

Quantum Computing

Jie Wang
Dept of Computer Science
University of Massachusetts, Lowell
wang@cs.uml.edu

Abstract

This note is written for 91.502 Foundations of Computer Science Class in Spring 2003. It aims to provide a quick introduction to the idea of quantum computing.

1 Introduction

We view computation as a sequence of operations that manipulate an input to produce an output, and the operations can be programmed and carried out on a man-made device. Electronic computers that follow simple physical laws of logic operations are the most common computing devices to carry out computation. Other means of performing computation, such as molecular computers, have also been experimented. Quantum computing models that follow physical laws of quantum mechanics are possible, at least in the form of a mathematical theory. In the present note I will present basic ideas of quantum computing.

In electronic computing we deal with binary **bits**. Each binary bit has a value 0 or 1. Logic gates and electronic circuits such as VLSI have been built that can manipulate binary bits accurately at fast speed. In quantum computing we deal with quantum bits, called **qubits**. Each qubit is in a quantum state, which can be $|0\rangle$ or $|1\rangle$ or any superposition of the form of

$$\alpha|0\rangle + \beta|1\rangle,$$

where $|0\rangle$ corresponds to the unit vector $(1, 0)$, $|1\rangle$ corresponds to the unit vector $(0, 1)$, and α and β are complex numbers, called the amplitudes, satisfying $\|\alpha\|^2 + \|\beta\|^2 = 1$. The symbol $|\cdot\rangle$ is pronounced “ket”. Let $\alpha = a + ib$. Denote by α^* the complex conjugate of α ; namely, $\alpha^* = a - ib$. Then $\|\alpha\| = \sqrt{\alpha^*\alpha}$, which is called the norm of α . Thus, $\|\alpha\|^2 = a^2 + b^2$. A qubit captures the quantum behavior of a single photon.

Manipulations of a sequence of qubits can be carried out as mathematical operations following the laws of quantum mechanics. While at the present time it is still unclear whether one can eventually build a physical device that can manipulate and measure qubits accurately, preliminary experiments have offered certain hopes.

We will discuss quantum computing as mathematical operations in finite dimensional Hilbert space. We note that $|0\rangle$ and $|1\rangle$ constitute an orthogonal basis to Hilbert space C^2 : a 2-dimensional vector space of complex numbers with the standard inner product. Let $v_1 = (\alpha_1, \beta_1)$ and (α_2, β_2) be two vectors in C^2 , then the inner product of the two vectors is

$$\langle v_1 | v_2 \rangle = \alpha_1^* \alpha_2 + \beta_1^* \beta_2.$$

Let $v = (\alpha, \beta) \in C^2$, then $\langle v | v \rangle = \|\alpha\|^2 + \|\beta\|^2$.

We use a sequence of n qubits to represent an arbitrary input of size n . This qubit “string”, called a qubit register, is an element in the tensor product (also called the tensor direct product) of n spaces: $C^2 \otimes C^2 \otimes \cdots \otimes C^2$. This is a Hilbert space spanned by the basis of 2^n vectors:

$$\begin{aligned} &|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle \\ &|0\rangle \otimes |0\rangle \otimes \cdots \otimes |1\rangle \\ &\quad \dots \\ &|1\rangle \otimes |1\rangle \otimes \cdots \otimes |1\rangle \end{aligned}$$

Denote $|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$ by $|x_1 x_2 \cdots x_n\rangle$, where $x_j \in \{0, 1\}$, and $j = 1, \dots, n$.

The tensor product of two matrices in matrix representations is identical to the Kronecker product of matrices. Namely, let $A = \{a_{ij}\}_{m \times n}$ and $B = \{b_{ij}\}_{p \times q}$ be two matrices of complex numbers, then

$$A \otimes B = \begin{pmatrix} a_{11}^* B & a_{12}^* B & \cdots & a_{1n}^* B \\ a_{21}^* B & a_{22}^* B & \cdots & a_{2n}^* B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}^* B & a_{m2}^* B & \cdots & a_{mn}^* B \end{pmatrix}.$$

Thus, $|0\rangle \otimes |1\rangle = (1, 0) \otimes (0, 1) = (0, 1, 0, 0)$, and $|0\rangle \otimes |1\rangle \otimes |0\rangle = (0, 1, 0, 0) \otimes (1, 0) = (0, 0, 1, 0, 0, 0, 0, 0)$.

