BASICS OF QUANTUM COMPUTING

- $|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

- $|+\rangle := \dfrac{|0\rangle + |1\rangle}{\sqrt{2}}$; $|-\rangle := \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$; $|\pm_Y\rangle = \dfrac{|0\rangle \pm i|1\rangle}{\sqrt{2}}$

- X, Y, Z Pauli : $-\boxed{X}-$, $-\boxed{Y}-$, $-\boxed{Z}-$

- P Pauli $\Rightarrow tr(P) = 0$, $P = P^{+}$, $P^2 = \mathbb{1}$. $tr(XY) = 0$, $tr(YZ) = 0$, $tr(XZ) = 0$

- $X|+\rangle = +|+\rangle$, $X|-\rangle = -|-\rangle$;

  $Y|\pm_Y\rangle = \pm|\pm_Y\rangle$

- $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle \Rightarrow H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $H^2 = \mathbb{1}$, $H^{+} = H$.

- $Z|+\rangle = |-\rangle$, $Z|-\rangle = |+\rangle$

- If $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, $n = 2 \Rightarrow$

$$|0\rangle_{1,2} = |0\rangle_1 \otimes |0\rangle_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle_{1,2} = |0\rangle_1 \otimes |1\rangle_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|2\rangle_{1,2} = |1\rangle_1 \otimes |0\rangle_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|3\rangle_{1,2} = |1\rangle_1 \otimes |1\rangle_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- If $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} \Rightarrow$

$$|0\rangle := |0\rangle_1 \otimes \cdots \otimes |0\rangle_n = |0\rangle^{\otimes N}$$

$$|1\rangle := |0\rangle_1 \otimes \cdots \otimes |1\rangle_n ;$$

$$|2\rangle := |0\rangle_1 \otimes \cdots \otimes |1\rangle_{n-1} \otimes |0\rangle_n ;$$

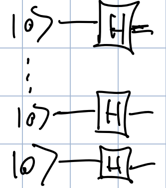
$$\vdots$$

$$|2^n - 1\rangle := |1\rangle_1 \otimes \cdots \otimes |1\rangle \otimes |1\rangle_n$$

- $$H^{\otimes N}|0\rangle^{\otimes N} = (H|0\rangle) \otimes (H|0\rangle) \otimes \cdots \otimes (H|0\rangle) = |+\rangle \otimes \cdots \otimes |+\rangle = |+\rangle^{\otimes N}$$

- $$|+\rangle^{\otimes N} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2^n}} \sum_{x_1, \dots, x_n \in \{0,1\}^n} \overbrace{|x_1\rangle \otimes \cdots \otimes |x_n\rangle}^{|x_1, \dots, x_n\rangle}$$

NOT ENTANGLED!

-  $= H^{\otimes N} |0\rangle^{\otimes N} = |+\rangle^{\otimes N}$

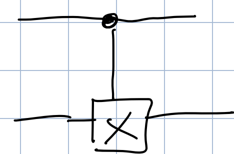
- if $x = 0,1 \Rightarrow H|x\rangle = \dfrac{|0\rangle + (-1)^x |1\rangle}{\sqrt{2}} = \sum_{z=\{0,1\}} \dfrac{(-1)^{x \cdot z}}{\sqrt{2}} |z\rangle$

- $H^{\otimes n} |x_1, \dots, x_n\rangle = \sum_{z_1=\{0,1\}} \dfrac{(-1)^{x_1 z_1}}{\sqrt{2}} |z_1\rangle \otimes \dots \otimes \sum_{z_n=\{0,1\}} \dfrac{(-1)^{x_n z_n}}{\sqrt{2}} |z_n\rangle$

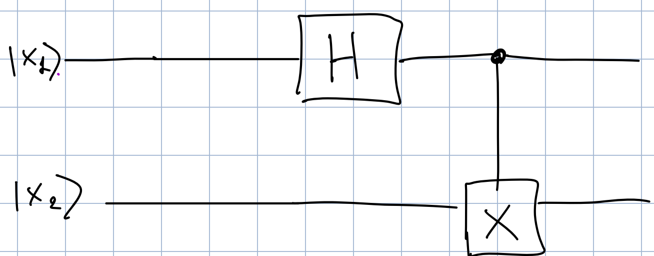
$$= \frac{1}{\sqrt{2^n}} \sum_{z_1,\dots,z_n=\{0,1\}} (-1)^{x_1 z_1 + \dots + x_n z_n} \underline{|z_1, \dots, z_n\rangle}$$

$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle$ with $x \cdot z = x_1 z_1 + \dots + x_n z_n$.

PAULI STRINGS

- $P = P_1 \otimes \dots \otimes P_n$ with $P_i = \{\mathbb{1}, X, Y, Z\} \Rightarrow tr(P) = 0$, $P = P^\dagger$, $P^2 = \mathbb{1}$.
- P^i, P^j Pauli strings $\Rightarrow tr(P_i P_j) = 2^n \delta_{i,j}$.
- $CNOT_{(A,B)} := |0\rangle\langle 0|_A \otimes \mathbb{1} + |1\rangle\langle 1|_B \otimes X_B \equiv$ 

- CNOT generates entanglement:



$= CNOT_{(1,2)} (H \otimes \mathbb{1}) |x_1\rangle \otimes |x_2\rangle =$

$= CNOT_{(1,2)} \left(\dfrac{|0\rangle + (-1)^{x_1} |1\rangle}{\sqrt{2}} \right) \otimes |x_2\rangle =$

$= \dfrac{1}{\sqrt{2}} |0\rangle \otimes |x_2\rangle + \dfrac{(-1)^{x_1}}{\sqrt{2}} |1\rangle \otimes |\bar{x}_2\rangle$

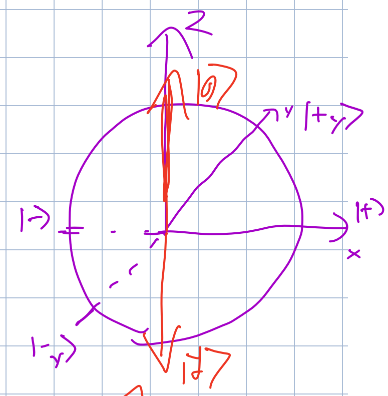
If $x_1, x_2 = 0 \Rightarrow \dfrac{|00\rangle + |11\rangle}{\sqrt{2}}$

- OTHER GATES :

Operator	Gate(s)		Matrix
Pauli-X (X)	$\boxed{X}$	$\oplus$	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = i e^{-i\frac{\pi}{2} X} \underset{\uparrow}{=} \left(\overset{0}{\cos\left(\frac{\pi}{2}\right)} \mathbb{1} - i \overset{1}{\sin\left(\frac{\pi}{2}\right)} X \right) i$
Pauli-Y (Y)	$\boxed{Y}$		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	$\boxed{Z}$		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = i e^{-i\frac{\pi}{2} Z}$
Hadamard (H)	$\boxed{H}$		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	$\boxed{S}$		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{+i\frac{\pi}{2}} \end{pmatrix} = e^{i\frac{\pi}{4}} \begin{pmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = e^{i\frac{\pi}{4}} e^{-i\frac{\pi}{4} Z}$ $= e^{i\frac{\pi}{4}} R_{\hat{z}}\left(\frac{\pi}{2}\right)$
$\pi/8$ (T)	$\boxed{T}$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)	$\bullet$ / $\oplus$		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	$\bullet$ / $\boxed{Z}$	$\bullet$ / $\bullet$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\times$ / $\times$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)	$\bullet$ / $\bullet$ / $\oplus$		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Rotation of π around $\hat{z}$ in the Bloch sphere.

$$\left(P^2 = \mathbb{1} \Rightarrow \quad e^{-i\frac{\theta}{2} P} = \cos\frac{\theta}{2}\, \mathbb{1} - i \sin\frac{\theta}{2}\, P \quad \left(e^{A} = \sum_{i=1}^{\infty} \frac{A^i}{i!} \right) \quad = R_{\hat{P}}(\theta) \right)$$



action of X.

