CS 410/510 Introduction to Quantum Computing

Lecture 2

Portland State U, Spring 2017 | April 6, 2017
Lecturer: Fang Song | *Scribe: Henry Cooney*

VERSION: APRIL 13, 2017

1 The tensor product

Recall that we may represent bits as probability vectors. For example, suppose we have u, v and:

$$u = \begin{bmatrix} 3/4 \\ 1/4 \end{bmatrix} \tag{1}$$

$$v = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} \tag{2}$$

Then the *tensor product* of u and v, denoted $u \otimes v$, is the vector:

$$u \otimes v = \begin{bmatrix} 1/4 \\ 1/2 \\ 1/12 \\ 1/6 \end{bmatrix} \tag{3}$$

$$\tag{4}$$

The tensor product is used to represent the *joint probability* of multiple bits in a single vector. Each row simply corresponds to the probability of that bit combination: for example the probability that u is 1, and v is 0, is the element $(u \otimes v)_{10}$, which is $1/12$.

1.1 Computing the Tensor Product

Suppose we have an $m \times n$ matrix A, and a $k \times l$ matrix B. So:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{l1} & \dots & b_{lk} \end{bmatrix} \tag{6}$$

$$\tag{7}$$

The tensor product is computed by multiplying each element of A with the entire matrix B, and tiling the the result. So:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{8}$$

$$\tag{9}$$

Thus, the dimension of $A \otimes B$ is $mk \times nl$.

So, for example, the tensor product of u and v, used above, is calculated as follows:

$$u \otimes v = \begin{bmatrix} 3/4 \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} \\ 1/4 \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} 1/4 \\ 1/2 \\ 1/12 \\ 1/6 \end{bmatrix} \tag{11}$$

1.2 Properties of the Tensor Product

A few tensor product properties:

1. Distributive: $A \otimes (B + C) = A \otimes B + A \otimes C$
2. You can move a scalar around: $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \oplus B)$
3. Preserves unitary-ness: if u and v are unitary, $u \otimes v$ is unitary.
4. NOT commutative!! It is not always true that $A \otimes B = B \otimes A$.

Another interesting property is that there are some vectors that cannot be written as a tensor product. For example, for the following vector x:

$$x = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix} \tag{12}$$

There is no pair of vectors u and v such that $u \otimes v = x$. In this case, we say that u and v are *correlated*. Only uncorrelated systems may be written as a tensor product. This will be an important property later.

1.3 Tensor produces with qubits

We can use similarly use tensor products to describe associated probabilities on qubits. Suppose X and Y are qubits such that:

$$X : |\phi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{13}$$

$$Y : |\psi\rangle = \gamma |0\rangle + \delta |1\rangle = \begin{bmatrix} \gamma \\ \delta \end{bmatrix} \tag{14}$$

Note: this is using Dirac notation, covered in Lecture 1 – or p. 10 of the Watrous notes (link on main course page).

So, what is the state of the joint system? We can compute the tensor product to find out:

$$X : |\phi\rangle \otimes |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \otimes \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix} \tag{15}$$

$$\tag{16}$$

This can equivalently be written as:

$$|\phi\rangle \otimes |\psi\rangle = (\alpha |0\rangle + \beta |1\rangle) \otimes (\gamma |0\rangle + \delta |1\rangle) \tag{17}$$

$$= \alpha\gamma(|0\rangle \otimes |0\rangle) + \alpha\delta(|0\rangle \otimes |1\rangle) + \beta\gamma(|1\rangle \otimes |0\rangle) + \beta\delta(|1\rangle \otimes |1\rangle) \tag{18}$$

2 Multiple Qubits and Dirac Notation

Above, we ended up with the long expression, $\alpha\gamma(|0\rangle \otimes |0\rangle) + \alpha\delta(|0\rangle \otimes |1\rangle) + \beta\gamma(|1\rangle \otimes |0\rangle) + \beta\delta(|1\rangle \otimes |1\rangle)$. Why write it this way? As it turns out, vectors like $|0\rangle \otimes |1\rangle$ form an orthonormal basis. There is a special notation for these vectors – for example, $|00\rangle = |0\rangle \otimes |0\rangle$. We can also denote the tensor product of any two states ϕ and ψ as follows: $\phi \otimes \psi = |\phi\rangle |\psi\rangle$.

So, our four basic states that can result from $|0\rangle$ and $|1\rangle$ (and thus the orthonormal basis for 2-bit systems) are:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{19}$$

$$|01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

$$|10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{21}$$

$$|11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{22}$$

These allow us to write certain superpositions more compactly using Dirac notation. For example:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \tag{23}$$

Note that the superposition above is one of the special vectors which *cannot be written as a tensor product*. A superposition which cannot be written as a tensor product is called *entangled*. A superposition that can be written as a tensor product is called a *product state*. Entangled superpositions will be important in the future. An pair of entangled qubits is also called an *EPR pair*.

