0.1 Classical Computation and Information

Before we begin with *quantum* computation, we should first give a quick overview of *classical* computing. There are many different models for classical computing. The most famous perhaps is the turing-machine or the von-Neumman architecture (this is what most actual computers are modeled after).

In practice, most algorithms are studied in programming languages or psuedocode. However, we will focus on the *circuit model* of computation which will give us a more natural framework to generalize into the quantum setting.

Before we get to circuits, we first need to introduce *logic gates* and *bits*:

Definition 1. A *bit* is any element $b \in \mathbb{Z}/2\mathbb{Z}$. We will express b as either 0 or 1, although at times it may be convenient to think of a bit as true or false.

Logic gates correspond to boolean operations.

A Motivating Example

Suppose I have two boxes. I write down my two favorite integers on separate pieces of paper and lock them in different boxes. Your goal is to determine whether the sum of the two numbers is even or odd. I will open the boxes for you if you really want me to, but I really don't like opening boxes (perhaps I have a bad childhood memory relating to boxes). How many boxes do you need me to open to answer the problem?

The Bit

The fundamental unit of information is the bit. The term bit was introduced by Claude Shannon in 1948. It was a contraction of "binary information digit". A bit is the smallest nontrivial unit of information.

A bit b has two possible states. Generally, this is represented as

$$b \in \{0,1\} \cong \{\text{True},\ \text{False}\} \simeq \mathbb{Z}/2\mathbb{Z}$$

Sometimes thinking of a bit as true/false is helpful.

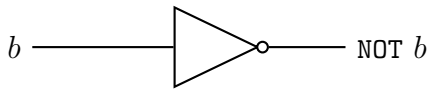
Importantly, we can have a list of several bits. A string s (of length n) is an ordered tuple of n-bits. This is expressed as a cartesian product

$$s \in \{0,1\}^n \cong (\mathbb{Z}/2\mathbb{Z})^n$$

Logic Gates and Bit Operations

We can perform operations on bits. We will express these as logic gates. If you've done propositional logic in a philosophy or math class, this is the same system just with different notation.

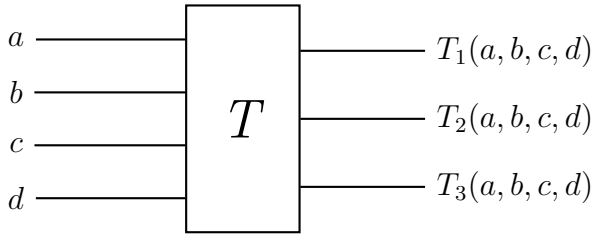
We start with the unary gates. These take one bit as input and return one bit. Here is a `NOT` gate:



It takes 0 to 1 and 1 to 0. It can be defined by the truth table:

0	1
1	0

There are in fact 4 unary gates, one for each of the 4 truth tables. Writing them out is a good exercise. Importantly More generally, the logic of any circuit is nothing more or less than a truth table. Here is an arbitrary circuit:

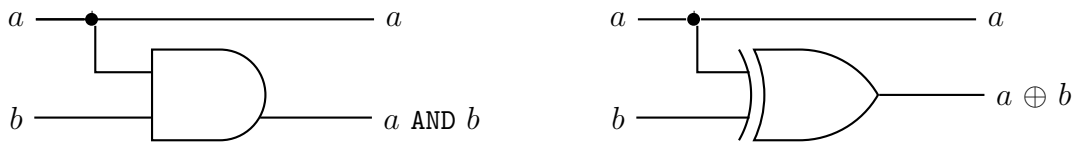


In this case $T : \{0,1\}^4 \to \{0,1\}^3$ takes as input 4 bits and returns 3 bits. I use the variable T to emphasize that could be thought of as a truth table.

There are two-bit gates `AND` and `XOR`. The fancy symbol here $\oplus$ means `XOR`. It gets a special symbol because it has the special property of being linear. In fact, it is

just addition mod 2.

I should mention that you can build any arbitrary circuit using `AND` and `NOT`. Here are two examples, `Controlled AND` and `Controlled XOR`



The second one, `Controlled XOR` is *reversible*. If we apply it twice, we get the identity. Such gates are classically important because they don't add entropy and thus generate less heat.

Quantum Computing and Quantum Information

The Qubit

Background: Just as the bit is the fundamental unit of information, the qubit is the fundamental unit of quantum information. It represents the state of a 2 state quantum mechanical system (the simplest nontrivial quantum system)

A classical bit is either 0 or 1. A qubit is a pair of complex numbers

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

satisfying the condition

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

Importantly, $\mathbb{C}^2$ is a 2-dimensional complex vector space and it has an orthonormal basis

$$\left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

In physics notation, we call these vectors $|0\rangle$ and $|1\rangle$, respectively.

Examples:

Multiple Qubits

We can also have multiple qubit strings. They are expressed as a tensor product. A two qubit string is a vector in $\mathbb{C}^2 \otimes \mathbb{C}^2$ and an n cubit string is a vector in $(\mathbb{C}^2)^{\otimes n}$. This may seem odd, especially if you are not familiar with the tensor product. However, there is good motivation for it. An important property of the tensor product, is that for two vector spaces V, W

$$\dim(V \otimes W) = \dim(V) \dim(W)$$

Therefore just as an n-bit string has 2^n possible states, an n-qubit string lives in a 2^n dimensional complex vector space. Although we can draw a (limited) picture for one qubit, it quickly becomes impossible even for two qubits.

You may ask why a qubit lives in $\mathbb{C}^2$. That is a good question that I would recommend asking a physicist. This does raise an issue though. $\mathbb{C}^2$ is a continuous vector space. This means that we couldn't even write down the state of a qubit as a finite number of classical bits.

Quantum Gates and Qubit Operations

There is one special qubit operation that is different from the rest, called measurement. It resolves this infinite information paradox we have run into.

When we measure a qubit, it "collapses" into a classical bit. It becomes $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. In math notation this is

$$\mathsf{Measure}\left[\alpha\begin{pmatrix}0\\1\end{pmatrix}+\beta\begin{pmatrix}1\\0\end{pmatrix}\right]=\begin{cases}\begin{pmatrix}0\\1\end{pmatrix} & \text{with probability } |\alpha|^2\\ \begin{pmatrix}1\\0\end{pmatrix} & \text{with probability } |\beta|^2\end{cases}$$