- $C\text{-}U_{(A,B)} = |0\rangle\langle 0|_A \otimes \mathbb{1}_B + |1\rangle\langle 1|_A \otimes U_B$

- $HZH = X \quad , \quad X = HZH$

$$\left(\begin{aligned} &HZH|0\rangle = HZ|+\rangle\, H|-\rangle = |1\rangle = X|0\rangle \\ &HZH|1\rangle = X|1\rangle \end{aligned} \Rightarrow HZH = X \underset{H^2 = \mathbb{1}}{\Rightarrow} Z = HXH \right)$$

- $(HS^{+})^{+} Z (HS^{+}) = Y \qquad (SXS^{+} = Y)$

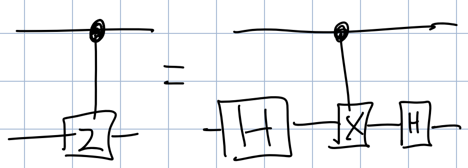
$$(HS^\dagger)^\dagger Z\, HS^\dagger = SHZHS^\dagger = SXS^\dagger = Y$$

$$\left(\begin{aligned} SXS^\dagger|0\rangle &= i|1\rangle = Y|0\rangle \\ SXS^\dagger|1\rangle &= -i|0\rangle = Y|1\rangle \end{aligned}\right)$$

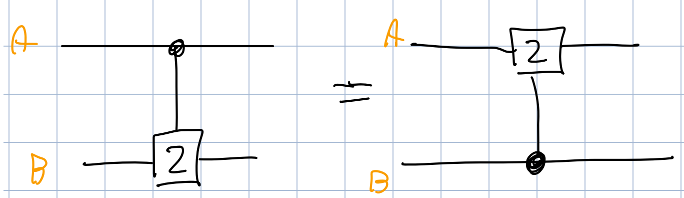
- $$C\text{-}Z = |0\rangle\langle 0| \otimes \overset{HH}{\overset{\shortparallel}{\mathbb{1}}} + |1\rangle\langle 1| \otimes \overset{HXH}{\overset{\shortparallel}{Z}}$$

$$\Rightarrow C\text{-}Z = \mathbb{1}\otimes H\,(C\text{-NOT})\,\mathbb{1}\otimes H \quad , \quad C\text{-NOT} = \mathbb{1}\otimes H\,(C\text{-}Z)\,\mathbb{1}\otimes H$$

- $H^2 = \mathbb{1}$
- $HXH = Z$



- $$C\text{-}Z_{(A,B)} = C\text{-}Z_{(B,A)}$$



- They act equally on computational bases:

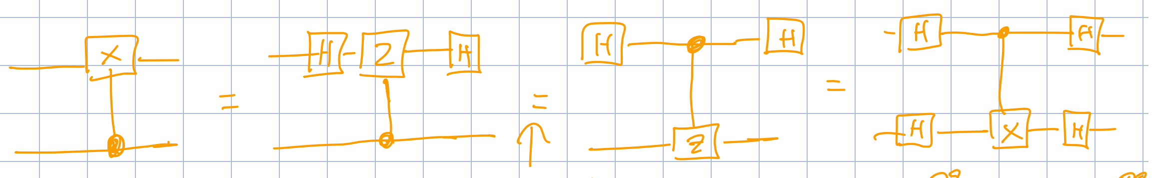
$$C\text{-}Z_{(A,B)}\, |x_A\rangle \otimes |x_B\rangle = (-1)^{x_A \cdot x_B}\, |x_A\rangle \otimes |x_B\rangle$$

$$= C\text{-}Z_{(B,A)}\, |x_A\rangle \otimes |x_B\rangle$$

- $$C\text{-NOT}_{(B,A)} = H^{\otimes 2}\; C\text{-NOT}_{(A,B)}\; H^{\otimes 2}$$

PROOF:

$C\text{-NOT}_{(BA)} =$



$C\text{-}Z_{(A,B)} = C\text{-}Z_{(B,A)}$

$= H^{\otimes 2}\; C\text{-NOT}_{(B,A)}\; H^{\otimes 2}$

• $\mathrm{SWAP}\,|i\rangle \otimes |j\rangle := |j\rangle \otimes |i\rangle \qquad \forall\, i,j = 0,1$

$\mathrm{SWAP}\,|\psi\rangle \otimes |\phi\rangle = |\phi\rangle \otimes |\psi\rangle$

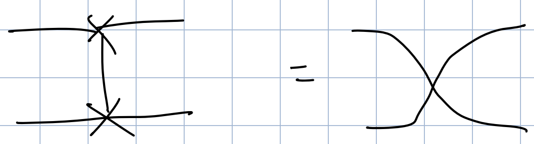
PROOF

$$\mathrm{SWAP}\,|\psi\rangle \otimes |\phi\rangle = \sum_{i,j} \psi_i \phi_j\, \mathrm{SWAP}\,|i\rangle \otimes |j\rangle = \sum_{i,j} \psi_i \phi_j\, |j\rangle \otimes |i\rangle = |\phi\rangle \otimes |\psi\rangle$$

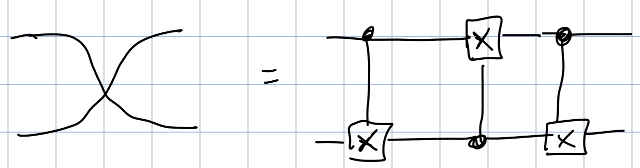
$|\psi\rangle = \sum_i \psi_i |i\rangle$

$|\phi\rangle = \sum_j \phi_j |j\rangle$

SWAP :=

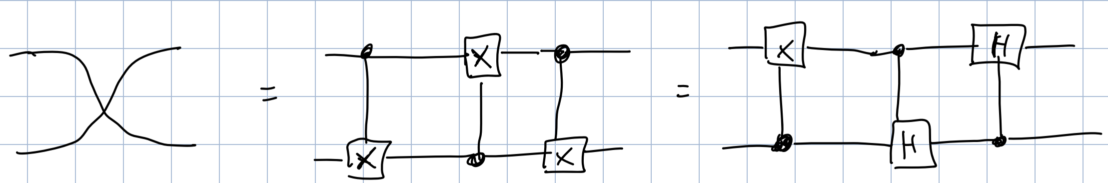


$$\mathrm{SWAP}_{(A,B)} = \text{C-NOT}_{(A,B)}\ \text{C-NOT}_{(B,A)}\ \text{C-NOT}_{(A,B)}$$



PROOF:

They act equally on $|00\rangle, |01\rangle, |10\rangle, |11\rangle$



1. **Quantum Fourier transform.** (9 points: 1+4+2+2) Perhaps at the heart of the majority of modern quantum algorithms lies the *phase estimation algorithm*. For this reason, it is crucial in the field of quantum computation to be familiar with phase estimation. It relies on an efficient implementation of the *quantum Fourier transform*, to which we devote this excercise.

In classical numerics the discrete Fourier transform (DFT) is defined as the linear map $F : \mathbb{C}^N \to \mathbb{C}^N$, $x \mapsto y$ with $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left\{\frac{2\pi \mathrm{i} jk}{N}\right\}$. The quantum Fourier transform is analogously defined as the unitary operation $\mathcal{F} : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, $|j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left\{\frac{2\pi \mathrm{i} jk}{2^n}\right\} |k\rangle$. (Note the identification $N = 2^n$.)

a) Look-up the computational complexity of the fastest classical algorithm for the Fourier transform.

