Chapter 1

Quantum Computing

1.1 Learning Problems

1.1.1 A Motivating Example

Suppose you're involved in a simple card game: A dealer places two cards[1] face down on a table. You win the game and a substantial prize if you can guess whether the two face-down cards share the same color. You're allowed to ask the dealer to reveal cards to you, but for each card revealed your potential prize gets smaller.

How many of the cards do you need to see to determine with certainty whether the cards share the same color? Maybe you don't need to know for certain. How does the probability of you being able to guess the correct answer relate to the number of cards seen?

If you have no information at all, you can't do substantially better (or worse) than a fifty-percent chance. In fact, seeing only a single card, does not give you any more knowledge of what the answer to the question is. If you wanted to know with certainty what the answer was, you would need to see both cards.

This is of course a very simple game, but it is an example of a type of problem that we refer to as *learning problems* or often *oracle problems*. These problems consist of a learner who is trying to determine the answer to some question, generally to find the output of a certain function. The learner starts out with incomplete information, but is given access to an *oracle function* she can query to gain more information. An oracle function–also sometimes called a black-box function–is one that you can evaluate the oracle at any input of your choosing, but you have no information about the function other than its responses to your inputs. The learner's goal is to determine

[1] Assume that the cards are either red or black, with equal probability of each occurring

the answer to the question in as few questions as possible.

We rephrase the scenario given above in this language. Choose 0 to represent a black card and 1 to represent a red card. Suppose the two facedown cards are labeled a and b.

Example 1. Given oracle access to a function $f : \{a, b\} \to \{0, 1\}$. What is the minimum number of queries required to determine $f(a) \oplus f(b)$ where $\oplus$ denotes addition mod 2?

TODO: rewrite this example

Phrasing problems in terms of oracles is an extremely useful tool. It provides an approach for using information theoretic arguments to provide algorithmic lower bounds. We know of very few tools to show that there is no way to possible solve something quickly.

Example 2. **TODO**: Example 2?

1.2 Classical Computing

1.2.1 History, Bits, Information Theory

Before we delve into *quantum computation*, we first briefly give an overview of the relevant concepts from the study of *classical* computation. Although we will not need information theory for any of the arguments we will make, it will be used frequently as a source of intuition.

The fundamental idea of information theory is that any the amount of information that an object contains is determined exactly by the number of possible states it could have been in. In this way, the bit, which can be in exactly one of two possible states is the most basic unit of information.

In the 16th and 18th centuries mathematicians and logicians worked to convert formal systems of logic into algebra and arithmetic. One of the most prominent successes of these attempts was by George Boole who managed to faithfully encode propositional logic in the language of arithmetic and algebra. The idea was to let 1 and 0 represent true and false respectively. Then, by interpreting addition and multiplication as mod 2, one obtains exactly the logical operations `AND` and `XOR`. In modern terms, we recognize this as the structure of the field $\mathbb{F}_2$. **TODO**: remove this paragraph

TODO: *These next two paragraphs just repeat the previous two paragraphs, clean this up*

We define a bit as an element of $\mathbb{Z}/2\mathbb{Z} = \{0, 1\}$. It also can be useful to think of bits as representing a true or false value. This relationship gives a direct correspondence between classical computation and propositional logic. We write the state space of a bit as $\mathbb{Z}/2\mathbb{Z}$ to emphasize that it has a natural additive operation given by mod 2 addition. Alluding to propositional logic, this operation is called exclusive-or, usually written as `XOR` or $\oplus$, although we will oftentimes just write $+$.

Of course having just one bit, is not particularly interesting. As a reference, it is common today to measure memory in terms of *gigabytes*. One gigabyte is equivalent to eight billion bits. We most often work with *strings of bits*. A string of 2 bits has four possible states, each described by an element of $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. Analogously, an n bit string has state space $(\mathbb{Z}/2\mathbb{Z})^n$. The group addition readily extends to n bit strings giving it the operation that is commonly known as bitwise XOR.

Thus any function $f : \{0, 1\}^m \to \{0, 1\}^n$ can be expressed by some sequence of logical operations and each sequence of logical operations corresponds to a function.

1.2.2 Circuits and Logic Gates

Definition 1 (Nielsen and Chuang, pg 129)**.** A *circuit* is made up of wires and gates, which carry information around, and perform simple computational tasks, respectively.

Definition 2 (Nielsen and Chuang, pg 129)**.** A *logic gate* is a function $f : \{0, 1\}^k \to \{0, 1\}^\ell$ from some fixed number k of input bits to some fixed number ℓ of output bits.

Example 3. Here is an example of a logic gate

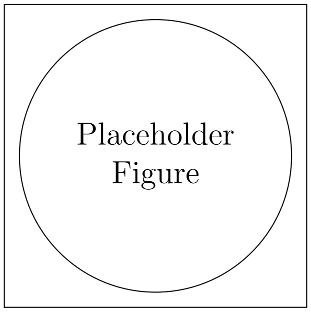


Figure 1.1: The logic gates `AND` and `XOR`

Example 4. Here is an example of a circuit

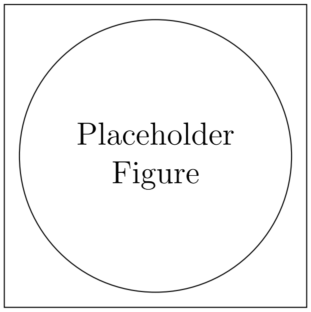


Figure 1.2: An example of a basic circuit, maybe make it match whatever example I use for a circuit family

Theorem 1. *Any function $f : \{0,1\}^n \to \{0,1\}$ can be represented by circuit a consisting of only* `AND`*,* `NOT`*, and* `OR` *gates.*

I don't know how to fill the rest of this circuit part in

1.2.3 Circuits as Algorithms

Definition 3 (verbatim from Sipser – cleanup)**.** A *circuit family* is an infinite list of circuits $(C_0, C_1, C_2, ...)$, where C_n has n input variables.