A binary string $x = x_1 x_2 \cdots x_n$ corresponds to quantum state $|x\rangle \equiv |x_1 x_2 \cdots x_n\rangle$. Computing a function $f : x_1 x_2 \cdots x_n \mapsto f(x_1 x_2 \cdots x_n)$ in a quantum computer is equivalent to performing transformation

$$|x_1 x_2 \cdots x_n\rangle \mapsto U |x_1 x_2 \cdots x_n\rangle = |f(x_1 x_2 \cdots x_n)\rangle,$$

with U being a $2^n \times 2^n$ unitary matrix, meaning that $UU^\dagger = I$, where U^\dagger is the transpose of the complex conjugate of U .

According to quantum mechanics, at any given moment a quantum particle (such as one photon) is in a linear superposition $\alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying $\|\alpha\|^2 + \|\beta\|^2 = 1$. Likewise, the superposition of n photons at any given moment is in the superposition of 2^n basis states:

$$\sum_{i=0}^{2^n-1} c_i |i\rangle,$$

where c_i are complex numbers and $\sum_{i=0}^{2^n-1} \|c_i\|^2 = 1$. A quantum state of n qubits is a superposition, which describes quantum correlations of these photons. Thus, to extract information (such as probing the memory of an electronic computer) we can project the superposition onto an orthonormal basis element so that only one of the 2^n many binary strings is to be observed. This is called an observation, or a measurement, of the n -qubit quantum register. For example, when a qubit $\alpha|0\rangle + \beta|1\rangle$ is measured (observed), we will get $|0\rangle$ with probability $\|\alpha\|^2$, and $|1\rangle$ with probability $\|\beta\|^2$. We note that this is just one simple kind of possible measurement. Other allowable measurements are possible.

After the system is measured, the exponential amount of information of the system is completely lost. To make the system useful for computing one needs to use the interference feature of quantum mechanics to cleverly cancel other computations so that only the one we are interested in remains to be observed.

Here is what would happen in a quantum computer. The input x (a binary string) of length n is represented by an n -qubit register with initial value $|x\rangle$ in Hilbert space C^{2^n} . A sequence of elementary operations will then be performed on the register, starting on its initial value. Each elementary operation is a unitary operator, corresponding to one computation step in the computation. For example, the following operation is the negation operation of one qubit, denoted by *NOT*:

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

Clearly, U is unitary. It is easy to see that

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

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle.$$

It is also easy to verify that $NOT(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle$. Similarly, we can define \sqrt{NOT} to be the following unitary operation:

$$\sqrt{NOT} = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix}.$$

Clearly, $\sqrt{NOT}\sqrt{NOT} = NOT$.

As another example we note that the effect of the following unitary operation rotates a qubit in the 3D space:

$$G_{\theta,\phi} = \begin{pmatrix} \cos \theta & e^{i\phi} \sin \theta \\ -e^{-i\phi} \sin \theta & \cos \theta \end{pmatrix}.$$

To see why $G_{\theta,\phi}$ is unitary it suffices to note that $e^{i\phi} = \cos \phi + i \sin \phi$, $(e^{i\phi})^* = e^{-i\phi}$, and $\cos^2 \theta + \sin^2 \theta = 1$.

We use $\langle x|$, pronounced “bra”, to represent the conjugate transpose of $|x\rangle$. So if $|x\rangle = (x_1, x_2, \dots, x_n)^T \in C^n$, then $\langle x| = (x_1^*, x_2^*, \dots, x_n^*)$. It is straightforward to verify that $\langle x|y\rangle = \langle x| \cdot |y\rangle$, where \cdot represents the inner product of two vectors. Naturally we can define an outer product of $|x\rangle$ and $|y\rangle$ as $|x\rangle\langle y|$. Thus, $(|x\rangle\langle y|)|z\rangle = (\langle y|z\rangle)|x\rangle$.

2 Deutsch-Jozsa’s algorithm

In a probabilistic Turing machine computation a certain computation path is selected with a certain probability, and the other computation paths (i.e., those what-could-have-been-chosen-but-were-not-chosen paths) do not influence the actual outcome of the selected path. On the contrary, in a quantum computation, all potential computation paths are taken simultaneously in a single piece of hardware following the superposition principle of quantum mechanics, and all the possible paths that would yield the same result interfere each other. To design a quantum algorithm is therefore to devise a unitary matrix, a sequence of unitary matrices, to be used as operator(s) to change the input to the desired output.

