

# 1. Fourier Transform.

傅里叶变换，能将一个复杂的函数分解成不同频率的正弦波的叠加。

也即 时域 (Time Domain)  $\longrightarrow$  频域 (Frequency Domain)

## 1.1 definition:

连续傅里叶变换 CFT

对于一连续可积函数  $x(t)$ , 我们定义:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-2\pi f t i} dt,$$
$$\hookrightarrow = \cos(2\pi f t) - i \sin(2\pi f t)$$

对应的, 我们有:

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{2\pi f t i} df$$

离散傅里叶变换 DFT.

对长度为  $n$  的复序列:  $\{x_0, x_1, \dots, x_{n-1}\}$ , 我们定义:

$$X_k = \sum_{j=0}^{n-1} x_j \cdot e^{-\frac{2\pi k j i}{n}}, \quad k=0, \dots, n-1$$

也可以写成:  $\vec{X} = F \vec{x}$ , 其中  $F$  是一个  $n \times n$  的矩阵

而在 QC 中, 我们也有类似定义:

## 2. 量子傅里叶变换.

### Quantum Fourier Transform.

#### 2.1. Definition.

QFT 是 DFT 在 QC 中的推广. 我们考虑定义如下的 Fourier Basis,

对于 d 维量子系统的 Fourier Basis  $\{|e_k\rangle\}_{k=0}^{d-1}$ , 有:

$$|e_k\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} e^{\frac{2\pi i k j}{d}} |j\rangle$$

也可简化地记作:

$$|e_k\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} w^{kj} |j\rangle . \quad w = e^{\frac{2\pi i}{d}}$$

$$QFT = \sum_{j=0}^{d-1} |e_j\rangle \langle j|$$

经数学变形, 我们有 (2^n 维)

$$|e_j\rangle = \bigotimes_{l=1}^n \left( \frac{|0\rangle + e^{2\pi i \cdot (0.j_{n-l+1} \dots j_n)}}{\sqrt{2}} |1\rangle \right)$$

proof: 展开即可, 注意到:  $e^{2\pi i} = 1$

#### 2.2. 实现 QFT

我们有受控相位:  $R_k = |0\rangle \langle 0| + e^{2\pi i \cdot 2^{-k}} |1\rangle \langle 1|$

$$\text{有: } R_k |+\rangle = \frac{1}{2} \left( |0\rangle + e^{2\pi i \cdot (0.0 \dots 0)} |1\rangle \right)$$

因此不难想到:

$$|e_j\rangle = \bigotimes_{l=1}^n \frac{1}{\sqrt{2}} \left( |0\rangle + e^{2\pi i \cdot (0.j_{n-l+1} \dots j_n)} |1\rangle \right)$$

$$= \bigotimes_{l=1}^n \frac{1}{\sqrt{2}} \left( R_l^{j_n} \dots R_1^{j_{n-l+1}} |+\rangle \right)$$

$$= \bigotimes_{l=1}^n \frac{1}{\sqrt{2}} \left( R_l^{j_n} \cdots R_1^{j_{n-l+1}} (R_l^{j_l} H |j_l\rangle) \right) \rightarrow \text{标注为 } |+\rangle$$

不难发现，我们不能提前改变要用到的信息！

我们考虑，在改变第  $l$  个 qubit 时，以第  $n-l+1$  个 qubit 作为输入

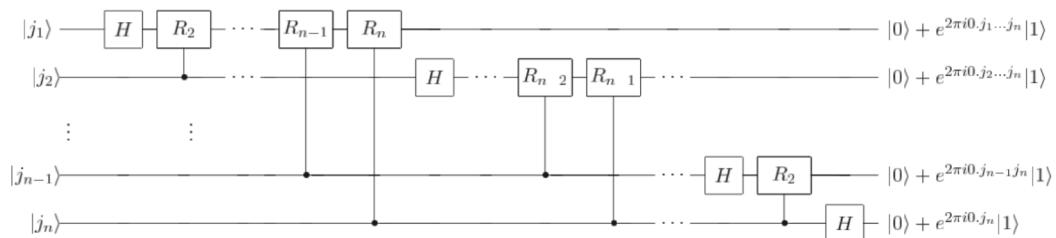
$$\text{Step 1. } |\psi_j\rangle' = \bigotimes_{l=1}^n \left( R_{n-l+1}^{j_n} \cdots R_2^{j_{l+1}} H |j_l\rangle \right)$$