As mentioned above, $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ form an orthonormal basis for $\mathbb{C}^4$. More bits would mean an orthonormal basis for a higher-dimensional space. For example, suppose we have n qubits. Then:

$$\{|0...0\rangle \; ... \; |1...1\rangle\} \tag{24}$$

Is the basis for $(\mathbb{C}^2)^n$, with dimensions 2^n.

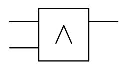
3 Circuit Models

3.1 Classical Circuit Models

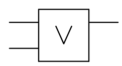
Circuit models represent computation as a network of logical gates. This model is equivalent in power to a Turing machine.

Some basic gates are and, or, and not:

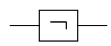
And:



Or:

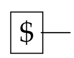


Not:

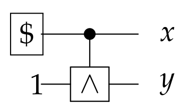


These make a very powerful combination. A circuit with s gates maybe converted to a Turing machine that runs in $O(s \log s)$ time. Similarly, a Turing machine that that runs in time t can be converted to a circuit with $O(t \log t)$ gates. One thing which classical circuits often lack is *randomness*. Therefore, we also introduce the 'coin flip' gate, which randomly outputs 0 or 1:

Coin flip:



We can use this to make a more interesting circuit:



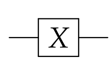
This circuit outputs either 11 or 00, each with probability 1/2.

3.2 Quantum Circuits

Quantum gates are different from classical circuit gates. Instead, they correspond to the unitary Pauli operators:

The X operator:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{25}$$



The Y operator:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{26}$$

$$-\boxed{Y}-$$

The Z operator:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{27}$$

$$-\boxed{Z}-$$

The Identity operator:

$$-\boxed{I}-$$

Finally, the Hadamard Operator:

$$Z = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \tag{28}$$

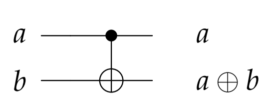
We can also create 2-qubit gates by taking two 1-qubit gates and applying the tensor product to their matrices. Not that, as per the rules of the tensor product, this will yield a 4x4 matrix.

3.3 More Interesting Gates

A more interesting gate is the 'controlled not' gate, or CNOT. It has the following matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{29}$$

CNOT takes as input a 'control' qubit and a 'target' qubit. The control qubit will pass through unchanged, but the target will be xored with the control. So, if the control is a and the target is b, CNOT will output a and $a \oplus b$. CNOT has a special circuit diagram, shown below:

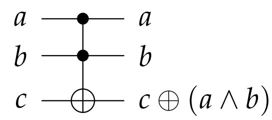


For the basic 2-bit states, the output of CNOT looks like this:

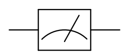
$$CNOT\,|00\rangle = |00\rangle \tag{30}$$
$$CNOT\,|01\rangle = |01\rangle \tag{31}$$
$$CNOT\,|10\rangle = |11\rangle \tag{32}$$
$$CNOT\,|11\rangle = |10\rangle \tag{33}$$

Note that, if the first bit is 1, the second bit is flipped – as one would expect for an xor-like operation.

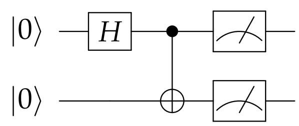
Another complex gate is the Tofolli gate. This takes 3 input qubits, a, b, and c, and changes c to $c \oplus (a \wedge b)$. You can look up the matrix, but the Toffoli gate circuit looks like this:



A final unique gate type is the Measure gate. This represents actually measuring the qubit, collapsing it to a classical value. It's represented by a meter symbol:



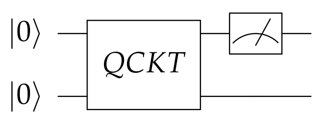
3.4 An Example Quantum Circuit



The possible outputs of this circuit are 00 and 11, each with probability 1/2. Note that, just before measurement, the state of the qubits is $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ – the entangled state that was mentioned in equation (21). So, this circuit makes use of an EPR pair!

3.5 Partial measurement

Suppose we have the following quantum circuit:



Here, QCKT is some arbitrary quantum circuit. Suppose we want to know the state of the qubits just before the measurement is taken, which will will call $|\phi\rangle$. Then we could write this as:

$$|\phi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \tag{34}$$

So, the probability of seeing a 0 in the first bit is:

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 \tag{35}$$

And the probability of seeing 1 in the first bit is:

$$|\alpha_{10}|^2 + |\alpha_{11}|^2 \tag{36}$$

However, this is BEFORE measurement. After measurement, the probability on the other bit will change – measuring the first bit gives you additional information about the second. Suppose you measure 0 for bit 1. Then bit 1's state is now:

$$\frac{\alpha_{00} |0\rangle + \alpha_{01} |1\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \tag{37}$$

And, if 1 is measured for bit 1, bit 2's state is:

$$\frac{\alpha_{10} |0\rangle + \alpha_{11} |1\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}} \tag{38}$$

Note that we must divide by the square root term to normalize the probabilities, so they still add up to one. A final note – the order in which measurements are performed does NOT affect the output probabilities. You may measure in any order and still get the same result.