Definition 4. The circuit complexity of a function is the size complexity of a minimal circuit family for that language.

Example 5. Example circuit family to compute something. Maybe how many input bits are odd.

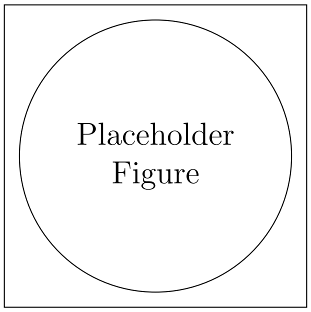


Figure 1.3:

1.2.4 Introduce controlled gate notation

Just example of Toffoli gate or whatever

1.2.5 Circuits with Oracles

We can consider circuits with oracle access. Give example from original motivating problem.

1.3 Quantum Computing

Now we introduce the concepts required for quantum computation. In the quantum world, the *qubit* is the fundamental unit of information. In the same way that we could express any classical algorithm as a circuit of logical operations on bits, we can express any quantum operation as a sequence of *unitary* operations on *qubits*.

The following definitions will give a complete description of a model of quantum computing. These can all be derived from the postulates of quantum mechanics.

Definition 5. Formally, a *qubit* is unit vector in $\mathbb{C}^2$. That is a pair of complex numbers

$$(\alpha, \beta) \in \mathbb{C}^2$$

satisfying the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1$$

We consider the state space $\mathbb{C}^2$ to be equipped with the standard inner-product.

It is very hard to picture qubits because they live on the unit sphere in $\mathbb{C}^2$. However, if we naively think of them as vectors in $\mathbb{R}^2$ we can generate some limited geometric intuition.

Although a qubit can exist in an uncountable number of states, paradoxically, they can only contain a finite amount of classical information. This is because in order to extract *classical* information from a qubit, it must be *measured*.

There are many interpretations of what measurement is and many deep philosophical questions, however it is very clear from experimental evidence what outcomes it produces.

Definition 6. A measurement is a

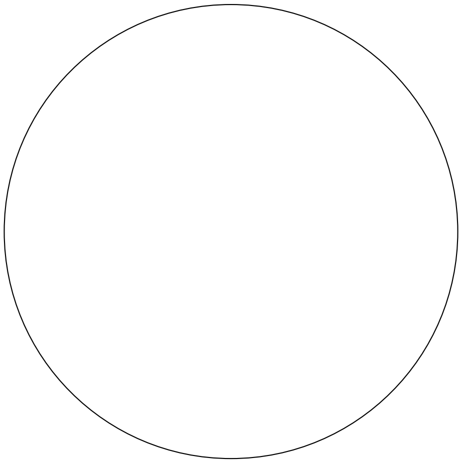


Figure 1.4: An unfaithful depiction of a qubit (finish this diagram)

Postulate 1. *Given a qubit $|\phi\rangle \in \mathbb{C}^2$, and any orthonormal basis $\mathcal{B} = \{|q_0\rangle, |q_1\rangle\}$ of $\mathbb{C}^2$, we may perform a* measurement *of ψ with respect to the basis $\mathcal{B}$. The result of this measurement is $|q_0\rangle$ with probability $\langle q_0|\phi\rangle^2$ and $|q_1\rangle$ with probability $\langle q_0|\phi\rangle^2$.*

This is only a specific case of the more general measurement postulate of quantum mechanics. Refer to [1].

Example 6. Given a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, a measurement of ϕ in the basis $|0\rangle, |1\rangle$ will return $|0\rangle$ with probability $|\alpha|^2$ and $|0\rangle$ with probability $|\beta|^2$.

1.3.1 Gates

As postulated, measurements can be made in any orthonormal basis. This means that, in some sense, qubits can only carry information up to a choice of orthonormal basis. For this reason, instead of thinking about measurement being performed in any basis, we could equivalently fix a basis for measurement, but allow for arbitrary change of basis maps on each qubit. **FIX**: This sounds awkward.

These maps are exactly the ones that preserve the inner-product on $\mathbb{C}^n$ (because they send pairs of orthonormal vectors to pairs of orthonormal vectors). This is often called the set (group) of isometries of $\mathbb{C}^n$ which is $U(n)$, the set of 2×2 unitary matrices with determinant one.

Definition 7. A matrix U is said to be unitary if its conjugate transpose is equal to its inverse. That is often written as $U^\dagger = U$.

Quantum computation is the manipulation of qubits by these unitary matrices.

Definition 8. An n-qubit *quantum gate* is a 2^n by 2^n unitary matrix.

Example 7. An important single qubit gate is the Hadamard gate defined by

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

It has the nice property that it is it's own inverse, i.e. $H^2 = I$.

The Hadamard sends

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \longmapsto \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \longmapsto \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

TODO: Unit circle diagram

TODO: Introduce Generators of U2?

TODO: Introduce other gates that we need

1.3.2 (Possibly) Relevant Theorems about Quantum Circuits

Theorem 2. *No cloning?*

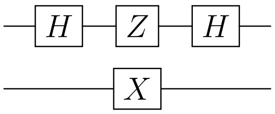
Definition 9. idk

Theorem 3. *Measurements can be made at the end*

1.3.3 Quantum Circuits

Much like classical circuits, a quantum circuit is made up of wires and gates.

Example 8. Deutsch-Josza



The Deutsch-Josza Algorithm

Example 9. Bernstein-Vazirani

Test

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.