Given $x \in \mathbb{C}^N$, $(\mathrm{DFT}(x))_k \equiv y_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \, e^{i\left(\frac{2\pi j k}{N}\right)}$, $k = 0, 1, \dots, N-1$.

The fastest algorithm known for DFT is FFT (Fast Fourier Transform), and runs in $O(N \log N)$ operations.

(It exploits symmetries of DFT(x))

Read WIKIPEDIA PAGE of DFT.

The quantum Fourier transform can be implemented using the Hadamard gates H,

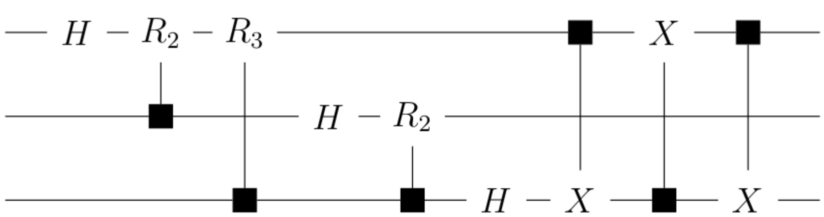
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1}$$

the controlled phase gate that applies

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{2\pi \mathrm{i}/2^k} \end{pmatrix} \tag{2}$$

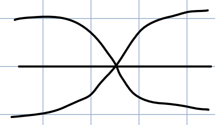
on a *target* qubit if a *control* qubit is in the state $|1\rangle$ (and the identity if the control is in $|0\rangle$) and CNOT gates. Note that in circuit diagrams controlled gates are conventionally represented by boxes on the target wires linked to dots on the control wires.

b) Show that the following circuit implements the three qubit quantum Fourier transform

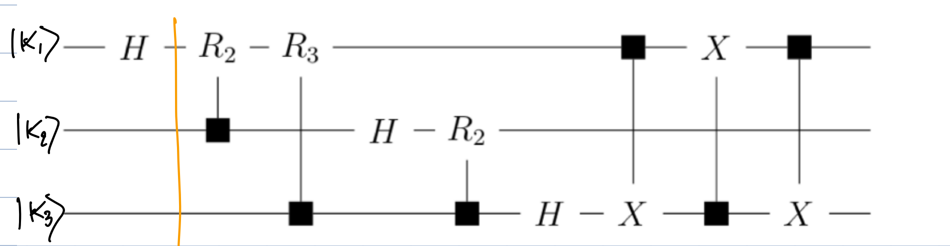


Hint: restrict your attention to generic computational basis states as inputs.

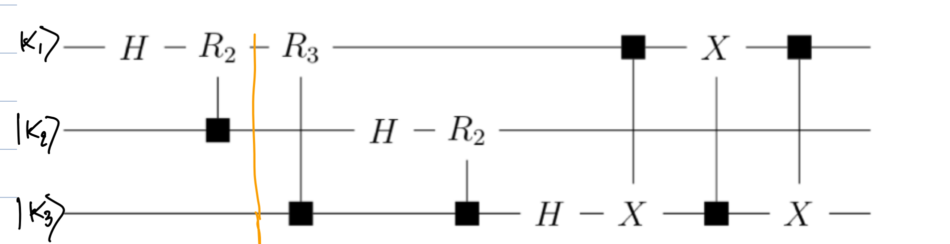
• This is a SWAP.



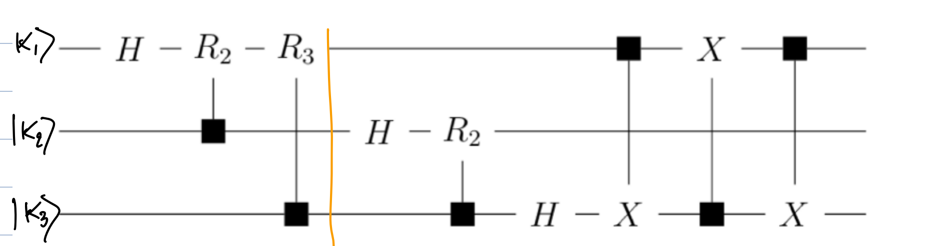
$R_K|0\rangle = |0\rangle$, $R_K|1\rangle = e^{\frac{2\pi i}{2^K}}|1\rangle$



$$|\psi_1\rangle = H\otimes 1\otimes 1\left(|k_1\rangle\otimes|k_2\rangle\otimes|k_3\rangle\right) = \left(\frac{|0\rangle + (-1)^{k_1}|1\rangle}{\sqrt{2}}\right)\otimes|k_2\rangle\otimes|k_3\rangle$$

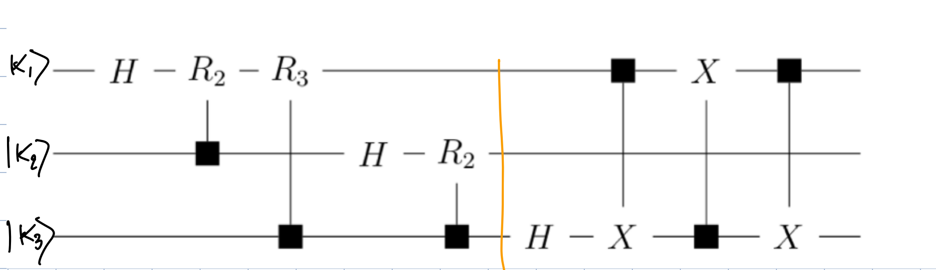


$$|\psi_2\rangle = C\text{-}R_{2_{(2,1)}}|\psi_1\rangle = \left(\frac{|0\rangle + (-1)^{k_1} e^{\frac{2\pi i}{2^2}\cdot k_2}|1\rangle}{\sqrt{2}}\right)\otimes|k_2\rangle\otimes|k_3\rangle$$

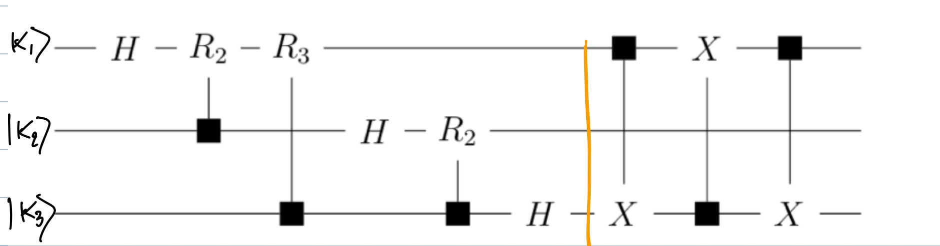


$$|\psi_3\rangle = C\text{-}R_{3_{(3,1)}}|\psi_2\rangle = \left(\frac{|0\rangle + e^{\frac{i2\pi k_1}{2} + \frac{2\pi i}{2^2}k_2 + \frac{2\pi i}{2^3}k_3}|1\rangle}{\sqrt{2}}\right)\otimes|k_2\rangle\otimes|k_3\rangle$$

