1 Fourier transforms over finite abelian groups

Let $G$ be a finite abelian group. The characters of $G$ are homomorphisms $\chi_j : G \to \mathbb{C}$. There are exactly $|G|$ characters, and they form a group, called the dual group, and denoted by $\hat{G}$. The Fourier transform over the group $G$ is given by:

$$|g\rangle \mapsto \frac{1}{\sqrt{|G|}} \sum_j \chi_j(g) |j\rangle$$

Consider, for example $G = \mathbb{Z}_N$. The characters are defined by $\chi_j(1) = \omega^j$ and $\chi_j(k) = \omega^{jk}$. And the Fourier transform is given by the familiar matrix $F$, with $F_{jk} = \frac{1}{\sqrt{N}} \omega^{jk}$.

In general, let $G \cong \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \times \cdots \times \mathbb{Z}_{N_l}$, so that any $g \in G$ can be written equivalently as $(a_1, a_2, \ldots, a_l)$, where $a_j \in \mathbb{Z}_{N_j}$. Now, for each choice of $k_1, \ldots, k_l$ we have a character given by the mapping:

$$\chi_{k_1,\ldots,k_l}(a_1, a_2, \ldots, a_l) = \omega_{N_1}^{k_1 a_1} \cdot \omega_{N_2}^{k_2 a_2} \cdots \omega_{N_l}^{k_l a_l}$$

Finally, the Fourier transform of $(a_1, a_2, \ldots, a_l)$ can be defined as

$$(a_1, a_2, \ldots, a_l) \mapsto \frac{1}{\sqrt{|G|}} \sum_{(k_1,\ldots,k_l)} \omega_{N_1}^{k_1 a_1} \omega_{N_2}^{k_2 a_2} \cdots \omega_{N_l}^{k_l a_l} |k_1 \cdots k_l\rangle$$

2 Subgroups and Cosets

Corresponding to each subgroup $H \subseteq G$, there is a subgroup $H^\perp \subseteq \hat{G}$, defined as $H^\perp = \{ k \in \hat{G} \mid k(h) = 1 \ \forall h \in H \}$, where $\hat{G}$ is the dual group of $G$. $|H^\perp| = \frac{|G|}{|H|}$. The Fourier transform over $G$ maps an equal superposition on $H$ to an equal superposition over $H^\perp$.

Claim

$$\frac{1}{\sqrt{|H|}} \sum_h |h\rangle \xrightarrow{FT_G} \sqrt{\frac{|H|}{|G|}} \sum_{k \in H^\perp} |k\rangle$$

Proof The amplitude of each element $k \in H^\perp$ is $\frac{1}{\sqrt{|G|}} \sum_{h \in H} k(h) = \frac{\sqrt{|H|}}{\sqrt{|G|}}$. But since $|H^\perp| = \frac{|G|}{|H|}$, the sum of squares of these amplitudes is 1, and therefore the amplitudes of elements not in $H^\perp$ is 0. The Fourier transform over $G$ treats equal superpositions over cosets of $H$ almost as well:
Claim
\[
\frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle \xrightarrow{FT} \frac{1}{\sqrt{|G|}} \sum_{k \in H} \chi_h(k) |k\rangle
\]

Proof This follows from the convolution-multiplication property of Fourier transforms. An equal superposition on the coset \(Hg\) can be obtained by convolving the equal superposition over the subgroup \(H\) with a delta function at \(g\). So after a Fourier transform, we get the pointwise multiplication of the two Fourier transforms: namely, an equal superposition over \(H^\perp\), and \(\chi_h\).

Since the phase \(\chi_h(k)\) has no effect on the probability of measuring \(|k\rangle\), Fourier sampling on an equal superposition on a coset of \(H\) will yield a uniformly random element \(k \in H^\perp\). This is a fundamental primitive in the quantum algorithm for the hidden subgroup problem.

Claim Fourier sampling performed on \(|\Phi\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle\) gives a uniformly random element \(k \in H^\perp\).

3 The hidden subgroup problem

Let \(G\) again be a finite abelian group, and \(H \subseteq G\) be a subgroup of \(G\). Given a function \(f : G \rightarrow S\) which is constant on cosets of \(H\) and distinct on distinct cosets (i.e., \(f(g) = f(g')\) iff there is an \(h \in H\) such that \(g = hg'\)), the challenge is to find \(H\).

The quantum algorithm to solve this problem is a distillation of the algorithms of Simon and Shor. It works in two stages:

Stage I Setting up a random coset state:

Start with two quantum registers, each large enough to store an element of the group \(G\). Initialize each of the two registers to \(|0\rangle\). Now compute the Fourier transform of the first register, and then store in the second register the result of applying \(f\) to the first register. Finally, measure the contents of the second register. The state of the first register is now a uniform superposition over a random coset of the hidden subgroup \(H\):

\[
|0\rangle |0\rangle \xrightarrow{FT} \frac{1}{\sqrt{|G|}} \sum_{a \in G} |a\rangle |0\rangle \xrightarrow{f} \frac{1}{\sqrt{|G|}} \sum_{a \in G} |a\rangle |f(a)\rangle \quad \text{measure 2nd reg} \quad \frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle
\]

Stage II Fourier sampling:

Compute the Fourier transform of the first register and measure. By the last claim of the previous section, this results in a random element of \(H^\perp\), i.e., random \(k : k(h) = 0 \forall h \in H\). By repeating this process, we can get a number of such random constraints on \(H\), which can then be solved to obtain \(H\).