也即  $|\psi_j\rangle'$  的第  $l$  个 qubit =  $|\psi_j\rangle$  中的第  $n-l+1$  个 qubit

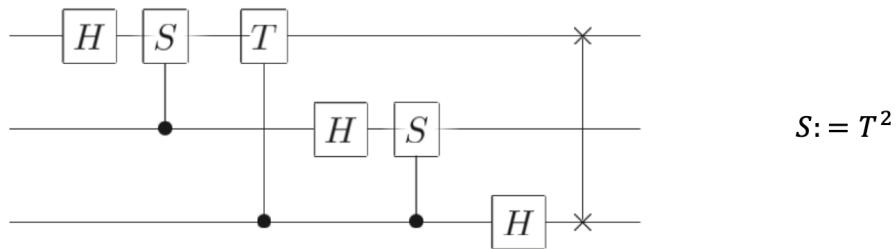
$$R_{n-l+1}^{j_n} \cdots R_2^{j_{l+1}} \cdot \underbrace{R_1^{j_l} R_{n-l+1}^{j_{n-l+1}}}_{\langle H | j_l \rangle} |j_{n-l+1}\rangle$$

## Circuit for QFT

- QFT circuit =



- QFT for 3 qubits:



## 2.3. QHT vs QFT

→ Hadamard

2.3.1 对于单qubit系统,  $QHT = QFT$

$$|e_0\rangle = |+\rangle = H|0\rangle$$

$$|e_1\rangle = |-\rangle = H|1\rangle$$

2.3.2 对于多qubit系统,  
 $QHT \rightarrow$  制备均匀的叠加态.

$$e_0 = \frac{1}{\sqrt{2^n}} \sum |i\rangle = H^{\otimes n} |0\rangle^{\otimes n}$$

## 2.4 QFT 的复杂度.

$$n\text{-qubits} \xrightarrow{QFT} \mathcal{O}(n^2)$$

$$\xrightarrow{QHT} \mathcal{O}(n)$$

对于传统DFT,  $n\text{-bits} \rightarrow \mathcal{O}(4^n)$

而快速DFT,  $n\text{-bits} \rightarrow \mathcal{O}(n^{2^n})$

### 3. QPE

## Quantum Phase Estimation

量子相位估计.

### 3.1 definition.

给定一个 quantum gate  $U$  及其本征态  $|u\rangle$ , s.t.

$$U|u\rangle = e^{2\pi i \varphi} |u\rangle$$

QPE 能高精度地估出相位  $\varphi \in [0, 1)$

$\varphi$  也即  $U$  相对于其本征态  $|u\rangle$  的本征值的相位  
(是本征值的相位)

### 3.2 线代数字背景.

任意酉算子 (unitary)  $U$  都可以被对角化为:

$$U = \sum_i e^{2\pi i s_i} |u_i\rangle \langle u_i|,$$

其中:

$\{|u_i\rangle\}$  是一组 DNB, 也是  $U$  的本征态集

$s_i \in [0, 1)$  是  $|u_i\rangle$  对应本征值的相位

### 3.3 Overview of QPE

Kitaev's QPE:

input: ① 一个 controlled-gate  $C(U)$

②  $U$  的本征态  $|u\rangle$ . s.t.  $U|u\rangle = e^{2\pi i \varphi} |u\rangle$

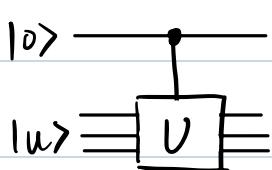
output: 对  $\varphi$  的估计  $\hat{\varphi}$ .

其中  $\hat{\varphi}$  为  $t - b_1 =$  进制数, s.t.  $|\hat{\varphi} - \varphi| < 2^{-t}$  且失败概率  $\leq \epsilon$

$$\text{使用 } t = n + \lceil \log_2(2 + \frac{1}{2\varepsilon}) \rceil \text{ 个 qubit}$$

我们考虑使用“phase kick back”.