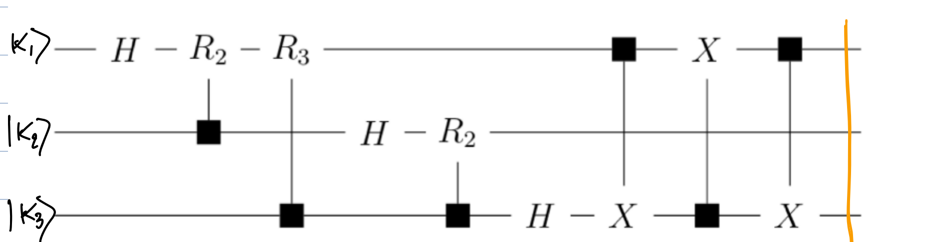
$$= \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_1}{2} + \frac{k_2}{2^2} + \frac{k_3}{2^3}\right)}|1\rangle}{\sqrt{2}}\right)\otimes|k_2\rangle\otimes|k_3\rangle$$



$$|\psi_4\rangle = \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_1}{2}+\frac{k_2}{2^2}+\frac{k_3}{2^3}\right)}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_2}{2}+\frac{k_3}{2^2}\right)}|1\rangle}{\sqrt{2}}\right) \otimes |k_3\rangle$$



$$|\psi_4\rangle = \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_1}{2}+\frac{k_2}{2^2}+\frac{k_3}{2^3}\right)}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_2}{2}+\frac{k_3}{2^2}\right)}}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + \overbrace{e^{i\frac{2\pi k_3}{2}}}^{(-1)^{k_3} \,=}|1\rangle}{\sqrt{2}}\right)$$



$$|\psi_5\rangle = \left(\frac{|0\rangle + e^{i\frac{2\pi}{2}k_3}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_2}{2}+\frac{k_3}{2^2}\right)}}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_1}{2}+\frac{k_2}{2^2}+\frac{k_3}{2^3}\right)}|1\rangle}{\sqrt{2}}\right)$$

INTERLUDE ON QFT:

- QFT: $$|\tilde{x}_K\rangle := \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{i\left(\frac{2\pi}{N} K x\right)} |x\rangle$$

- $|x\rangle$ is ONB basis for assumption.
- This implies $|\tilde{x}_K\rangle$ is ONB basis.

PROOF

- $$\langle \tilde{x}_l | \tilde{x}_m\rangle = \frac{1}{N} \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} e^{-i\frac{2\pi}{N} l j_1} e^{i\frac{2\pi}{N} m j_2} \langle j_1 | j_2\rangle$$

$$= \frac{1}{N} \sum_{j_1=0}^{N-1} e^{i\frac{2\pi}{N} j_1 (l-m)} = \frac{1}{N}\left(N\, \delta_{\frac{2\pi}{N}(l-m),0}\right)$$

$$\left(\sum_{j=0}^{N-1} e^{i\alpha j} \underset{\substack{\uparrow \\ \text{GEOMETRIC} \\ \text{SUM}}}{=} \begin{cases} N, & \alpha = 0 \pmod{2\pi} \\ \dfrac{e^{i\alpha N}-1}{e^{i\alpha}-1}, & \alpha \neq 0 \pmod{2\pi} \end{cases} \right)$$

$$= \delta_{l,m}$$

- $\exists\, U_{QFT} : |\tilde{x}_K\rangle = U_{QFT}|K\rangle \quad \forall |K\rangle$ in the basis.

$$\Rightarrow \quad U_{QFT} = U_{QFT} \sum_{K=0}^{N-1} |K\rangle\langle K| = \frac{1}{\sqrt{N}} \sum_{\substack{j=0 \\ K=0}}^{N-1} e^{i\left(\frac{2\pi}{N} K j\right)} |j\rangle\langle K|$$

$$= \frac{1}{\sqrt{N}} \begin{array}{c} \begin{matrix} |0\rangle\langle 0| & |0\rangle\langle 1| & & \end{matrix} \\ \begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \cdots & \omega^{0(N-1)} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \cdots & \omega^{1\cdot(N-1)} \\ \vdots & \vdots & & \vdots \\ \omega^{(N-1)\cdot 0} & \omega^{(N-1)\cdot 1} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \end{array} \quad \text{with } \omega = e^{i\frac{2\pi}{N}}$$

- $$U_{QFT}^{-1} = U_{QFT}^{+} = \frac{1}{\sqrt{N}} \sum_{\substack{j=0 \\ K=0}}^{N-1} e^{-i\left(\frac{2\pi}{N} K j\right)} |K\rangle\langle j|$$

$$|\tilde{x}_k\rangle := \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{i\left(\frac{2\pi}{N} kx\right)} |x\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x_1,\dots,x_n \in \{0,1\}^n} e^{\left(i\frac{2\pi}{2^n}k\right)\left(\sum_{l=0}^{n-1} x_{n-l}\, 2^l\right)} |x_1\rangle \otimes \dots \otimes |x_n\rangle =$$

$$\left(\begin{array}{l} \cdot\ N = 2^n \\ \cdot\ |x\rangle = |x_1\rangle \otimes \dots \otimes |x_n\rangle \\ \quad x = x_n 2^0 + x_{n-1}\cdot 2 + \dots + x_1 \cdot 2^{n-1} \end{array} \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x_1,\dots,x_n \in \{0,1\}^n} \prod_{l=0}^{n-1} e^{\left(i\frac{2\pi}{2^n}k\right) x_{n-l}\, 2^l} |x_1\rangle \otimes \dots \otimes |x_n\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x_1,\dots,x_n \in \{0,1\}^n} e^{\left(i\frac{2\pi}{2^n}k\right) x_1 2^{n-1}} |x_1\rangle \otimes \dots \otimes e^{\left(i\frac{2\pi}{2^n}k\right) x_n \cdot 2^0} |x_n\rangle =$$

$$= \left(\frac{1}{\sqrt{2}} \sum_{x_1=0}^{1} e^{\left(i\frac{2\pi}{2^n}k\right) x_1 2^{n-1}} |x_1\rangle\right) \otimes \dots \otimes \left(\frac{1}{\sqrt{2}} \sum_{x_n=0}^{1} e^{\left(i\frac{2\pi}{2^n}k\right) x_n} |x_n\rangle\right) =$$

$$= \left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}k\cdot 2^{n-1}} |1\rangle}{\sqrt{2}}\right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}k} |1\rangle}{\sqrt{2}}\right)$$

$$= \left(\frac{|0\rangle + e^{i\frac{2\pi k}{2^1}} |1\rangle}{\sqrt{2}}\right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}k} |1\rangle}{\sqrt{2}}\right)$$

$$= \bigotimes_{l=1}^{n} \left(\frac{|0\rangle + e^{i\frac{2\pi k}{2^l}} |1\rangle}{\sqrt{2}}\right)$$

$$= \bigotimes_{l=1}^{n} \left(\frac{|0\rangle + e^{i\frac{2\pi}{2^l}\left(2^0 k_n + 2^1 k_{n-1} + \dots + 2^{n-1} k_1\right)} |1\rangle}{\sqrt{2}}\right)$$

$$= \bigotimes_{l=1}^{n} \left(\frac{|0\rangle + e^{i2\pi\left(\frac{2^0 k_n}{2^l} + \frac{2^1 k_{n-1}}{2^l} + \dots + \frac{2^{l-1}}{2^l} k_{n-l+1}\right)} |1\rangle}{\sqrt{2}}\right)$$

$$= \bigotimes_{l=1}^{n} \left(\frac{|0\rangle + e^{i2\pi(2^{-l}k_n + 2^{-l+1}k_{n-1} + \dots + 2^{-1}k_{n-l+1})}|1\rangle}{\sqrt{2}} \right) = U_{QFT}(|k_1\rangle \otimes \dots \otimes |k_n\rangle)$$

