Lecture 2 (30 Jan 2025)
Quantum probability and computation
Registers, states, gates, measurements, probabilities

Jaikumar Radhakrishnan
jaikumar.radhakrishnan@icts.res.in

Please let me know if you spot an error.

We discuss circuits with quantum registers, and model the states of the registers and the computation on them as vectors and linear transformations. We observe that the evolution of quantum states is tracked using linear algebraic computations using *unitary* matrices, analogous to how we tracked randomized computation using *stochastic* matrices. We will study how probabilities arise when a state is measured *in a basis*. We will study the state of a qubit, and visualize the actions of Pauli and Hadamard matrices using rotations and reflections performed on the Bloch sphere. Finally, we present the well-known quantum circuits of (i) superdense coding and (ii) Deutsch.

Quantum computation is performed by quantum circuits consisting of quantum registers and quantum gates. The picture is very similar to the ones we have seen for classical computing, but there are crucial differences:

(i) The state of the registers is no longer a probability distribution over basis states, but a superposition. Suppose there are n registers. These registers have computational basis states of the form $|x\rangle$ for $x \in \{0,1\}^n$. These are special states corresponding to an observation that can distinguish between them. However, the n registers can in general be in a state of the form

$$\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle ,$$

where the α_x's are complex numbers such that $\sum_x |\alpha_x|^2 = 1$. The coefficients α_x's are called amplitudes, and the linear combination is called a *superposition* of basis states. We will use $|\psi\rangle$, $|\phi\rangle$, etc., to denote denote quantum states. If we arrange the 2^n amplitudes as a column vector indexed by $x \in \{0,1\}^n$, then we get an alternative representation of the quantum state; note that we get a unit vector with respect to the standard inner-product in the 2^n-dimensional vector space $\mathbb{C}^{2^n}$. The above description corresponds to a *pure* quantum state; we will discuss the more general notion of a mixed state later.

(ii) As operations are performed on the registers, their state evolves linearly, that is, the corresponding transformation T satisfies, $T(\alpha|v\rangle + \beta|w\rangle) = \alpha T|v\rangle + \beta T|w\rangle$. There is a constraint: the operation must preserve the length of the vector, implying that the $2^n \times 2^n$

matrix corresponding to the linear transformation is unitary. The reversible classical gates we encountered before are clearly unitary. The dollar gate, which is central to our definition of randomized computation, is not reversible. Yet, we will see that randomized computation can be embedded in a suitable quantum computation. As in our study of randomized algorithms, the main goal will be efficiency, i.e., realizing the desired unitary using a circuit with a small number of gates and a small number of additional registers. Transformations implemented using quantum circuits that rely on additional work registers (called *ancilla*) but discard them in the end, are rather more involved: to understand them in some generality, we will need to go beyond pure states and unitary operations.

(iii) Probabilities arise in quantum computation when states are *measured*. Suppose $|\psi\rangle$ is the state of a system of n registers—we may think of $|\psi\rangle$ as a vector in $\mathbb{C}^{2^n}$, that records the amplitudes corresponding to each basis state. When the registers are measured (in the standard basis), we obtain an outcome and the state of the registers collapses to a state that is consistent with the outcome. When we *measure* n registers that are in the n-qubit state

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle ,$$

in the standard basis, we obtain one of the outcomes in $\{0,1\}^n$, just as we would if we were to observe the state of a classical register which is in the state $|p\rangle\!\rangle = \sum_{x\in\{0,1\}^n} p_x |x\rangle\!\rangle$. However in the quantum measurement, the probability of obtaining the outcome x is $|\alpha_x|^2$; note that we assumed that $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$. It is as if the n registers now take on the classical state $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 |x\rangle\!\rangle$. Then, if the outcome x is observed, the state collapses to the classical state $|x\rangle\!\rangle$. The framework just outlined will be adequate for the study of quantum algorithms, where one usually makes a measurement only at the end. However, this framework leaves the results unspecified in two important situations: (i) What happens if only some of the registers are measured?, (ii) What happens if the state is evolved together with some ancilla registers, and then some of the registers are discarded (we may then be left with fewer or more registers than we started with)? To systematically address these two questions, we will have to adopt a notion of a quantum state that simultaneously incorporates quantum superpositions and classical probability. The notion of quantum operations will also have to be enlarged to incorporate actions that add and discard registers, or use classical randomness.

Quantum gates

All classical reversible gates are quantum gates; a classical circuit consisting of reversible gates can be used as a quantum circuit. In particular, the NOT gate, the CNOT gate, the Fredkin gate and the Toffoli gate are valid quantum gates. Suppose we have such a circuit C acting on n classical registers. The action of C corresponds to a matrix with 2^n rows and 2^n columns, where each row and each column has exactly one 1. Often, it is easier to describe the action of such circuits using $|x\rangle$ and $\langle y|$, which are the quantum analogues of $|x\rangle\!\rangle$ and $\langle\!\langle y|$ that we used earlier. Thus, the CNOT gate is written as

$$|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|.$$

For the moment, we consider $\langle x|$ only when x is a standard basis state, that is, a string of zeros and ones; when $\langle x|$ meets $|y\rangle$ (assuming x and y have the same number of symbols), we get $\langle x|y\rangle$, which the scalar 0 if $x \neq y$ and 1 if $x = y$.