注意到：



$$\begin{aligned} C(U)(|0\rangle \otimes |1u\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle \otimes |1u\rangle + |1\rangle \otimes (U|1u\rangle)) \\ &= \frac{1}{\sqrt{2}}|0\rangle \otimes u + \frac{1}{\sqrt{2}}|1\rangle \otimes e^{2\pi i \varphi}|1u\rangle \\ &= \frac{1}{\sqrt{2}}|0\rangle \otimes u + \frac{1}{\sqrt{2}}e^{2\pi i \varphi} \otimes |1u\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \varphi}|1\rangle) \otimes u \end{aligned}$$

类似QFT的形式。

PS: 我们只能测得不同 state 间的相位差。

此时，我们将在相位  $e^{2\pi i \varphi}$  从  $|1u\rangle$  “反冲 kickback” 回到 qubit  $|+\rangle$ ，

将隐藏在 U 中的相位信息传递到了可测量的控制 qubit 上。

### 3.4 QPE circuit .

对于  $-t$ -qubit 系统， $|0\rangle^{\otimes t}$ . 对所有 qubit 附加  $|+\rangle$ ，会得到 QFT 中的  $|e_o\rangle$

即：

$$|+\rangle^{\otimes t} = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle$$

我们将这个 circuit 转化为指数数量级的  $t$ -bit state 叠加！

我们考虑如下 Circuit

$$\sum_j \frac{e^{2\pi i \varphi_j}}{\sqrt{2^t}} |j\rangle$$

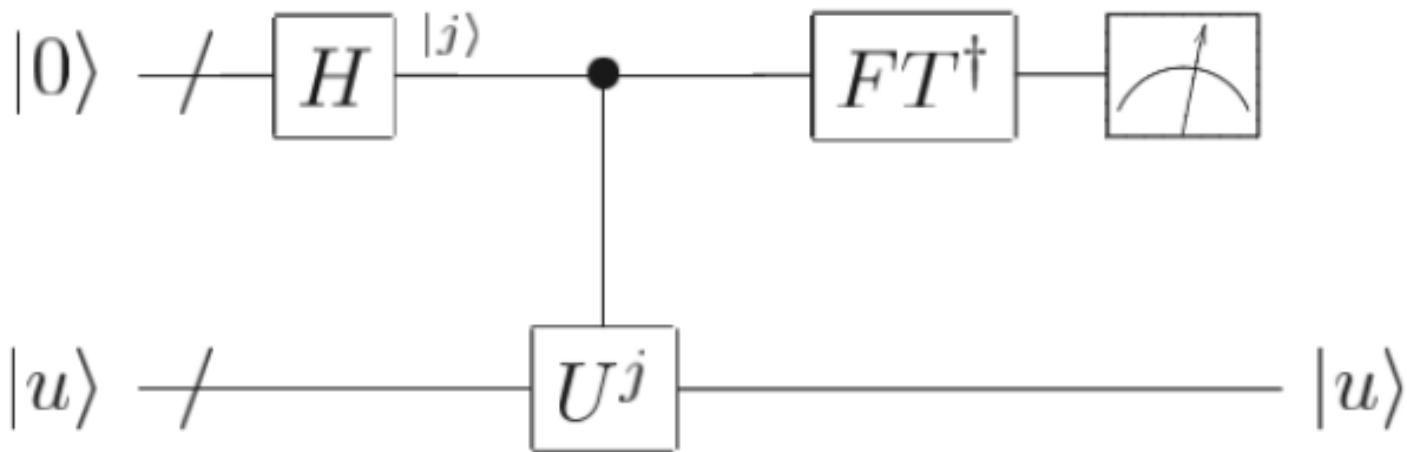
clock register  
时钟寄存器

(First Register)

Second register  
(the input register)

在此之后，我们对 Clock Register 应加逆QFT；即  $FT^\dagger$ 。用以计算基  
进行测量。

完整电路示意



3.5. Error Analyse.

$$p(j) = \left| \langle e_j | FT^\dagger | \sum_{k=0}^{t-1} \frac{e^{2\pi i \varphi_k}}{\sqrt{2^t}} |k\rangle \right|^2$$

$$\text{Let } \delta \varphi_j = \varphi - j \rightarrow (j_1 \dots j_t)_2$$

$$P(j) = \left| \sum_k \frac{e^{2\pi i \cdot k \cdot 8\varphi_j}}{2^t} \right| \quad (\text{几何级数求和})$$

$$= \frac{1}{2^{2t}} \left( \frac{\sin(\pi \cdot 2^t \cdot 8\varphi_j)}{\sin(\pi \cdot 8\varphi_j)} \right)^2$$

$$\leq \frac{1}{2^{2t}} \left( \frac{1}{\sin(\pi \cdot 8\varphi_j)} \right)^2$$

$$\leq \frac{1}{2^{2t}} \left( \frac{1}{8\varphi_j} \right)^2$$

i.e. 我们使用  $t$ -bit system 时，对误差为 3 的近似结果，  
正确的概率平均为  $\frac{1}{2^t} \cdot \frac{1}{3^2}$

