CSE491/596 Lecture Friday, 11/17/23: Multi-Qubit Matrices and Gates

An n-qubit quantum state is denoted by a unit vector in $\mathbb{C}^N$ where $N = 2^n$. Thus, a 2-qubit state is represented by a unit vector in $\mathbb{C}^4$. That takes up 8 real dimensions, and even trying tricks as for the Bloch Sphere would bring that down only to a 6-dimensional hypersurface in $\mathbb{R}^7$. Until we have a Hyper-Zoom able to help us visualize 7-dimensional space, we have to rely on linear algebra and some general ideas shared by Hilbert Spaces whether real or complex.

One of those ideas is the **standard basis**. In 4-space, this is given by the vectors:

$$e_0 = (1,0,0,0),\ e_1 = (0,1,0,0),\ e_2 = (0,0,1,0),\ e_3 = (0,0,0,1).$$

The indexing scheme for **quantum coordinates** changes the labels to come from $\{0,1\}^2$ instead of from $\{1,2,3,4\}$, using the canonical binary order $00, 01, 10, 11$. Then we have:

$$e_{00} = (1,0,0,0),\ e_{01} = (0,1,0,0),\ e_{10} = (0,0,1,0),\ e_{11} = (0,0,0,1).$$

The big advantage is that these basis elements are all separable and the labels respect the tensor products involved:

$$\begin{aligned}
|00\rangle &= e_{00} = (1,0,0,0) = (1,0)\otimes(1,0) = e_0 \otimes e_0 = |0\rangle\otimes|0\rangle = |0\rangle|0\rangle \\
|01\rangle &= e_{01} = (0,1,0,0) = (1,0)\otimes(0,1) = e_0 \otimes e_1 = |0\rangle\otimes|1\rangle = |0\rangle|1\rangle \\
|10\rangle &= e_{10} = (0,0,1,0) = (0,1)\otimes(1,0) = e_1 \otimes e_0 = |1\rangle\otimes|0\rangle = |1\rangle|0\rangle \\
|11\rangle &= e_{11} = (0,0,0,1) = (0,1)\otimes(0,1) = e_1 \otimes e_1 = |1\rangle\otimes|1\rangle = |1\rangle|1\rangle
\end{aligned}$$

It is OK to picture the tensoring with row vectors, but because humanity chose to write matrix-vector products as Mv rather than vM, they need to be treated as column vectors. This will lead to cognitive dissonance when we read quantum circuits left-to-right but have to compose matrices right-to-left. Lipton and I are curious whether a "non-handed" description of nature can work.

With the "plus" and "minus" states, we also have (note $\frac{1}{\sqrt{2}}(1,1)\otimes\frac{1}{\sqrt{2}}(1,1) = \frac{1}{2}(1,1)\otimes(1,1)$):

$$\begin{aligned}
|{+}{+}\rangle &= |+\rangle\otimes|+\rangle = \tfrac{1}{2}(1,\ 1)\otimes(1,\ 1) = \tfrac{1}{2}(1,\ 1,\ 1,\ 1) = \frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2} \\
|{+}{-}\rangle &= |+\rangle\otimes|-\rangle = \tfrac{1}{2}(1,\ 1)\otimes(1,-1) = \tfrac{1}{2}(1,-1,\ 1,-1) = \frac{|00\rangle-|01\rangle+|10\rangle-|11\rangle}{2} \\
|{-}{+}\rangle &= |-\rangle\otimes|+\rangle = \tfrac{1}{2}(1,-1)\otimes(1,\ 1) = \tfrac{1}{2}(1,\ 1,-1,-1) = \frac{|00\rangle+|01\rangle-|10\rangle-|11\rangle}{2} \\
|{-}{-}\rangle &= |-\rangle\otimes|-\rangle = \tfrac{1}{2}(1,-1)\otimes(1,-1) = \tfrac{1}{2}(1,-1,-1,\ 1) = \frac{|00\rangle-|01\rangle-|10\rangle+|11\rangle}{2}
\end{aligned}$$

These four vectors are linearly independent and mutually orthogonal, so they form an orthonormal basis. We can map the standard 4-dimensional basis to this one by forming the target vectors into a

matrix---happily the matrix is symmetric and real so "handedness" does not come into play:

$$\frac{1}{2}\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right] = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \mathbf{H} \otimes \mathbf{H} = \mathbf{H}^{\otimes 2}.$$

Well, this is the case $m = 2$ of the Hadamard transform $\mathbf{H}^{\otimes m}$, about which more on Monday. Also note the following tensor products of 2×2 matrices:

$$\mathbf{H} \otimes \mathbf{I} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{array}\right],$$

$$\mathbf{I} \otimes \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right].$$

Some examples of states you can produce with these matrices are:

$$|{+}0\rangle = |+\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(1,1) \otimes (1,0) = \frac{1}{\sqrt{2}}(1,0,1,0) = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

$$|0+\rangle = |0\rangle \otimes |+\rangle = \frac{1}{\sqrt{2}}(1,0) \otimes (1,1) = \frac{1}{\sqrt{2}}(1,1,0,0) = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$