Example 1 (The swap gate) *The Swap gate acts on two registers and 'swaps their state'. That is, it acts as follows:* $|xy\rangle \to |yx\rangle$*, or more elaborately,*

$$|00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|.$$

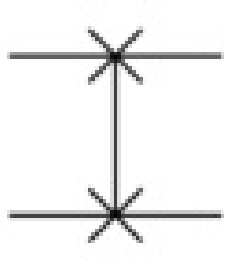

Figure 1: The two-qubit swap gate (Source: Wiki)

The action of such gates on superpositions of basis states follows rules that are familiar to us from classical randomized computing. Apart from these 'classical' gates, we will allow the use of some gates that do not have classical analogues. We allow all single-qubit gates that preserve the length of the state vector, that is, whose 2×2 matrices are unitary. In particular, we allow the single-bit rotation gates, $\mathrm{Rot}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ ($\theta \in [0, 2\pi)$), the remarkable Walsh-Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, the Pauli gates $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (another name for the NOT gate), $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and the $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ (also called the phase-flip gate). There are many more gates, of course (see, e.g., Wiki's list of quantum gates). It is clearly, unrealistic to allow computation with gates without considering if they can be implemented in practice on the specific platform used for quantum computation. In particular, rotations by very precisely specified angles are clearly questionable. Similarly, we are likely to be able to tell the difference if we substitute one operation by another operation whose matrix is close to that of the original one. In our discussion of algorithms, we will pay little attention to such practical considerations, though they are likely to very important when algorithms are required to

implemented on a physical platform. It turns out that we may restrict attention to a finite set of gates for all our quantum computation. In particular, computation performed using t gates from one set of two-qubit gates can be implemented with precision ϵ using gates from the set $\{\mathrm{H}, \mathrm{CNOT}, \mathrm{Rot}_{\pi/8}\}$ with at most $t \log^c(t/\epsilon)$ gates for $c = 3.97$ (this is know as the Solovay-Kitaev theorem, see Dawson and Nielsen or AM Child's lecture notes). While presenting our quantum algorithms, we will assume that arbitrary two-qubit gates are available, although most of the time, controlled version of the single qubit gate is all we need.

Self-check 1 *Recall the* CNOT *gate from Lecture 1 given by* $\textsc{cnot}(x, y) = (x, y \oplus x)$. *Implement the swap gate (from example 1) using CNOT gates. Use the following idea from programming to swap two variables without a temporary storage.*

$$x \leftarrow x + y; \quad y \leftarrow x - y; \quad x \leftarrow x - y;$$

The Bloch sphere

The state of qubit is a unit two-dimensional complex vector with amplitudes corresponding to $|0\rangle$ and $|1\rangle$. Since probabilities are obtained by squaring the absolute value of the amplitudes, multiplying such a state vector by a until complex number does not change the behaviour of the measurements. In this sense, the representation of the state vector as a unit complex vector is redundant: $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and $\frac{i}{\sqrt{2}}(|0\rangle - |1\rangle)$ correspond to same state of the qubit. If we were to be careful, we would say that the state of single qubit register is a vector of the form $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$, where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. Such a state is conveniently represented on the unit sphere in $\mathbb{R}^3$—the Bloch sphere..

Note that $|0\rangle$ is the north-pole and $|1\rangle$ is the south-pole. Every point on the equator corresponds to an 'equal' superposition of $|0\rangle$ and $|1\rangle$; in particular, the point where the X-axis meets the sphere is the $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. A Bloch sphere also helps in visualizing operations on a qubit it. For example, orthogonal states correspond to antipodal vectors on the sphere (check!). The Pauli gates X, Y, and Z correspond to rotations by π of vectors on the Block sphere, respectively about the X-axis, Y-axis and the Z-axis. In fact, the action of any unitary operation U on a qubit corresponds to rotation about a certain axis (the axis formed by the eigen vectors of U). The Hadamard gate H corresponds to rotation about the axis that makes angle $\frac{\pi}{4}$ with both the X-axis and the Z-axis; the two states that lie on this axis are $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, the eigen vectors of H. The Bloch sphere offers an appealing platform for visualizing single-qubit states

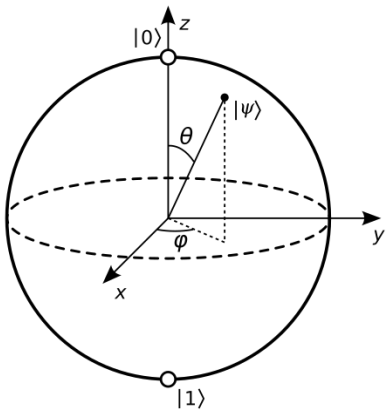

Figure 2: The Bloch sphere (source: Wiki)

and operations on them, but its utility for quantum algorithms is generally limited. (If used without care, it can sometimes be misleading: we said that rotation by an angle π about the Z-axis corresponds to the Pauli Z gate, which seems to suggest that it maps both $|0\rangle$ and $|1\rangle$ to themselves, but in fact it maps $|1\rangle$ to $-|1\rangle$.)