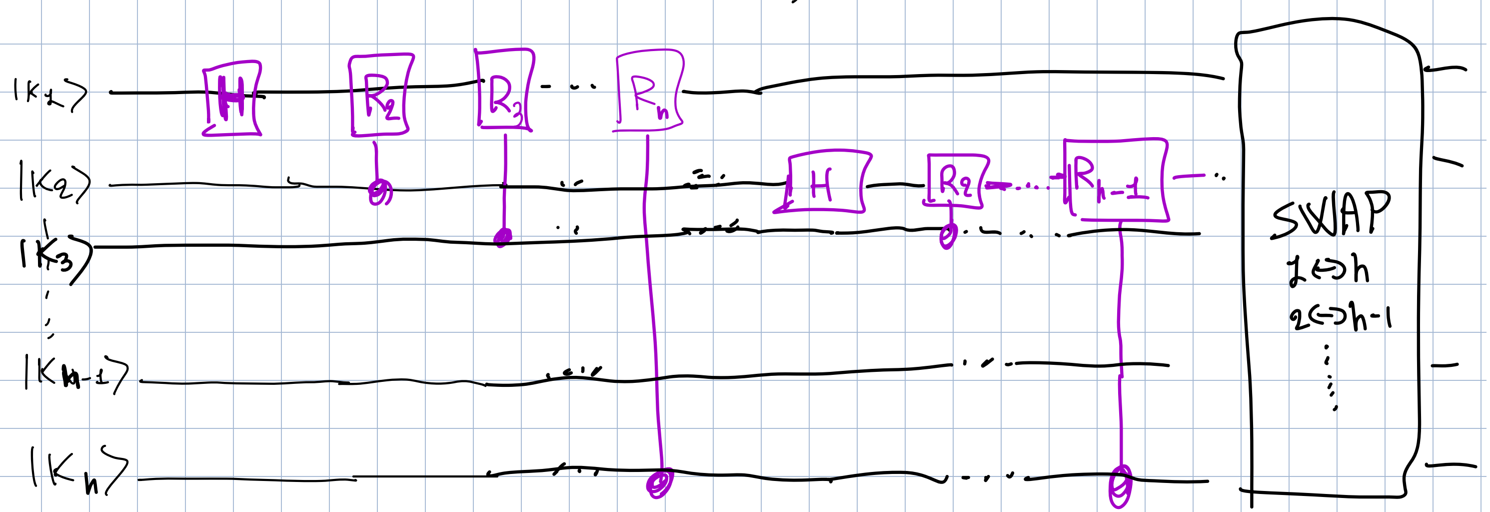
- For 3-QUBITS we had exactly this:

$$|\psi\rangle = U|k_1\rangle\otimes|k_2\rangle\otimes|k_3\rangle = \left(\frac{|0\rangle + e^{i\frac{2\pi k_3}{2}}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_2}{2}+\frac{k_3}{2^2}\right)}}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{i2\pi\left(\frac{k_1}{2}+\frac{k_2}{2^2}+\frac{k_3}{2^3}\right)}|1\rangle}{\sqrt{2}}\right)$$

c) How does this generalise to the n qubit quantum Fourier transform?

$$|\tilde{x}_k\rangle := \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{i\left(\frac{2\pi kx}{N}\right)} |x\rangle =$$

$$= \bigotimes_{l=1}^{n} \left(\frac{|0\rangle + e^{i2\pi(2^{-l}k_n + 2^{-l+1}k_{n-1} + \dots + 2^{-1}k_{n-l+1})}|1\rangle}{\sqrt{2}} \right)$$



d) What is the circuit complexity of the quantum Fourier transform and how does it compare to the classical DFT algorithms?

Note that the quantum Fourier transform can in fact be approximately implemented with only $\mathcal{O}(n \log n)$ gates [1].

Our circuit complexity (i.e. # of gates used) is:

$$\underbrace{n + (n-1) + (n-2) + \dots 1}_{= \frac{n(n+1)}{2} = O(n^2)} + \underbrace{\# \text{ SWAP}}_{= O(n)}$$

$$= O(n^2)$$

So our QFT algo has gate complexity $O(n^2)$.

We saw that classical DFT (with FFT) has complexity $O(N \log N)$ where $N = 2^n$ $\}$ # bits.

$$= O(2^n n)$$

2. **Stabilizer quantum computation.** (11 Points: 3+2+1+2+1+1+1)

One of the most celebrated results in quantum computation is a statement about the resource costs of simulating quantum computations on a classical computers. The *Gottesman-Knill theorem* states that quantum computations composed of *Clifford gates* with *stabilizer states* as inputs and a final measurement in the computational basis can be classically simulated in the sense that there exists a classical algorithm with polynomial runtime which can sample from the output distribution of such a computation. Furthermore, the so-called stabilizer formalism plays an important rôle in the development of quantum error correction.

In this problem we will trace the reasoning underlying this result. Throughout, we will let n be the number of qubits and hence $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ be the Hilbert space. Let us start with some definitions

(i) Let $G_1 = \{\pm\mathbb{1}, \pm X, \pm Y, \pm Z, \pm\mathbb{1}, \pm \mathrm{i}X, \pm\mathrm{i}Y, \pm\mathrm{i}Z\}$ be the single-qubit *Pauli group* where multiplication is the group operation.[2]

(ii) Let $G_n := \{\bigotimes_{i=1}^n P_i, P_i \in G_1\}$ be the n-qubit Pauli group.

(iii) A *stabilizer state* is a quantum state $|\psi\rangle \in \mathcal{H}$ that is uniquely (up to a global phase) described by a set $\mathcal{S}_{|\psi\rangle} = \{S_1, \dots, S_n\} \subset G_n$ satisfying $S_i|\psi\rangle = +1|\psi\rangle$. We call the generalised pauli-operators S_i the stabilizers of $|\psi\rangle$.[3] We note that stabilizers are linearly independent and commutative with each others.

(iv) A Clifford operator C is a unitary on $\mathcal{H}$ which leaves G_n invariant, i.e. for all $g \in G_n$ it holds that $CgC^\dagger \in G_n$. In group theoretic slang the Clifford group $\mathcal{C} \subset U(2^n)$ is the normalizer of G_n.

Ok, now we are ready to begin.

a) Show that the set $\mathcal{S} = \{Z_1, Z_2, \dots, Z_n\}$ uniquely stabilizes the state $|0\rangle^{\otimes n}$, where we use the notation $Z_i = \mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes \underbrace{Z}_{i\text{-th qubit}} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1}$ for the operator acting as Z on the i-th qubit and as the identity on all other qubits.

- $|\psi\rangle = \sum_{x_1,\dots,x_n \in \{0,1\}^n} c_{x_1,\dots,x_n} |x_1,\dots,x_n\rangle$ with $c_{x_1,\dots,x_n} = \langle x_1,\dots,x_n|\psi\rangle \in \mathbb{C}$.

- $Z_i|\psi\rangle = |\psi\rangle \quad \forall\, i = 1,\dots,n.$

- $Z_i|\psi\rangle = \sum_{x_1,\dots,x_n \in \{0,1\}^n} c_{x_1,\dots,x_n} Z_i|x_1,\dots,x_n\rangle =$

$$= \sum_{x_1,\dots,x_n \in \{0,1\}^n} c_{x_1,\dots,x_n} (-1)^{x_i} |x_1,\dots,x_n\rangle$$

$$\overset{Z_i|\psi\rangle = |\psi\rangle}{\Longrightarrow} \sum_{x_1,\dots,x_n \in \{0,1\}^n} c_{x_1,\dots,x_n} (-1)^{x_i} |x_1,\dots,x_n\rangle = \sum_{x_1,\dots,x_n \in \{0,1\}^n} c_{x_1,\dots,x_n} |x_1,\dots,x_n\rangle$$

$$\Longleftrightarrow \quad c_{x_1,\dots,x_n} (-1)^{x_i} = c_{x_1,\dots,x_n} \quad \forall i=1,\dots,n \Longrightarrow \quad c_{0,\dots,0} = 1, \quad c_{0,\dots,1} = 0, \quad \vdots \quad c_{1,\dots,1} = 0 \quad \Rightarrow |\psi\rangle = |0\rangle^{\otimes n}$$

$|x_1,\dots,x_n\rangle$ indipendent

b) Show that n stabilizers suffice to uniquely characterize an arbitrary state in the *Clifford orbit* of $|0\rangle^{\otimes n}$, that is the states $|\psi\rangle$ for which there exists a (unique) Clifford operator C such that $|\psi\rangle = C\,|0\rangle^{\otimes n}$.