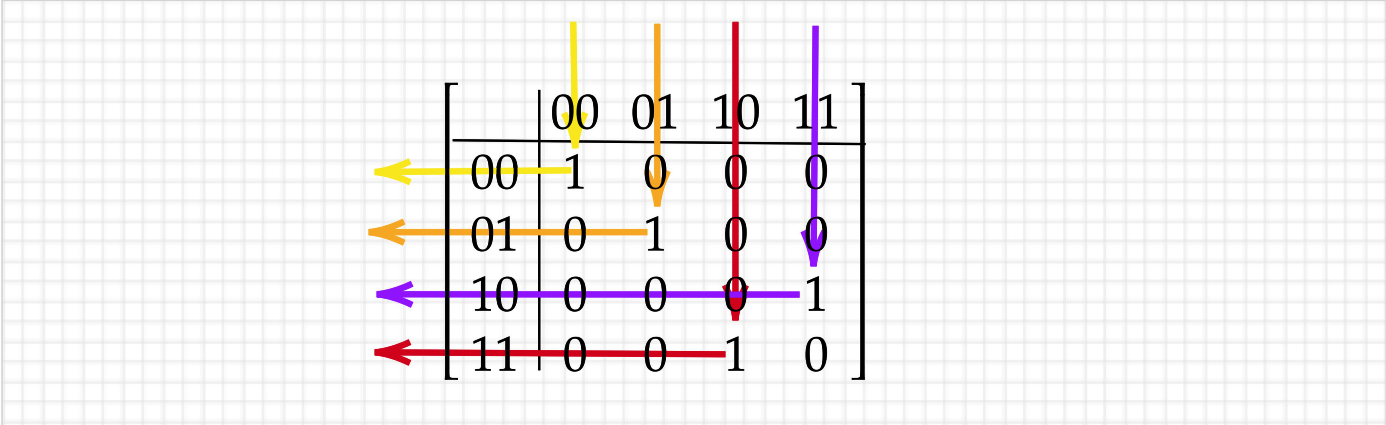
Meanwhile,

$$|{+}1\rangle = |+\rangle \otimes |1\rangle = \frac{1}{\sqrt{2}}(1,1) \otimes (0,1) = \frac{1}{\sqrt{2}}(0,1,0,1) = \frac{|01\rangle + |11\rangle}{\sqrt{2}}$$

can be gotten as $\mathbf{H} \otimes \mathbf{I}$ applied to the column vector $(0,1,0,0)^T = |01\rangle$. However, the state $\frac{1}{\sqrt{2}}(1,0,0,1) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$, which we saw in the last lecture is entangled, cannot be gotten this way. Instead, it needs the help of a 4×4 unitary matrix that is not a tensor product of two smaller matrices. The most omnipresent one of these is:

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Any linear operator is uniquely defined by its values on a particular basis, and on the standard basis, the values are: $\mathbf{CNOT}e_{00} = \mathbf{CNOT}|00\rangle = |00\rangle$, $\mathbf{CNOT}e_{01} = \mathbf{CNOT}|01\rangle = |01\rangle$, $\mathbf{CNOT}e_{10} = \mathbf{CNOT}|10\rangle = |11\rangle$, and $\mathbf{CNOT}e_{11} = \mathbf{CNOT}|11\rangle = |10\rangle$. We can get these from the respective columns of the $\mathbf{CNOT}$ matrix, and we can label the quantum coordinates right on it:



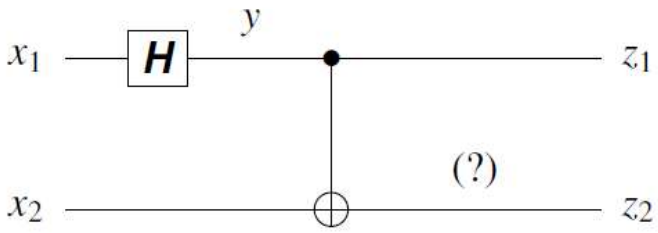
Because we multiply column vectors, the co-ordinates of the argument vector come in the top and go out to the left. If the first qubit is 0, then the whole gate acts as the identity. But if the first qubit is 1, then the basis value of the second qubit gets flipped---the same action as the **NOT** gate $\mathbf{X}$. Hence the name Controlled-NOT, abbreviated $\mathbf{CNOT}$: the **NOT** action is controlled by the first qubit. The action on a general 2-qubit quantum state $\phi = (a, b, c, d)$ is even easier to picture:

$$\mathbf{CNOT}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ d \\ c \end{pmatrix}.$$

All it does is switch the third and fourth components---of any 4-dim. state vector. Hence, $\mathbf{CNOT}$ is a **permutation gate** and is entirely deterministic. Permuting these two indices is exactly what we need to transform the separable state $\frac{1}{\sqrt{2}}(1, 0, 1, 0)$ into the entangled state $\frac{1}{\sqrt{2}}(1, 0, 0, 1)$. Since we got the former state from $\mathbf{H} \otimes \mathbf{I}$ applied to $\mathbf{e_{00}}$, the matrix we want is

$$\mathbf{CNOT} \cdot (\mathbf{H} \otimes \mathbf{I}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left[\begin{array}{c|ccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{array}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We can see the result coming from the first column. When we do a quantum circuit left-to-right, however, the $(\mathbf{H} \otimes \mathbf{I})$ part comes first on the left. The symbol for a **CNOT** gate is to use a black dot to represent the control on the *source qubit* and $\oplus$ (which I have used as a symbol for XOR) on the *target qubit*. This is more easily pictured by a quantum circuit diagram:



If $x_1 = |0\rangle$, then we can tell exactly what y is: it is the $|+\rangle$ state. And if $x_1 = |1\rangle$, then $y = |-\rangle$. If x_1 is any separate qubit state $(a, b) = a|0\rangle + b|1\rangle$, then by linearity we know that $y = a|+\rangle + b|-\rangle$. This expresses y over the transformed basis; in the standard basis it is

$$\frac{1}{\sqrt{2}}(a(1,1) + b(1,-1)) = \frac{1}{\sqrt{2}}(a+b, a-b)\ .$$

So we can say exactly what the input coming in to the first "wire" of the CNOT gate is. And the input to the second wire is just whatever x_2 is. But because that gate does entanglement, we cannot specify individual values for the wires coming *out*. The state is an inseparable 2-qubit state:

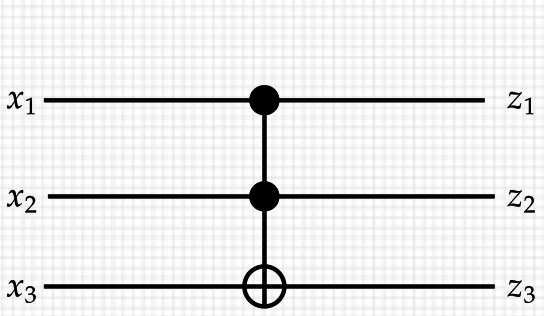
$$\frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big).$$

If you measure either qubit individually, you get 0 or 1 with equal probability. This is the same as if you measured the state $|{+}{+}\rangle$. But that state is outwardly as well as inwardly different. When *both* qubits to be measured, it allows 01 and 10 as possible outcomes, whereas measuring the entangled state does not. I've seen papers telling ways to visualize entangled states of 2 or 3 qubits, but none implemented by an applet so far---`quantum-circuit.com` just shows Bloch spheres with the black dot at the center for the "completely mixed state": $|\ ¯\_(ツ)_/¯\ \rangle$.

Three Qubits and More

The **CNOT** gate by itself has the logical description $z_1 = x_1$ and $z_2 = x_1 \oplus x_2$. This means that if $x_1 = 0$ then $z_2 = x_2$, but if $x_1 = 1$ then $z_2 = \neg x_2$. Since this description is complete for all of the standard basis inputs $x = x_1x_2 = 00, 01, 10, 11$, it extends by linearity to all quantum states. We can use this idea to specify the 3-qubit **Toffoli gate** (**Tof**). It has inputs x_1, x_2, x_3 and symbolic outputs z_1, z_2, z_3 (which, however, might not have individual values in non-basis cases owing to entanglement). Its spec in the basis quantum coordinates is:

$$z_1 = x_1,\ z_2 = x_2, z_3 = x_3 \oplus (x_1 \wedge x_2).$$



	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	1	0

Of particular note is that if x_3 is fixed to be a constant-1 input, then

$$z_3 \;=\; \neg(x_1 \wedge x_2) \;=\; NAND(x_1, x_2).$$

Thus the Toffoli gate subsumes a classical NAND gate, except that you need an extra "helper wire" to put $x_3 \;=\; 1$ and you gate two extra output wires z_1, z_2 that only compute the identity on x_1, x_2 (in classical logic, that is---a non-basis quantum state can have knock-on effects even though all Toffoli does is switch the 7th and 8th components of the state vectors). If you have polynomially many Toffoli gates, then you get only polynomially much wastage of wires, and you can use the good ones to simulate any polynomial-size Boolean circuit of NAND gates. Because DTIME$[t(n)]$ has Boolean circuits of size $\widetilde{O}(t(n))$, and because Toffoli gates are deterministic, we can state an immediate consequence:

Theorem: For fully time-constructible $t(n)$ between linear and exponential.

$$\text{DTIME}[t(n)] \;\subseteq\; \text{DQ}\left[\widetilde{O}(t(n))\right].$$

In particular, P $\subseteq$ DQP $\subseteq$ BQP.

Well, we need to say more broadly what it means for quantum computations to be (polynomially) **feasible.** The community convention is simply to count up gates of 1, 2, or 3 qubits as constant cost. Gates involving more qubits are OK if they can be built up out of the small gates. We have already seen that $\mathbf{H}^{\otimes n}$ is just n binary Hadamard gates laid out in parallel. The n-qubit **quantum Fourier transform** (next week) can be built up out of $O(n^2)$ smaller gates---this actually has more "fine print" than sources usually say and is pursued in the chapter exercises of my book with Lipton.

And BQP is to DQP as BPP is to P. We should describe measurements in more detail and see smaller-scale deterministic and randomized examples first.