Deutsch's algorithm: phase kickback

Consider a quantum circuit F that implements a function f from $\{0,1\}$ to $\{0,1\}$ in a reversibly. That is, F has two registers A and B, and acts as follows on basis states. $F : (x,y) \mapsto (x, y \oplus f(x))$. We are given a circuit computing F and we would like to determine if f is a constant function. This is Deutsch's problem. We may embed C in bigger circuit, generate arbitrary quantum states on the two registers and process the state that results before making the final measurement to determine the answer. No classical randomized method can correctly determine the answer with probability of correctness better than $\frac{1}{2}$ (why?) if it is allowed to probe F only once, that is, if C is allowed to have only one copy of F embedded in it. On the other hand, the following quantum circuit, which has only one copy of F, determines the answer correctly with probability 1.

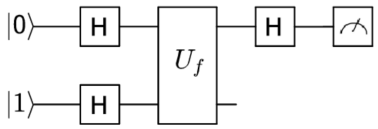

Figure 3: The circuit for the Deutsch's problem (credit: Timoteo Carletti)

The method is based on a uniquely quantum phenomenon that has come to be known as *phase kickback*. If the input registers to F are prepared in the state $\frac{1}{\sqrt{2}}|b\rangle(|0\rangle - |1\rangle)$, the registers come out in the state $\frac{(-1)^{b(f(0)-f(1))}}{\sqrt{2}}|b\rangle(|0\rangle - |1\rangle)$ (the $(-1)^{b(f(0)-f(1))}$ is the phase kick-back). Note that the second register is 'restored' to the state it started in. So focusing on the first register alone, we have have two cases. If $f(0) = f(1)$ (f is a constant function), then the circuit behaves like the identity gate; if $f(0) \neq f(1)$, the the circuit behaves like a Z gate. Is there a state that we could prepare the input registers in so that I and Z would take them to orthogonal states? Indeed, the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Putting these ideas together, we obtain a circuit fig. 3 to solve the Deutsch's problem (with no error).

The EPR state, entanglement, superdense coding

Suppose we have two quantum registers, each holding a qubit. They are originally in the following state (how might have they got into this state?):

$$|\mathrm{EPR}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B\right).$$

Now consider the action of the following four operations on register A: I, X, Z and ZX (not doing anything on B amounts to appying the

identity on it). We have

$$(I_A \otimes I_B)\,|\mathrm{EPR}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|0\rangle_A\,|0\rangle_B + |1\rangle_A\,|1\rangle_B\right)$$
$$(X \otimes I_B)\,|\mathrm{EPR}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|1\rangle_A\,|0\rangle_B + |0\rangle_A\,|1\rangle_B\right)$$
$$(Z \otimes I_B)\,|\mathrm{EPR}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|0\rangle_A\,|0\rangle_B - |1\rangle_A\,|1\rangle_B\right)$$
$$(ZX \otimes I_B)\,|\mathrm{EPR}\rangle_{AB} = \frac{1}{\sqrt{2}}\left(-|1\rangle_A\,|0\rangle_B + |0\rangle_A\,|1\rangle_B\right)$$

Note that the four states that result are mutually orthogonal. Thus, there is a unitary transformation U_{AB} that maps the states $|00\rangle_{AB}$, $|10\rangle_{AB}$, $|01\rangle_{AB}$ and $|11\rangle_{AB}$ to them respectively. Indeed, applying the Hadamard gate to A followed by CNOT to AB (controlled by A) achieves precisely this mapping (check!). This observation can be used to efficiently transmit classical information in the presence of entanglement.

Suppose there are two parties, Alice and Bob. Alice has two bits of information: $a_1a_2\ldots \in \{0,1\}^2$. Alice and Bob have two other other registers A and B, where register A is in Alice's possession and register B is in Bob's possession. These registers are in the state $|\mathrm{EPR}\rangle_{AB}$. (In **??**, Bob prepares $|\mathrm{EPR}\rangle_{AB}$ and send the register A over to Alice.) The equations above show how by selectively apply Z and X to the register A alone, Alice can produce four mutually orthogonal two-qubit states. If she then sends register A to Bob, the state of the pair of registers A and B together can reveal precisely which of four possible operations (namely, no operation, only X, only Z and ZX) Alice had performed on register A. So the strategy is as follows. First, if $a_2 = 1$, then Alice applies X to register A; next if $a_1 = 1$, then Alice applies Z to A. Alice send A over to Bob, who determines the bits a_1a_2 using the inverse of the operation U_{AB} discussed above (see fig. 4).

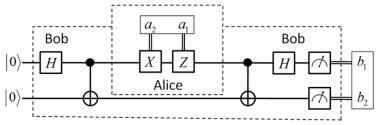

Figure 4: Superdense coding (credit: Walter V. Pogosov)