Lecture 5: Quantum Circuits and a Simple Quantum Algorithm

Scribed by: Dion Harmon

Department of Mathematics, MIT

September 18, 2003

1 Administrativa

- On homework 1, the book uses σ_1, σ_2, and σ_3 for σ_x, σ_y, and σ_z.
- We'll go over the homework next lecture.

2 Review from last lecture

2.1 Classical circuits [§3.1.2 pp 129–135]

Classical circuits take input bits (and possibly some work bits) on lines and are supposed to use a network of various logic gates to deliver output bits to other lines. Circuits can compute any function that a Turing machine can compute. It can be proved that if a Turing machine can compute a function G in $f(n)$ steps for input of length n there is a uniform family of circuits (see below) computing G such that the circuit with n input bits has $\leq cf(n)^2$ where c depends on the complexity of the Turing machine.

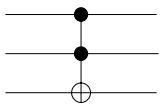
A uniform family of circuits is a set of circuits with one circuit for each number of input bits. Some Turing machine must be able to construct these circuits in the following manner: on input n, the machine outputs a description of the circuit for n input bits. These are the only sorts of circuits we can realistically work with.

2.2 Reversible circuits and computation [§3.2.5 pp 153–161]

Reversible circuits are circuits which do not lose information. This implies more than just keeping the input bits around during computation: information conservation is a local rather than global statement. Given the output from any gate in the reversible circuit, we must be able to uniquely determine the bits on the gate's input lines. This implies global conservation: given the output from the circuit, we can unambiguously determine the input. Reversible computations are defined

similarly (see textbook). A Toffoli gate is a controlled controlled flip:

$$\begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & & 1 \\ & & & & & & 1 & \end{pmatrix}$$

or  or

IN	OUT
000	000
001	001
010	010
011	011
100	100
101	101
110	111
111	110

This gate is universal: with ancillary input zeros, it can be used to compute any function.

3 Quantum programs and quantum circuits [Chapter 4, particularly §4.6 pp 202–204]

Quantum programs are usually specified in terms of circuits (though other models do exist). The model used in the book (and in class) is described in §4.6 of the textbook on page 202. The major points are summarized below:

- **Classical resources:** For efficiency and simplicity, we can use classical computations when we want (but not directly on the quantum state space). We show later in the lecture that this is not needed, but it can make describing problems much simpler.
- **Suitable state space:** The quantum portion of our state space is a 2^n-dimensional Hilbert space which is the tensor product of n quantum bits. Product states of the form $|x_0\rangle \otimes |x_1\rangle \otimes \cdots \otimes |x_n\rangle$ with $x_i \in \{0, 1\}$ are computational basis states of the computer.
- **Ability to prepare states in the computational basis:** We assume we can set our quantum state space to any state of the computational basis in n steps.
- **Ability to perform quantum gates:** We can apply quantum gates to any set of our qubits of fixed cardinality. Usually we say three or fewer qubits, but we could get away with only two. Note that all of the quantum gates "operate" on the entire state space, but
- **Ability to perform measurements in the computational basis:** We can measure the quantum state vector in computational basis.

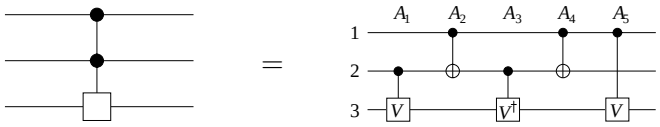
A uniform family of quantum circuits is a set of circuits $\{C_n\}$ such that a *classical* Turing machine can produce a description of C_n on input n in time polynomial in the length of n.

4 Arbitrary quantum gates from controlled gates (and not gates)

We now show that we can build up arbitrary quantum gates (on any number of bits) by using operators which operate on at most two bits at a time.

4.1 A doubly controlled gate from singly controlled gate [§4.3]

We wish to build a three qubit controlled controlled gate from some combination of singly controlled gates. The controlled controlled gate does nothing unless bits one and two are both one, when it applies the unitary transform U to bit three. We claim that



does the trick where $V^2 = U$. It is straight forward to check, but we can do the matrix multiplication for rigor. The matrix for the controlled controlled gate is

$$\begin{pmatrix} I & & & \\ & I & & \\ & & I & \\ & & & U \end{pmatrix}$$

where the identity matrices are all 2×2. The two-bit gates are labeled A_1 through A_5. Their actions are straight forward to determine, and their matrices can be read off by examination or a little scratch work. They are:

$$A_1 = \begin{pmatrix} I & & & \\ & V & & \\ & & I & \\ & & & V \end{pmatrix} \quad A_2 = \begin{pmatrix} I & & & \\ & I & & \\ & & I & \\ & & & I \end{pmatrix} = A_4$$

$$A_3 = \begin{pmatrix} I & & & \\ & V^\dagger & & \\ & & I & \\ & & & V^\dagger \end{pmatrix} \quad A_5 = \begin{pmatrix} I & & & \\ & I & & \\ & & V & \\ & & & V \end{pmatrix}$$

Carrying out the multiplications sequentially yields the following.

$$A_2A_1 = \begin{pmatrix} I & & & \\ & V & & \\ & & I & \\ & & & V \end{pmatrix} \qquad A_3A_2A_1 = \begin{pmatrix} I & & & \\ & I & & \\ & & V^\dagger & \\ & & & V \end{pmatrix}$$

$$A_4A_3A_2A_1 = \begin{pmatrix} I & & & \\ & I & & \\ & & V^\dagger & \\ & & & V \end{pmatrix} \quad A_5A_4A_3A_2A_1 = \begin{pmatrix} I & & & \\ & I & & \\ & & I & \\ & & & V^2 \end{pmatrix}$$

We can always take the square root of a unitary matrix (and the root will also be unitary), so picking an appropriate V is always possible.

