Algebra and Computation Course Instructor: V. Arvind

Lecture 20: Introduction to quantum computation

Lecturer: V. Arvind *Scribe: Vipul Naik*

1 Physical systems and computational models

1.1 Computers as physical systems

A computer program takes certain input data, manipulates it using certain rules, and produces some output. If we assume that this manipulation is subject to "physical laws" we can consider the loose analogy:

- The computer is a physical system, or lab apparatus
- Running the program is like conducting an experiment
- The output is like the observations made from the experiment

For any computational model to be of practical interest to us, it should be implementable as a physical system. The interesting question is the reverse one: given any physical system, can we turn it into a useful computational model?

1.2 Feynman's question

Feynman wanted to know if quantum mechanics could be used to provide a useful computational model. There are the following questions:

- How can we describe an abstract computational model whose corresponding physical system is subject to the laws of quantum mechanics?
- How does the computational power of such a model compare with that of physical systems subject to the laws of classical mechanics?

1.3 The situation before quantum mechanics

Turing and Church had considered various computational models, such as Turing machines, random-access machines, and so on. All these computational models could be implemented through physical systems subject to the laws of classical mechanics. While studying many such computational models, computer scientists came up with the following Holy Grails:

1. **Church-Turing thesis**: This states that any computational model is as powerful as the Turing machine. In other words, given any computational model, we can simulate computations on that model using the Turing machine. The simulation may of course involve a blow-up in time taken as well as in space used.

2. **Strong Church-Turing thesis**: This states that for any computational model, a polynomial-time algorithm for a decision problem in that computational model can be simulated by a polynomial-time algorithm in the Turing machine model. In looser language, if we think of polynomial time as the notion of *tractability*, then tractability in any computational model is equivalent to tractability in the Turing machine model.

3. **Strong Church-Turing thesis** (randomized version): This states that for any computational model, a bounded-error probabilistic polynomial time algorithm for a decision problem in that computational model can be simulated by a bounded-error probabilistic polynomial time algorithm for the problem in the Turing machine model. In looser language, if we think of BPP as the notion of *tractability*, then BPP is any computational model is equivalent to tractability in the Turing machine model.

While (1) remains unchallenged, quantum computation challenges (2) and (3) – if we can think of the quantum computation model as sufficiently *reasonable*.

2 The two-slit experiment

2.1 The two-slit experiment with particles

Suppose a gun is placed behind a wall with two slits – with the gun firing bullets uniformly in all directions. There is a screen behind the wall that "picks up" those bullets which pass through the slits.

Now, we have the following intuitively clear fact: Suppose p denotes the total number of particles hitting per unit time at a point on the screen when both slits are open, p_1 denotes the number when one slit is open, and p_2 denotes the number when the other slit is open. Then $p = p_1 + p_2$.

In other words, every particular passes either through one slit or the other.

2.2 The two-slit experiment with waves

In the waves version of the two-slit experiment, the "source" is a light source rather than a gun, and light is radiated uniformly in all directions. Now, if A denotes the amplitude of light received at a point on the screen behind when both slits are open, and A_1 denotes the amplitude when only the first slit is open, and A_2 denotes the amplitude when only the second slit is open, then:

$$A^2 = A_1^2 + A_2^2$$

In other words, it is *not* true that $A = A_1 + A_2$ – there is a *cancellation* due to interference. What gets added is the total *energy* and not the amplitudes.

2.3 The dual nature of matter and waves

The surprising thing about quantum theory is that the same thing could behave both as particle and as wave – it behaves as a particle when the amplitudes simply add, and it behaves like a wave when the squares of the amplitudes add.

3 The setup of quantum theory

3.1 Basic axioms

In quantum theory, we have the following regarding the state of a physical system:

1. The possible outcomes form a basis of a $\mathbb{C}$-vector space, called the state space.

2. The current state of the system is an element in the state space, viz a $\mathbb{C}$-linear combination of the outcomes. If ψ_i denotes the component of this state along outcome i, then we have $\sum_i |\alpha_i|^2 = 1$

   The current state of the system is termed a quantum superposition of the states i for which $\alpha_i \neq 0$.

3. Given the current state of the system, if we try to "measure" the outcome, we will get outcome i with probability $|\alpha_i|^2$.

We can express the state vector as a column vector with the i^{th} entry being α_i.

3.2 Hermitian inner product

Let V be a $\mathbb{C}$-vector space. An inner product is a map $\langle | \rangle : V \times V \to \mathbb{C}$ satisfying the following conditions:

1. It is conjugate-linear in the first variable, viz:

$$\langle a + b | c \rangle = \langle a | c \rangle + \langle b | c \rangle$$
$$\langle \alpha a | b \rangle = \overline{\alpha} \langle a | b \rangle$$

2. It is linear in the second variable, viz:

$$\langle a | b + c \rangle = \langle a | b \rangle + \langle a | c \rangle$$
$$\langle a | \alpha b \rangle = \alpha \langle a | b \rangle$$

3. It is Hermitian-symmetric, viz:

$$\langle a | b \rangle = \overline{\langle b | a \rangle}$$

4. It is positive definite, viz:

$$\langle a | a \rangle > 0$$

   whenever $a \neq 0$

We will follow Dirac's notation. The basis vector corresponding to outcome i will be denoted as $|i\rangle$. This is also called the *ket* vector.