Example Simon’s Algorithm: In this case \(G = \mathbb{Z}_2^n\) and \(H = \{0, s\}\). Stage I sets up a random coset state \(1/\sqrt{2} |x\rangle + 1/\sqrt{2} |x+s\rangle\). Fourier sampling in stage II gives a random \(k \in \mathbb{Z}_2^n\) such that \(k \cdot s = 0\). Repeating this \(n - 1\) times gives \(n - 1\) random linear constraints on \(s\). With probability at least \(1/e\) these linear constraints have full rank, and therefore \(s\) is the unique non-zero solution to these simultaneous linear constraints.

4 Factoring and discrete log

Recall that factoring is closely related to the problem of order finding. To define this problem, recall that:
The set of integers that are relatively prime to \( N \) form a group under the operation of multiplication modulo \( N \): \( \mathbb{Z}_N^* = \{ x \in \mathbb{Z}_N : \gcd(x,N) = 1 \} \).

Let \( x \in \mathbb{Z}_N^* \). The order of \( x \) (denoted by \( \text{ord}_N(x) \)) is \( \min_{r \geq 1} x^r \equiv 1 \mod N \).

The task of factoring \( N \) can be reduced to the task of computing the order of a given \( x \in \mathbb{Z}_N^* \). Recall that \( |\mathbb{Z}_N^*| = \Phi(N) \), where \( \Phi(N) \) is the Euler Phi function. If \( N = p_1^{e_1} \cdots p_k^{e_k} \) then \( \Phi(N) = (p_1 - 1)p_1^{e_1 - 1} \cdots (p_k - 1)p_k^{e_k - 1} \). Clearly, \( \text{ord}_N(x)|\Phi(N) \).

Consider the function \( f : \mathbb{Z}_{\Phi(N)} \rightarrow \mathbb{Z}_N^* \), where \( f(a) = x^a \mod N \). Then \( f(a) = 1 \) if \( a \in \langle r \rangle \), where \( r = \text{ord}_N(x) \), and \( \langle r \rangle \) denotes the subgroup of \( \mathbb{Z}_N^* \) generated by \( r \). Similarly if \( a \in \langle r \rangle + k \), a coset of \( \langle r \rangle \), then \( f(a) = x^k \mod N \). Thus \( f \) is constant on cosets of \( H = \langle r \rangle \).

The quantum algorithm for finding the order \( r \) or \( x \) first uses \( f \) to set up a random coset state, and then does Fourier sampling to obtain a random element from \( H^\perp \). Notice that the random element will have the form

\[
 k = s \cdot \frac{\phi(N)}{r}
\]

where \( s \) is picked randomly from \( \{0, \ldots, r - 1\} \). If \( \gcd(s, r) = 1 \) (which holds for random \( s \) with reasonably high probability), \( \gcd(k, \phi(N)) = \phi(N)/r \). From this it is easy to recover \( r \). There is no problem discarding bad runs of the algorithm, since the correct value of \( r \) can be used to split \( N \) into non-trivial factors.

Here we assumed that we know \( \phi(N) \) or at least a multiple of it. However, given \( N \) computing \( \phi(N) \) is as hard as factoring \( N \). Shor’s factoring algorithm relies on the fact that the result of doing a fourier transform over \( \mathbb{Z}_N \) may be closely approximated by carrying out the fourier transform over \( \mathbb{Z}_M \) for \( M >> N \) and reinterpreting results.

**Discrete Log Problem:**

Computing discrete logarithms is another fundamental problem in modern cryptography. Its assumed hardness underlies the Diffie-Helman cryptosystem.

In the Discrete Log problem is the following: given a prime \( p \), a generator \( g \) of \( Z_p^* \) (\( Z_p^* \) is cyclic if \( p \) is a prime), and an element \( x \in Z_p^* \); find \( r \) such that \( g^r \equiv x \mod p \).

Define \( f : Z_{p-1} \times Z_{p-1} \rightarrow Z_p^* \) as follows: \( f(a,b) = g^a x^{-b} \mod p \).

Notice that \( f(a,b) = 1 \) exactly when \( a = br \). Equivalently, when \( (a,b) \in \langle (r,1) \rangle \), where \( \langle (r,1) \rangle \) denotes the subgroup of \( Z_{p-1} \times Z_{p-1} \) generated by \( (r,1) \).

Similarly, \( f(a,b) = g^k \) for \( (a,b) \in \langle (r,1) \rangle + (k,0) \). Therefore, \( f \) is constant on cosets of \( H = \langle (r,1) \rangle \).

Again the quantum algorithm first uses \( f \) to set up a random coset state, and then does Fourier sampling to obtain a random element from \( H^\perp \), i.e. \( (c,d) \) such that \( rc + d = 0 \mod p - 1 \). For a random such choice of \( (c,d) \), with reasonably high probability \( \gcd(c, p - 1) = 1 \), and therefore \( r = -dc^{-1} \mod p - 1 \). Once again, it is easy to check whether we have a good run, by simply computing \( g^r \mod p \) and checking to see whether it is equal to \( x \).