If U is the bit flip, the above construction makes a Toffoli gate. We can use this to compute any computable function with our quantum computer (though we would then lose the hypothesized time benefits of the quantum computer).

4.2 More complicated gates [§4.5.1 pp 189–191]

Consider an arbitrary $2^n \times 2^n$ unitary matrix U representing a quantum gate on n qubits. We can zero out the ith element of the first column by left multiplying by

$$V_1 = \begin{pmatrix} \alpha^\dagger & 0 & \cdots & \beta^\dagger & \cdots \\ 0 & 1 & \cdots & & \\ \vdots & & \ddots & & \\ \beta & \cdots & & -\alpha & \cdots \\ \vdots & & & \vdots & \ddots \end{pmatrix}.$$

The row with β as the first element is the ith row and

$$\begin{aligned} \alpha &= \frac{u_{11}}{\sqrt{|u_{11}|^2 + |u_{i1}|^2}}, \\ \beta &= \frac{u_{i1}}{\sqrt{|u_{11}|^2 + |u_{i1}|^2}}. \end{aligned}$$

It is easy to check that ${V_1}^\dagger V_1 = I$ so $V_1 U$ is also unitary. We can continue using similar V's to zero out the rest of the first column at which point the matrix will be of the form

$$V_{2^n-1} \cdots V_1 U = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & u'_{22} & u'_{23} & \cdots \\ 0 & u'_{32} & \ddots & \\ \vdots & \vdots & & \end{pmatrix}$$

The first row is in this form as the matrix is unitary. Now we can continue with matrices of the form

$$V_1 = \begin{pmatrix} 1 & 0 & \cdots & & & \\ 0 & \alpha^\dagger & 0 & \cdots & \beta^\dagger & \cdots \\ 0 & 0 & 1 & \cdots & & \\ \vdots & \vdots & & \ddots & & \\ 0 & \beta & \cdots & & -\alpha & \cdots \\ \vdots & & & & \vdots & \ddots \end{pmatrix}$$

to zero out the next layer of the matrix. Continuing in this form we will get the identity matrix. The V matrices gates are permutations with some not gates and finally a controlled gate where all but one of the bits are control bits, and the other bit can be operated on arbitrarily.

5 Deutsch-Jozsa algorithm [§1.4.4 pp 34–36]

Problem We are given an oracle for a black box function $f : \{0,1\}^n \to \{0,1\}$. The function f is guaranteed to be either constant or balanced and we wish to decide which. (A balanced function is 0 on exactly half of all possible inputs and hence 1 on the other half.)

Classical random algorithm Deciding if f is balanced or constant is possible in polynomial time with bounded error provided we have an oracle for f. Pick m random n bit inputs (uniformly). If f on all the inputs is the same, decide that the function is constant. If we get both 0 and 1 on some of the inputs, decide the function is balanced. If the function is balanced, the probability that we get the same output on all is 2^{-m+1} for $m \geq 2$. (We need it to be the same as the *first* output, which can be either 0 or 1). Thus, to get an error probability of less than (say) 1/3, we need only pick $m > 3$ and the error probability decreases exponentially in the number of random inputs we pick.

5.1 Quantum algorithm

We can compute the answer with certainty using quantum computation. Input is of the form $|0\rangle^{\otimes n}|1\rangle$, and we assume that we have a black box quantum gate, B_b on $n+1$ bits satisfying

$$B_b|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle .$$

where x is an n-bit binary number.

In the following description, we make use of the Hadamard gate

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

Let $|x\rangle$ represent the tensor product of $|0\rangle$ and $|1\rangle$ vectors in the order of the binary representation of x and $x \cdot z$ be the dot product of bit representations of x and z. Note that

$$H^{\otimes n}|x\rangle = \sum_{z=0}^{2^n-1} (-1)^{x\cdot z}|z\rangle$$

We begin:

$$\text{Apply } H^{\otimes(n+1)}: \qquad \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

$$\text{Apply } B_b: \qquad \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle (-1)^{f(x)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

$$\text{Apply } H^{\otimes n} \otimes I_{n+1}: \qquad \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{f(x)+x\cdot z}|z\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

Now measure all bits in the $|0\rangle$, $|1\rangle$ basis. There are two possibilities. Exchange the order of summations.

- If $f(x)$ is constant, the dot product for $z = 0$ is uniformly 0 so the summation over x with $z = 0$ produces a factor of 2^n. Since the state vector is normalized, it must be $(-1)^{f(0^n)}|0\rangle^{\otimes n}(|0\rangle - |1\rangle)/\sqrt{2}$. Thus we will measure $|0\rangle^{\otimes n}$ for the first n bits with certainty if f is constant.
- If $f(x)$ is balanced, the x summation over $|0\rangle^{\otimes n}$ is negative on half the x and positive on the other half and so cancels out: there is no probability of measuring $|0\rangle^{\otimes n}$.