## 4. Shor's Algorithm.

### 4.1. background.

我们先介绍一种加密方式：RSA公钥加密

RSA密钥利用了一个数论事实：对于两个大质数  $p, q$ ，计算  $p \cdot q$  容易，但因式分解  $N = p \cdot q$  却十分困难。

加密过程：

Step 1. 选择两个大质数  $p, q$  (现在通常取 2048 位)

Step 2. 计算  $N = p \cdot q$

Step 3. 计算 Euler's function:  $\phi(N) = (p-1)(q-1)$

我们选择一个公钥指数  $e$ ，满足：

$$\textcircled{1} \quad 1 < e < \phi(N)$$

$$\textcircled{2} \quad \gcd(e, \phi(N)) = 1$$

$$\textcircled{3} \quad \text{常用 } e = 2^b + 1 = 65537$$

此后，我们计算私钥指数  $d$ ，满足：

$$e \cdot d \equiv 1 \pmod{\phi(N)}$$

(也即  $d$  是  $e \pmod{\phi(N)}$  的乘法逆元，可通过扩展欧几里得算法实现)

我们得到了：

公钥:  $(n, e)$

私钥:  $(n, d)$  (通常已带有  $p, q$ , 以便分解)

加密时：

$$\text{密文 } C = m^e \pmod{N}$$

$$\text{明文 } m = C^d \pmod{N}$$

不难看出，RSA 安全建立在“大数难以被分解”。但是 Shor's Algorithm 可以  
以多项式时间解决！

## 4.2. Shor's 算法.

Note that: if  $r^{2k} \equiv 1 \pmod{N}$ , then:

$$(r^k + 1)(r^k - 1) \mid N.$$

至少有一个  $N$  的 non-trivial factor

所以我们将其归向 RSA 简化：

破解 RSA  $\rightarrow$  分解大数  $N = p \cdot q \rightarrow$  找到  $r$ , s.t.  $r$  的阶为强数。

$\rightarrow$  试求出  $\forall r \in \{1, \dots, N\}$  的阶  $\times$

## 4.3 Order-finding

(Step 1) 我们对模数乘法有一个 Oracle  $O$ , s.t.

$$O: |y\rangle \mapsto |(xy) \pmod{N}\rangle$$

我们考虑这样一组基：

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{2\pi i \cdot (-\frac{ks}{r})} |x^k \pmod{N}\rangle$$

有两个性质：

①  $\{|u_s\rangle\}$  是一组 DNB.

i.e.  $\forall s, t$ , we have :

$$\begin{cases} \langle u_s | u_t \rangle = 1 & \text{IFF } s=t \\ \langle u_s | u_t \rangle = 0 & \text{IFF } s \neq t. \end{cases}$$

