the other hand, in the group $\mathbb{Z}_N$, the Fourier transform is

$$|a_i Q_i (Q_i^{-1} \bmod p_i^{e_i})\rangle \longrightarrow \sum_{b (\text{mod } N)} \omega_N^{a_i Q_i (Q_i^{-1} (\text{mod } p_i^{e_i})) b} |b\rangle.$$

These two can easily be seen to be equal.

Also, property 6.4 and 6.5 follow. To show property 6.4, let

$$\sum_{\vec{a} \in H} |\vec{a}\rangle \longrightarrow \sum_{\vec{b}} \sum_{\vec{a}} \omega_N^{\vec{a}.\vec{b}} |\vec{b}\rangle$$

If for all $\vec{a}$, $\vec{a}.\vec{b} = 0$, then we are done (as before). Else, we want the entire amplitude of $|\vec{b}\rangle$ to be zero. Let $S = \sum_{\vec{a}} \omega_N^{\vec{a}.\vec{b}}$. Since $H = H + \vec{a}$, we get $S = \omega_N^{\vec{a}.\vec{b}} S$. Since $\omega_N^{\vec{a}.\vec{b}} \neq 1$, this immediately implies $S = 0$.

References

[1] D. Simon, On the power of quantum computation, *Proc. 35th Annual Symposium on Fundamentals of Computer Science*, 1994, pp. 116-123.