Given any state A we denote by $\langle A | \psi \rangle$ the Hermitian inner product of A and ψ, and we also call this the probability amplitude of ψ in A.

3.3 The way quantum states evolve

A unitary operator is an invertible linear operator from the state space to itself under which the Hermitian inner product evolves. When we apply a unitary operator, we essentially switch from the original orthonormal basis to a new orthonormal basis. This means that when we now make a measurement in the new basis, we will get one of the new basis vectors, with probability equalling the square of the modulus of its amplitude.

The power of quantum theory lies in the following fact: in discrete time, the evolution of the quantum state of a system is given by a unitary operator. That is, there is a unitary operator U on the state space that maps the initial state to the final state.

In matrix terms, we can view U as a unitary matrix which takes a state written as a column vector in the original basis, and outputs the column vector for it in the new basis.

3.4 Different quantum states giving the same probability

Note that if $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $(\beta_1, \beta_2, \ldots, \beta_n)$ are two different states such that β_i/α_i has norm 1 for every i, then they give rise to the same probability distribution.

In the particular case where β_i/α_i is the same for all i, we say that the two quantum states differ by a *phase* of ϕ (where the common ratio is $e^{i\phi}$).

Here are two points:

- The quantum states that we are interested in are those on the unit sphere (that is, those of norm 1) upto phase. That is, we identify two quantum states if they differ by a multiplicative factor of a phase.

  In mathematical lingo, this is the projective complex space of $n-1$ dimensions.

  Note that if two quantum states differ only by phase, then applying the unitary operator to both of them again gives quantum states that differ by the same phase.

- It may be possible for two inequivalent quantum states to give the same probability distribution – this happens when the ratios for each coordinate are complex numbers.

  However, it is *not* true that if two quantum states give the same probability distribution, then applying any unitary operator to both of them also yields quantum states giving the same probability distribution. In other words, the quantum state carries *more* information than simply the associated probability distribution.

4 Quantum superposition versus random sampling

4.1 Probability distribution versus quantum superposition

A probability distribution over a set $\{1, 2, \cdots, n\}$ is an association of a nonnegative real number p_i to each i such that $\sum_i p_i = 1$. Suppose we sample randomly from this probability distribution, and associate a reward a_i to picking i. Then the expected reward is:

$$\sum_i a_i p_i$$

A quantum superposition over a set $\{1, 2, \cdots, n\}$ of states, on the other hand, is an association of a complex number ψ_i to each i such that the corresponding probability distribution associates, to each i, the value $|\psi_i|^2$. In other words, given a quantum superposition, the probability of measuring the value i from that superposition is $|\psi_i|^2$.

This immediately raises some questions:

- If two quantum superpositions give rise to the same probability distribution, how are they physically distinguishable?
- What are the ways in which we can transform one quantum superposition into another?

4.2 Transforming probability distributions

Suppose we are given a probability distribution. Then, to transform the probability distribution, we could do the following: consider a transition, which, if starting at state j, goes to state i with probability q_{ij}. Then if the current probability distribution vector is $p = (p_1, p_2, \ldots, p_n)^t$, and Q is the matrix of q_{ij}s, the new probability distribution vector is Qp.

The matrix Q here has the property that every column sum is exactly one; such a matrix is termed a stochastic matrix.

Note that since we are multiplying with a stochastic matrix, and all the entries of a stochastic matrix are nonnegative, it is not possible to make probabilities *cancel*, or *kill*, each other.

4.3 Transforming quantum states

The fundamental difference between the classical probabilistic model and the quantum model is that in the quantum model, we perform the operator, not

on the probability distribution, but on the underlying quantum state. That is, we pick on a unitary operator, and transform the quantum superposition according to the unitary operator. Here are some important points to note:

- The entries of a unitary operator can be both positive and negative (in fact, they can even be complex). Hence, it is possible to use a unitary operator to make terms *cancel* each other
- Note that we are making the unitary operator act on the underlying quantum state. Hence, two quantum superpositions that start off by giving the *same* probability distribution could end up giving separate probability distributions once we apply the unitary operator.

5 Quantum theory and Boolean circuits

5.1 Boolean circuits

Instead of looking at the Turing machine model (a model of variable-length computation) let us look at the Boolean circuit model (a model of fixed-length computation). The reason for choosing the Boolean circuit model to compare with quantum theory is that in quantum theory, we need to work in a state space of fixed dimension.

There are two aspects to the Boolean circuit:

- The values taken by a finite set of variables at any given time. These correspond to the classical "state" of the system.
- The gates themselves, which perform Boolean functions on some values and output the results.