We present in this section Deutsch-Jozsa’s algorithm as a concrete example of quantum algorithm. Suppose we are given 2^n binary bits, in which either all binary bits are 0 (“constant”), or half are 0 and half are 1 (“balanced”). We want to distinguish between these two cases as fast as we can. A deterministic algorithm requires to query $2^{n-1} + 1$ bits in order to always give a correct answer. However, if we are willing to accept an answer with small probability of error, we can find the correct answer much more quickly as follows. We randomly select a binary bit, query its value, and repeat this experiment k times. If the value 1 is observed, output “balanced”, otherwise, output “constant”. The probability of making a wrong answer when outputting

“constant” is equal to 2^{-k} . Deutsch and Jozsa (1992) showed that using a quantum computer one can obtain the correct answer with probability 1 in just one query!

We may rephrase the problem as follows. Assume that we are given a function $f : \{0, 1, \dots, 2^n - 1\} \rightarrow \{0, 1\}$, where either $f(i)$ are all equal to 0, or half are 0 and half are 1. We will use a quantum gate U_f (a unitary operation) operated on $n + 1$ qubits, which is controlled by f . In particular, for any $0 \leq i < 2^n$ and $j \in \{0, 1\}$,

$$U_f|i\rangle \otimes |j\rangle = |i\rangle \otimes |j \oplus f(i)\rangle,$$

where \oplus is the exclusive-or bit operation, namely, $a \oplus b = (a + b) \bmod 2$. This means that U_f converts $|j\rangle$ to its negation if $f(i) = 1$. Otherwise, $|j\rangle$ remains unchanged. Let H be the following unitary operation on one qubit, called the Hadamard transform:

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

Note that the inverse $H^{-1} = H$. We also note that

$$H|j\rangle = \frac{1}{\sqrt{2}} \sum_{\ell=0}^1 (-1)^{j \cdot \ell} |j\rangle.$$

We can generate H to form a $2^n \times 2^n$ unitary matrix operate on n qubits. This operation is the discrete Fourier transform T over group Z_2^n . Clearly $T^{-1} = T$.

We use two qubit registers, one consisting of n qubits and the other consisting of one qubit. Initially, these two qubit registers are set to $|0^n\rangle \otimes |1\rangle$. We then apply T on $|0^n\rangle$ in the first register and H on $|1\rangle$ in the second register, resulting in a new state:

$$\frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} |i\rangle \right) \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle).$$

Let $|\eta\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} |i\rangle \right)$ and $|j\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$. Make one query to U_f on $|\eta\rangle \otimes |j\rangle$ we get

$$\begin{aligned} U_f|\eta\rangle \otimes |j\rangle &= U_f \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left(|i\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} U_f|i\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \otimes \frac{1}{\sqrt{2}} ((|0\rangle - |1\rangle) \oplus f(i)) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \otimes (-1)^{f(i)} \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} |i\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle).
\end{aligned}$$

Applying the reversed Fourier transform T^{-1} on the n qubits $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} |i\rangle$ in the first register we get

$$|\xi\rangle \otimes \left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right).$$

Measure the first register. If the outcome is equal to 0^n , then output “constant”; otherwise, output “balanced”.

We now show that why this algorithm works. Denote by $|\xi_c\rangle$ the vector $|\xi\rangle$ in the case “constant” and $|\xi_b\rangle$ the vector $|\xi\rangle$ in the case “balanced”. If $f(i)$ is constant, then the second Fourier transform simply undoes the first Fourier transform, and so $|\xi_c\rangle = |0^n\rangle$. On the other hand, if $f(i)$ is balanced, then the vector

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} |i\rangle \text{ is orthogonal to } \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

We note that unitary operations preserve angles between vectors. Thus, $|\xi_b\rangle$ is orthogonal to $|\xi_c\rangle$. Hence, the probability of observing 0^n in the case “balanced” is 0. This means that the algorithm gives the correct answer with probability 1.

Acknowledgement

I thank Professors Jay Belanger, Paul Duvall, and Mark Yin for answering my questions on Hilbert space and tensor products, and Professor Steve Homer for his comments on the first draft of my note.