- $S_{|0\rangle^{\otimes n}} = \{Z_1, \dots, Z_n\}$ stabilize $|0\rangle^{\otimes n}$ uniquely.

- $|\psi\rangle := C|0\rangle^{\otimes n} \Rightarrow \underbrace{|\psi\rangle}_{= C|0\rangle^{\otimes n}} = C Z_i |0\rangle^{\otimes n} = (C Z_i C^{\dagger}) C |0\rangle^{\otimes n}$

$\Rightarrow C Z_i C^{\dagger}$ stabilizes $|\psi\rangle \quad \forall i$.

- $S_{|\psi\rangle} = \{C Z_1 C^{\dagger}, \dots, C Z_n C^{\dagger}\}$.

- They commute and they are indipendent. $\Rightarrow S_{|\psi\rangle}$ stabilize a 1-dim vector space.

(check "STABILIZERS FORMALISM" PDF FOR MORE DETAILS).

c) Give a stabilizer representation of $|+\rangle \otimes |0\rangle \otimes |-\rangle$.

- $\{X_1, Z_2, -X_3\}$

Any Clifford operator can be expressed as a product of single- and two-qubit Clifford operators, and indeed as a product from the generating set $\{CNOT, H, S\}$, where

$$S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3}$$

d) Show that this gate set is sufficient to generate all Pauli matrices starting from any single-qubit Pauli matrix.

I need to show that $\forall\, P \subseteq \{\mathbb{1}, X, Y, Z\}^{\otimes 2} / \mathbb{1}^{\otimes 2}$

$\exists\, C$ generated by $\{CNOT, H, S\}$ such that $C Z_1 C^{\dagger} = P$

(WLOG we could start by Z_i)

- $H_1 Z_1 H_1 = X_1$

$HZH|0\rangle = HZ|+\rangle = H|-\rangle = |1\rangle = X|0\rangle$

Analog. for $|1\rangle$.

- $S_1 X_1 S_1^{\dagger} = Y_1 \quad \Rightarrow \quad (S_1 H_1) Z_1 (H S_1^{\dagger}) = Y_1$

$SXS^{\dagger}|0\rangle = S|1\rangle = i|1\rangle = Y|0\rangle$

Analog. for $|1\rangle$.

- $CNOT_{1,2}\, X_1\, CNOT_{1,2} = X_1 X_2$

$CNOT\, X_1\, CNOT\, |x_1, x_2\rangle = CNOT\, X_1 |x_1, x_1 \oplus x_2\rangle =$

$= CNOT\, |\overline{x_1}, x_1 \oplus x_2\rangle =$

$= |\overline{x_1}, x_1 \oplus x_2 \oplus \overline{x_1}\rangle$

$= |\overline{x_1}, \overline{x_2}\rangle = X_1 X_2 |x_1, x_2\rangle$

- $(CNOT_{1,2}\, H_1) Z_1 (H_1\, CNOT_{1,2}) = X_1 X_2$

- $X_1 Z_2 = H_2\, X_1 X_2\, H_2 =$

$$= \left(H_2\, CNOT_{1,2}\, H_1\right) Z_1 \left(H_1\, CNOT_{1,2}\, H_2\right)$$

- $Z_1 Z_2 = H_1 H_2\, X_1 X_2\, H_1 H_2$

$$= \left(\underbrace{H_1 H_2\, CNOT_{1,2}\, H_1}_{\ddot{U}_{(1,2)}}\right) Z_1 \left(H_1\, CNOT_{1,2}\, H_1 H_2\right)$$

- To generate $Z_1 Z_2 Z_3$, I can use $Z_1 \xrightarrow{U_{(1,2)}} Z_1 Z_2 \xrightarrow{U_{(2,3)}} Z_1 Z_2 Z_3$ and similarly for other Paulis.

Operation	Input	Output
controlled-NOT	X_1	$X_1 X_2$
	X_2	X_2
	Z_1	Z_1
	Z_2	$Z_1 Z_2$
H	X	Z
	Z	X
S	X	Y
	Z	Z
X	X	X
	Z	$-Z$
Y	X	$-X$
	Z	$-Z$
Z	X	$-X$
	Z	Z

Figure 10.7. Transformation properties of elements of the Pauli group under conjugation by various common operations. The controlled-NOT has qubit 1 as the control and qubit 2 as the target.

e) Argue that one can efficiently (in the number of qubits and gates) determine the stabilizer set of a state generated by a (known) Clifford circuit (comprising $CNOT, H, S$ gates) applied to a stabilizer state.

- Given $S \subseteq G_n$: $S = \{S_1, \dots, S_n\}$.
- If we apply a gate C from $\{CNOT, H, S\} \Rightarrow S_{C|\psi\rangle} = \{C S_1 C^\dagger, \dots, C S_n C^\dagger\}$.
- C is a 2-qubits gate and acts non-trivially only on 2-qubits of the Pauli string S_i.
  We know how to compute $C S_i C^\dagger$ from the table above.
- We need to do this operation for each of the n qubits.
- So if we have "m" gates $\Rightarrow O(mn)$ computational time.

From the above reasoning, we conclude that we can efficiently simulate the effect of a Clifford circuit applied to a stabilizer state by keeping track of the stabilizers.

Now, let us assume that we measure the first qubit in the Z basis.

f) Assume Z_1 commutes with all stabilizers. What is the probability of obtaining outcome $+1$?

- Let $S_i \in S_{|\psi\rangle}$.
  $Z_1|\psi\rangle = Z_1 S_i|\psi\rangle = S_i Z_1|\psi\rangle \Rightarrow Z_1|\psi\rangle \in S_{|\psi\rangle}$.
- $S_{|\psi\rangle}$ is 1-dim. $\Rightarrow Z_1|\psi\rangle = e^{i\phi}|\psi\rangle \underset{(Z_1)^2 = \mathbb{1}}{\Longrightarrow} Z_1|\psi\rangle = \pm|\psi\rangle$

  $\Longrightarrow \langle\psi|Z_1|\psi\rangle = \pm 1$

- We measure the POVM $\{\underbrace{|0\rangle\langle 0|\otimes\mathbb{1}}_{E_1}, \underbrace{|1\rangle\langle 1|\otimes\mathbb{1}}_{E_2 = \mathbb{1} - |0\rangle\langle 0|\otimes\mathbb{1}}\}$ $(E_1 + E_2 = \mathbb{1}, E_1 \geq 0, E_2 \geq 0)$

- $P(+1) = \mathrm{tr}\left(|\psi\rangle\langle\psi| \left(|0\rangle\langle 0| \otimes \mathbb{1}\right)\right) = \mathrm{tr}\left(|\psi\rangle\langle\psi| \left(\frac{\mathbb{1}+Z}{2}\right) \otimes \mathbb{1}\right) =$

$$= \frac{1}{2} + \frac{1}{2}\mathrm{tr}\left(|\psi\rangle\langle\psi| Z_1\right)$$

$$= \frac{1}{2} + \frac{1}{2}\langle\psi|Z_1|\psi\rangle$$

- $\left(P(0) = 1 - P(+1) = \frac{1}{2} - \frac{1}{2}\langle\psi|Z_1|\psi\rangle. \right)$

- So given that $\langle\psi|Z_1|\psi\rangle = \pm 1$ we have:

$$P(+1) = \begin{cases} 1, & \text{if } \langle\psi|Z_1|\psi\rangle = +1 \\ 0, & \text{if } \langle\psi|Z_1|\psi\rangle = -1 \end{cases}$$

- How do we know if $\langle\psi|Z_1|\psi\rangle = +1$ or -1?