At any stage in the evaluation of a Boolean function using a Boolean circuit, we have some Boolean variables and some values associated with those Boolean variables. To convert this to the quantum setting, we need to consider a state space where each possible assignment of values to the Boolean variables constitutes an outcome. In other words, if there are n Boolean variables, there are 2^n possible outcomes, and the state space is the space $\mathbb{C}^{2^n}$. The set of feasible states is the unit sphere in this space, and if we go upto phase, then the set of feasible states is the projective space.

Having converted the current state of the system to a quantum outcome, the next step is to view the gates in terms of unitary operators which can thus be simulated in a quantum system. There are the following immediate problems:

- The Boolean gates we have seen have more inputs than outputs, so they don't even preserve the number of states
- The Boolean functions for AND and OR are far from invertible, whereas any unitary operator must be invertible.

We shall see how to overcome both these problems at once, by associating to any Boolean function f a unitary operator U_f with approximately the same number of variables, such that computing f is classically the same as computing U_f.

5.2 Unitary operator for a Boolean function

Suppose $f : \{0,1\}^n \to \{0,1\}^m$ is a Boolean function. Then consider first the following Boolean function: it takes as input $(m+n)$ Boolean variables $(x_1, x_2, \ldots, x_n, z_1, z_2, \ldots, z_m)$ and outputs $(m+n)$ Boolean variables, namely $(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m)$ where $u = z \oplus f(x)$.

First of all note that this Boolean function is involutive – it equals its own inverse. Thus, in particular, it is also invertible.

Now, this Boolean function is a permutation (in fact, an involutive permutation) on the set of all possible elements in $\{0,1\}^{m+n}$. Thus, it can be viewed as a permutation matrix sitting inside $\mathbb{C}^{\{0,1\}^{m+n}}$. Since permutation matrices are unitary matrices, we obtain a unitary operator U_f that computes this function.

5.3 Boolean circuit in the quantum language

In the classical picture of building a Boolean circuit, we start off with a Boolean function $\{0,1\}^n \to \{0,1\}^m$ and try to express it as a composite of the AND, OR and NOT functions in various ways.

Note that each of these AND, OR and NOT functions take in only a small number of variables and output only a small number of variables – they don't touch the other variables at all. Thus, even if they transform the state of the entire system, their actual effect is only in some small part of the system.

The parallel in the case of unitary operators would be: Write the unitary operator U_f as a short-length product of unitary operators each of which "affects" only a small number of variables. To make these notions rigourous, we need to introduce the notion of tensor products of operators.

5.4 Tensor product of vector spaces

Given two vector spaces V and W with bases e_i and f_j, the tensor product $V \otimes W$ is defined as the vector space with basis b_{ij}, with a map

$V \times W \to V \otimes W$

that sends $\left(\sum_i v_i e_i, \sum_j w_j e_j\right)$ to $\sum_{i,j} v_i w_j b_{ij}$.

The tensor product acquires a natural significance for state spaces. Namely, if $V = \mathbb{C}^m$ is the state space spanned by outcomes e_i of experiment I and $W = \mathbb{C}^n$ is the state space spanned by outcomes f_j of experiment J, then the tensor product $V \otimes W$ is the state space for possible outcomes of the combined experiment I, J.

In other words, each outcome for $V \otimes W$ gives both the outcome for experiment I and the outcome for experiment J.

Further, the amplitude of the outcome (i, j) for the combined experiment is the product of the amplitude of outcome i for experiment I and outcome j for experiment J.

Now, given a tensor product $V \otimes W$, it makes sense to talk of the tensor product of operators A and B where A is a linear operator on V and B is a linear operator on W. The idea is roughly to map each b_{ij} to the vector $Ae_i \otimes Bf_j$.

5.5 A quantum circuit

We can now see that if the "memory" stores n variables at any given time, then the state space is the n-fold tensor product of $\mathbb{C}^2$ where $\mathbb{C}^2$ is the state space for one quantum bit (or qubit).

Further, suppose the current state has n variables and there is a quantum gate that inputs r variables and outputs r of them (by applying a unitary operator). Then, if U_g denotes the unitary operator for that quantum gate (when acting only on those r variables), the overall unitary operator is:

$$U_g \otimes I$$

where U_g is viewed as acting on the space $\mathbb{C}^{2^r}$ of those r variables, and I is acting on the space of $\mathbb{C}^{2^{n-r}}$ for the remaining $n - r$ variables.

This helps tell us what the notion of a good quantum circuit should be:

A quantum circuit for U_f is an expression of U_f as a product of unitary operators, each of which can be expressed as the tensor product of a unitary operator acting on a small number of variables, with the identity map.

When the quantum circuit arises from a Boolean circuit, each of those small unitary operators will be the unitary operators corresponding to that Boolean circuit.

5.6 Particular cases: controlled NOT and controlled AND

The controlled NOT gate is obtained as a special case of the general construction.

5.7 Solovay's theorem