Exercise Set 2 : 01 March 2018
Calcul Quantique

**Exercise 1** *Matrix representations of classical gates*

(a) For the NOT gate we have $\mathrm{NOT}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ and $\mathrm{NOT}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$. Therefore,

$$\mathrm{NOT} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

For the CNOT gate using its defintion (see Chapter 2) :

$$\mathrm{CNOT}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \qquad \mathrm{CNOT}\begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},$$
$$\mathrm{CNOT}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}, \qquad \mathrm{CNOT}\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}.$$

Therefore

$$\mathrm{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}.$$

For the CCNOT gate we have 8 possible entries :

$$000 = \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}, \qquad 001 = \begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}, \qquad 010 = \begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix}, \qquad 011 = \begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{pmatrix},$$

$$100 = \begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix}, \qquad 101 = \begin{pmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{pmatrix}, \qquad 110 = \begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix}, \qquad 111 = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}.$$

From $\mathrm{CCNOT}(x, y, z) = (x, y, z \oplus xy)$ we get for the first 6 vectors CCNOT outputs the same vector (since $xy = 0$). For the last two :

$$\mathrm{CCNOT}\begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}, \qquad \mathrm{CCNOT}\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix}.$$

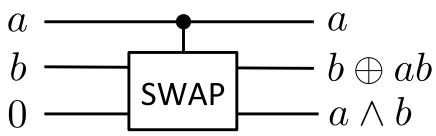
Thus

$$\mathrm{CCNOT} = \begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}.$$

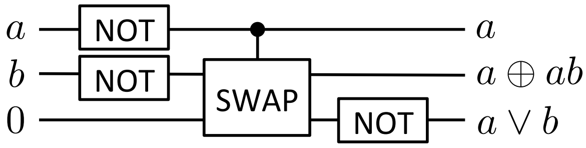
(b) NOT permutes the 2 basis vectors of $\mathbb{C}^2$. CNOT permutes the last 2 basis vectors of $\mathbb{C}^4$. CCNOT permutes the last 2 basis vectors of $\mathbb{C}^8$. Thus obviously $\mathrm{NOT}^2 = I$ so $\big(\mathrm{NOT}\big)^{-1} = \mathrm{NOT}$, $\mathrm{CNOT}^2 = I$ so $\big(\mathrm{CNOT}\big)^{-1} = \mathrm{CNOT}$ and $\mathrm{CCNOT}^2 = I$ so $\big(\mathrm{CCNOT}\big)^{-1} = \mathrm{CCNOT}$. Note also that these matrices are unitary, *i.e.*, $U^{-1} = U^\dagger$ for $U = \mathrm{NOT}, \mathrm{CNOT}, \mathrm{CCNOT}$.

**Exercise 2** *Fredkin gate*

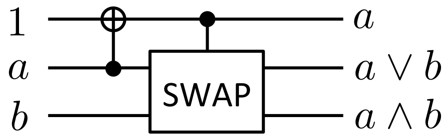
(a) The AND gate can be represented as follows with only the Fredkin gate :



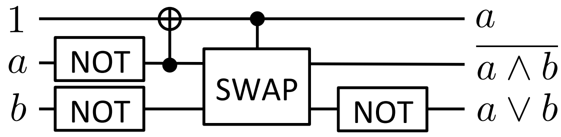

The OR gate is then (using $a \vee b = \mathrm{NOT}\big(\mathrm{NOT}(a) \wedge \mathrm{NOT}(b)\big)$)




Another solution for both AND and OR uses a combination of CSWAP and CNOT :



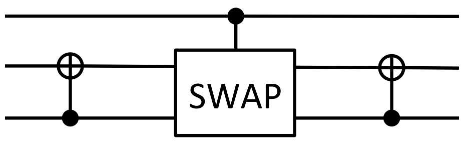

For the OR gate, alternatively, we then have :




(b) The Fredkin is a controlled SWAP which swap's the last two bits if the first one is equal to 1. Thus we find

$$\text{CSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

(c) From the matrix representation of Fredkin, we see that to obtain the matrix representation of CCNOT, we have to permute on rows 5,6,7,8. With a bit of thought one can find that the CCNOT gate can be represented as




Another way is by nothing that

$$\begin{aligned} \text{CNOT}(x,y) &= (x, x \oplus y), \\ \text{CCNOT}(x,y,z) &= (x, y, z \oplus xy), \\ \text{CSWAP}(x,y,z) &= (x, y \oplus x(y \oplus z), z \oplus x(y \oplus z)). \end{aligned}$$

Thus an input $(x, y, z)$ becomes $(x, y \oplus z, z)$ after the first CNOT gate, $(x, y \oplus z \oplus xy, z \oplus xy)$ after the Fredkin gate and $(x, y, z \oplus xy)$ after the second CNOT gate.

**Exercise 3** *Billiard Ball Model of a classical computation (cultural aside)*

Discussed in exercise session. The billiard ball model of computation shows that it is possible to compute any Boolean function with elastic collisions between balls in billiards. This is a dissipationless computation which moreover conserves the number of balls (mass). Note that only collisions between pairs of balls are needed (or triple collision needed). This is in contrast to CCNOT and Fredkin which use 3 bits. But note that Fredkin conserves the "number of one's".

■