- If we know the clifford gates applied to $|0\rangle^{\otimes n}$ to create $|\psi\rangle$ i.e.

$|\psi\rangle = C_m \cdots C_2 C_1 |0\rangle^{\otimes n}$. with $C_i \in \{CNOT, H, S\}$

Then $\langle\psi|Z_1|\psi\rangle = {}^{\otimes n}\langle 0| C^\dagger Z_1 C |0\rangle^{\otimes n} = {}^{\otimes n}\langle 0| C_1^\dagger C_2^\dagger \cdots C_m^\dagger Z_1 C_m \cdots C_1 |0\rangle^{\otimes n} =$

$$= {}^{\otimes n}\langle 0| C_1^\dagger C_2^\dagger \cdots C_{m-1}^\dagger P_m C_{m-1} \cdots C_1 |0\rangle^{\otimes n} =$$

$P_m := C_m^\dagger Z_1 C_m$ is a Pauli which can be computed lookin at the table in ex. d).

$$= {}^{\otimes n}\langle 0| P |0\rangle^{\otimes n}$$

$P_i = C_i^\dagger P_{i+1} C_i$

← "efficient" to compute: $O\left(\text{TIME to compute } P + \text{TIME TO COMPUTE } \langle\psi|P|\psi\rangle\right) = O(m+n)$

(Not $O(\exp(n))$)

• If we don't know the gates, but only the stabilizers:

We observe that $Z_1|\psi\rangle = \pm|\psi\rangle \Rightarrow +Z_1$ or $-Z_1$ is in $S_{|\psi\rangle}$.

(Not both of them otherwise $S_{|\psi\rangle} = \{0\}$.)

Remember that $[S_i, S_j] = 0$ and $S_i^2 = \mathbb{1}$.

We need to check if $\exists\, x_1, \dots, x_n \in \{0,1\}$: $S_1^{x_1} \cdots S_n^{x_n} = Z_1$

or

$S_1^{x_1} \cdots S_n^{x_n} = -Z_1$

One of the two cases is true.

• Is there an efficient algo to check this?
(not $O(\exp(n))$)

• We define a 2n-dim. vector representation $R(P)$ of a Pauli $P = \pm P_1 \otimes P_2 \otimes \cdots \otimes P_n$

$P_i = \mathbb{1} \iff (R(P))_i = 0, \quad (R(P))_{n+i} = 0$

$P_i = X \iff (R(P))_i = 1, \quad (R(P))_{n+i} = 0$

$P_i = Z \iff (R(P))_i = 0, \quad (R(P))_{n+i} = 1$

$P_i = Y \iff (R(P))_i = 1, \quad (R(P))_{n+i} = 1$

e.g. $P = X \otimes Z \otimes \mathbb{1} \otimes \underset{= iZX}{Y} \Rightarrow R(P) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \hline 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\0\\0\\1\end{matrix}}\right\} X \\ \left.\vphantom{\begin{matrix}0\\1\\0\\1\end{matrix}}\right\} Z \end{matrix}$

• We don't consider phases in this representation.

- $P^{(A)} P^{(B)} = P^{(C)} \implies \left(R(P^{(A)}) + R(P^{(B)}) = R(P^{(C)}) \right) \mod (2).$

$P^{(A)} P^{(B)} = \left(P_1^{(A)} \otimes \cdots \otimes P_n^{(A)}\right) \left(P_1^{(B)} \otimes \cdots \otimes P_n^{(B)}\right) = P_1^{(A)} P_1^{(B)} \otimes \cdots \otimes P_n^{(A)} P_n^{(B)}$

We can verify it case by case.

e.g. $P_j^{(A)} = X$, $P_j^{(B)} = Y$ $\implies XY = iZ = P_j^{(C)}$

$$\underset{1}{\underset{\shortparallel}{(R(X))_j}} + \underset{1}{\underset{\shortparallel}{(R(Y))_j}} = (R(XY))_j = \underset{0}{\underset{\shortparallel}{(R(Z))_j}}$$

$1 + 1 = 0$ ok!

$$\underset{0}{\underset{\shortparallel}{(R(X))_{j+n}}} + \underset{1}{\underset{\shortparallel}{(R(Y))_{j+n}}} = (R(XY))_{j+n} = \underset{1}{\underset{\shortparallel}{(R(Z))_{j+n}}}$$

$0 + 1 = 1$ ok!

- $Q = S_n^{x_n} \cdots S_1^{x_1}$ with $x_1, \dots, x_n \in \{0,1\} \implies R(Q) = x_1 R(S_1) + \dots + x_n R(S_n)$

- $$R(Q) = x_1 R(S_1) + \dots + x_n R(S_n) = \underbrace{\begin{pmatrix} R(S_1) & R(S_2) & \cdots & R(S_n) \end{pmatrix}}_{2n \times n \text{ matrix}\ =:\ \underline{\underline{R}}_S} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} =: \underline{\underline{R}}_S \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- We know that $\exists\ x_1, \dots, x_n \in \{0,1\} : S_1^{x_1} \cdots S_n^{x_n} = Z_1$,
or
$S_1^{x_1} \cdots S_n^{x_n} = -Z_1$

$\Rightarrow \quad R(Z_1) = \underline{\underline{A_S}} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$

$\nearrow$
linear system of equation. It can be solved in $O(n^3)$ time.
and find $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$.

- We can compute $S_1^{x_1} \cdots S_n^{x_n}$ and check if it is $S_1^{x_1} \cdots S_n^{x_n} = +Z_1$
  $S_1^{x_1} \cdots S_n^{x_n} = -Z_1$.

One can show that in case Z_1 does not commute with all stabilizers, (generators) one can find an alternative set of stabilizers such that it anti-commutes with one of them but commutes with all remaining ones.

g) Use the existence of such a stabilizer to show that the measurement outcome is uniformly random. What is the post-measurement state?

In fact, this generalizes to the measurement of an arbitrary Pauli operator $g \in G_n$. Therefore, we see that checking commutation with the stabilizers gives us a recipe for efficiently simulating samples resulting from computational basis measurements.

- Z_1 does not commute with at least one of the stab. generators: $\{S_1, \dots, S_n\}$
- Let's say S_1 does not commute WLOG. If S_i also does not commute
  We can replace it by $S_i S_1$ (this now commutes with Z_1: $Z_1 (S_i S_1) =$
  $= -S_i Z_1 S_1 =$
  $= +(S_i S_1) Z_1$)

If $\{S_1, \dots, S_i, \dots, S_n\}$ is a set of generators.
$\Rightarrow \{S_1, \dots, S_i S_1, \dots, S_n\}$ is a set of generators.

- $P(+1) = \mathrm{tr}\left(|\psi\rangle\langle\psi| \left(\frac{\mathbb{1} + Z_1}{2}\right)\right) = \frac{1}{2} + \frac{1}{2}\langle\psi|Z_1|\psi\rangle =$

$$= \frac{1}{2} + \frac{1}{2} \cdot 0 = \frac{1}{2}$$

$$\langle\psi|Z_1|\psi\rangle = \langle\psi|Z_1 S_1|\psi\rangle$$
$$= -\langle\psi|S_1 Z_1|\psi\rangle$$
$$= -\langle\psi|Z_1|\psi\rangle$$

- $S_1^\dagger = S_1$
- $\langle\psi|S_1^\dagger = \langle\psi|$

- $P(0) = \frac{1}{2}$

- We flip a coin and simulate the measurement outcome.

- The post measurement state is $|0\rangle\langle 0| \otimes \mathbb{1}\, |\psi\rangle$ (if outcome was $+1$)

or $|1\rangle\langle 1| \otimes \mathbb{1}\, |\psi\rangle$ (if outcome was -1).

POST-MEASURE STABILIZERS:

$\langle Z_1, S_2, \dots, S_n\rangle$

$\langle -Z_1, S_2, \dots, S_n\rangle$

Freie Universität Berlin
Tutorials on Quantum Information Theory
Winter term 2022/23

Problem Sheet 9
Entanglement Witnesses and Cryptography

J. Eisert, A. Townsend-Teague, A. Mele, A. Burchards, J. Denzler

1. **Quantum Fourier transform.** (9 points: 1+4+2+2) Perhaps at the heart of the majority of modern quantum algorithms lies the *phase estimation algorithm.* For this reason, it is crucial in the field of quantum computation to be familiar with phase estimation. It relies on an efficient implementation of the *quantum Fourier transform*, to which we devote this excercise.

   In classical numerics the discrete Fourier transform (DFT) is defined as the linear map $F : \mathbb{C}^N \to \mathbb{C}^N$, $x \mapsto y$ with $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left\{\frac{2\pi \mathrm{i} jk}{N}\right\}$. The quantum Fourier transform is analogously defined as the unitary operation $\mathcal{F} : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, $|j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left\{\frac{2\pi \mathrm{i} jk}{2^n}\right\} |k\rangle$. (Note the identification $N = 2^n$.)

   a) Look-up the computational complexity of the fastest classical algorithm for the Fourier transform.

   The quantum Fourier transform can be implemented using the Hadamard gates H,

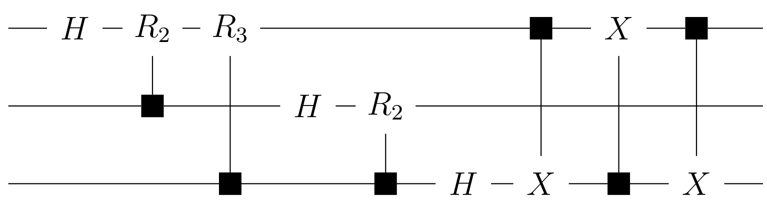
   $$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1}$$

   the controlled phase gate that applies

   $$R_k = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{2\pi \mathrm{i}/2^k} \end{pmatrix} \tag{2}$$

   on a *target* qubit if a *control* qubit is in the state $|1\rangle$ (and the identity if the control is in $|0\rangle$) and CNOT gates. Note that in circuit diagrams controlled gates are conventionally represented by boxes on the target wires linked to dots on the control wires.

   b) Show that the following circuit implements the three qubit quantum Fourier transform

   Hint: restrict your attention to generic computational basis states as inputs.

   c) How does this generalise to the n qubit quantum Fourier transform?

   d) What is the circuit complexity of the quantum Fourier transform and how does it compare to the classical DFT algorithms?

   Note that the quantum Fourier transform can in fact be approximately implemented with only $\mathcal{O}(n \log n)$ gates [1].

[1] Cleve, Richard, and John Watrous. "Fast parallel circuits for the quantum Fourier transform." Proceedings 41st Annual Symposium on Foundations of Computer Science. IEEE, 2000.

2. **Stabilizer quantum computation.** (11 Points: 3+2+1+2+1+1+1)

One of the most celebrated results in quantum computation is a statement about the resource costs of simulating quantum computations on a classical computers. The *Gottesman-Knill theorem* states that quantum computations composed of *Clifford gates* with *stabilizer states* as inputs and a final measurement in the computational basis can be classically simulated in the sense that there exists a classical algorithm with polynomial runtime which can sample from the output distribution of such a computation. Furthermore, the so-called stabilizer formalism plays an important rôle in the development of quantum error correction.

In this problem we will trace the reasoning underlying this result. Throughout, we will let n be the number of qubits and hence $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ be the Hilbert space. Let us start with some definitions

(i) Let $G_1 = \{\pm\mathbb{1}, \pm X, \pm Y, \pm Z, \pm\mathbb{1}, \pm \mathrm{i}X, \pm\mathrm{i}Y, \pm\mathrm{i}Z\}$ be the single-qubit *Pauli group* where multiplication is the group operation[2].

(ii) Let $G_n := \{\bigotimes_{i=1}^n P_i, P_i \in G_1\}$ be the n-qubit Pauli group.

(iii) A *stabilizer state* is a quantum state $|\psi\rangle \in \mathcal{H}$ that is uniquely (up to a global phase) described by a set $\mathcal{S}_{|\psi\rangle} = \{S_1, \dots, S_n\} \subset G_n$ satisfying $S_i|\psi\rangle = +1|\psi\rangle$. We call the generalised pauli-operators S_i the stabilizers of $|\psi\rangle$.[3] We note that stabilizers are linearly independent and commutative with each others.

(iv) A Clifford operator C is a unitary on $\mathcal{H}$ which leaves G_n invariant, i.e. for all $g \in G_n$ it holds that $CgC^\dagger \in G_n$. In group theoretic slang the Clifford group $\mathcal{C} \subset U(2^n)$ is the normalizer of G_n.

Ok, now we are ready to begin.

a) Show that the set $\mathcal{S} = \{Z_1, Z_2, \dots, Z_n\}$ uniquely stabilizes the state $|0\rangle^{\otimes n}$, where we use the notation $Z_i = \mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes \underbrace{Z}_{i\text{-th qubit}} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1}$ for the operator acting as Z on the i-th qubit and as the identity on all other qubits.

b) Show that n stabilizers suffice to uniquely characterize an arbitrary state in the *Clifford orbit* of $|0\rangle^{\otimes n}$, that is the states $|\psi\rangle$ for which there exists a (unique) Clifford operator C such that $|\psi\rangle = C|0\rangle^{\otimes n}$.

c) Give a stabilizer representation of $|+\rangle \otimes |0\rangle \otimes |-\rangle$.

Any Clifford operator can be expressed as a product of single- and two-qubit Clifford operators, and indeed as a product from the generating set $\{CNOT, H, S\}$, where

$$S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3}$$

d) Show that this gate set is sufficient to generate all Pauli matrices starting from any single-qubit Pauli matrix.

e) Argue that one can efficiently (in the number of qubits and gates) determine the stabilizer set of a state generated by a (known) Clifford circuit (comprising $CNOT, H, S$ gates) applied to a stabilizer state.

From the above reasoning, we conclude that we can efficiently simulate the effect of a Clifford circuit applied to a stabilizer state by keeping track of the stabilizers.

Now, let us assume that we measure the first qubit in the Z basis.

[2] Convince yourself that G_1 is closed under multiplication and the unsigned Pauli matrices are not.

[3] More generally, we can talk about subspaces stabilized by a set $\mathcal{S} \subset G_n$. This is a key insight in the theory of error correction codes.

f) Assume Z_1 commutes with all stabilizers. What is the probability of obtaining outcome $+1$?

One can show that in case Z_1 does not commute with all stabilizers, one can find an alternative set of stabilizers such that it anti-commutes with one of them but commutes with all remaining ones.

g) Use the existence of such a stabilizer to show that the measurement outcome is uniformly random. What is the post-measurement state?

In fact, this generalizes to the measurement of an arbitrary Pauli operator $g \in G_n$. Therefore, we see that checking commutation with the stabilizers gives us a recipe for efficiently simulating samples resulting from computational basis measurements.