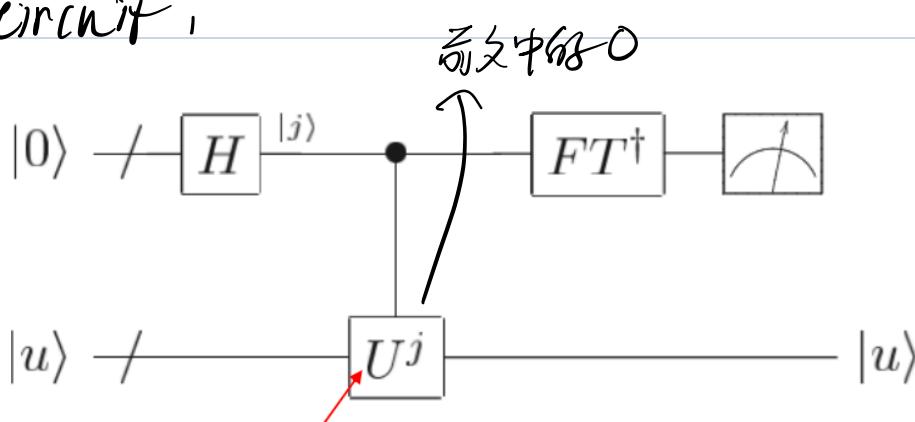
$$\begin{aligned}
 \text{prwf: } & \langle u_s | u_t \rangle \\
 &= \frac{1}{r} \left( \sum_{k=0}^r e^{2\pi i \cdot \left(\frac{ks}{r}\right)} \langle x^k \pmod{N} \rangle \right) \left( \sum_{k=0}^r e^{2\pi i \cdot \left(-\frac{kt}{r}\right)} |x^k \pmod{N} \rangle \right) \\
 &= \frac{1}{r} \sum_{k=0}^r \sum_{k'=0}^r e^{\frac{2\pi i}{r} (sk - tk')} \langle x^{k'} | x^k \rangle \quad \begin{cases} 1, k=k' \\ 0, k \neq k' \end{cases} \\
 &= \frac{1}{r} \sum_{k=0}^r e^{\frac{2\pi i k t}{r}} (s-t) \\
 &= \begin{cases} 0, s \neq t \\ 1, s = t \end{cases} \quad \checkmark \\
 \text{Q.E.D.}
 \end{aligned}$$

③  $|u_s\rangle$  是  $O$  的本征态 eigenstates.

$$O|u_s\rangle = e^{\frac{2\pi i s}{r}} |u_s\rangle$$

$$\begin{aligned}
 \text{proof: } O|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^r e^{2\pi i \left(-\frac{ks}{r}\right)} |x^{k+1} \pmod{N} \rangle \quad \rightarrow \text{目的系统} \\
 &= e^{\frac{2\pi i s}{r}} |u_s\rangle \quad \begin{array}{l} \text{通过 QPE, 我们得到 } \frac{s}{r} \\ \text{我们想要的} \end{array}
 \end{aligned}$$

我们的 circuit:



## 一、QPE Oracle 的实际构造 (Realization of the QPE Oracle)

问题背景：

在标准 QPE 中，我们需要对一个酉算子  $U$  实现其受控版本  $C(U), C(U^2), C(U^4), \dots$

但对模乘法算子  $U|y\rangle = |xy \bmod N\rangle$ ，逐个构造这些受控门效率低。

✓ 更高效的做法：

不单独构造  $U$  再做受控，而是直接构造整个 QPE 所需的联合 oracle：

$$O_{\text{QPE}} = \sum_{j=0}^{2^t-1} |j\rangle\langle j| \otimes U^j$$

这个算子的作用是：

$$|j\rangle|q\rangle \mapsto |j\rangle \cdot U^j|q\rangle = |j\rangle|q \cdot x^j \bmod N\rangle$$

这正是 模幂运算 (modular exponentiation)：给定  $j$ ，计算  $x^j \bmod N$  并作用在寄存器上。

✗ 关键结论：

- 这个 oracle 记作  $O_{x,N}$  (即文中  $O_{L,N}$ )。
- 它可以在  $O((\log N)^3)$  个量子门内实现 (利用经典模幂<sup>63</sup> 有效电路，见 Nielsen & Chuang 书 Box 5.2)。
- 只需一次调用  $O_{x,N}$ ，就完成了 QPE 中所有  $U^{2^k}$  的作用 (通过二进制展开  $j = j_0 + 2j_1 + 4j_2 + \dots$ )。

注意到第一事实：

$$Hr, |1\rangle = \sum_{s=0}^{r-1} |u_s\rangle$$

i.e., 我们不需要知道  $r$  的值，也可以剩余一组  $|u_s\rangle$  的  
均值加！

→ 某整数， $[0, r)$  中任整数。

通过 step 1，我们得到了对  $\frac{s}{r}$  的近似。  
↓ order

step 2. 连分数算法。

→  $\mathcal{O}((\log N)^3)$

continued fraction algorithm

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2r^2}$$

输入一个  $[0, 1]$  实数，给出最近的有理数近似 (一定精度内)

也即我们从测得得到的  $\varphi \rightarrow$  最近的有理数  $\frac{s}{r}$

若  $(r, s) = 1$ ，则可从  $\frac{s}{r}$  推出  $r$ ！

Step3. 若  $x^r \equiv 1 \pmod{N}$ , 则  $\exists$   
 else, 重试

完整 circuit:

