CS 150 Quantum Computer Science

Lecture 2: Digital Logic and Reversible computation

January 22, 2024

Lecturer: Saeed Mehraban *Scribe: Preliminary notes*

1 Logistics

- Course website: https://www.cs.tufts.edu/comp/150QCS/
- Piazza: https://piazza.com/tufts/spring2024/cs151
- Pre-assessment survey
- First problem set will be released tonight: Please study these topics Inner products, Hermitian matrices, Unitary matrices, Projectors. Your reading assignment for this problem set is is Sections 1-9 of https://www.cs.tufts.edu/comp/150QCS/Premath.pdf.

2 Overview

Last time:

- Overview: History, Models of Computation, Outlook.

Today:

- Classical v.s. quantum bits
- Digital logic
- Reversible computation

3 Quantum v.s. classical bits

3.1 Classical bits

- One bit of information is a Boolean number x which can be 0 or 1. Such bit may encode the answer to a Yes/No question or may mean an operations was successful or not, etc. In general binary data are represented using strings of bits. A string is concatenation of zeros and ones. E.g. 001010100.

- A digital computer is a machine that takes as input a sequence of bits, goes through certain (mechanical) steps, and convert the input string into an output string. For instance consider the task of deciding whether a number represented in binary is even or odd. The input is a number in binary for example 110. The output is 0 if the number is even and is 1 if the number is odd. In this case since 110 is even the output is 0. Computation in this case is simple looking at the first bit and copying it to the output.
- Classical bits can be realized using physical architectures. For example:
  - Switch: open or closed.
  - Magnet: up or down.
- At the level of implementation, we can store a string on a physical tape. Each position on the tape is a deterministic value 0 or 1 called "bit".
- Dirac notation: $101 \to |1\rangle|0\rangle|1\rangle = |101\rangle = |5\rangle$
- $|0\rangle_a|1\rangle_b$: register a holds 0. register b holds 1.

3.2 Classical probability

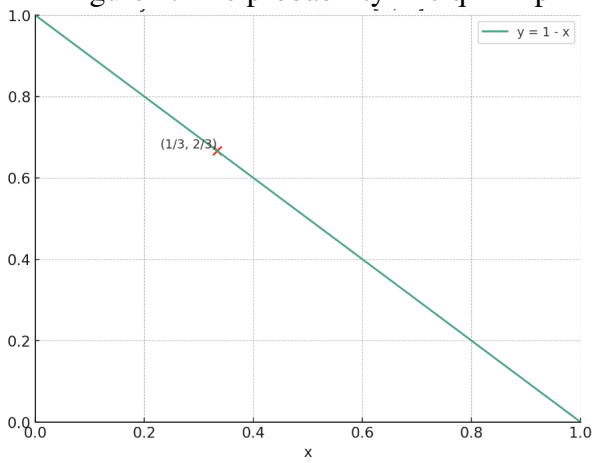
- We can also talk about probability. For example we can have a register that contains value 0 with probability $2/3$ and 1 with probability $1/3$. We can represent this state as a probability vector $\begin{pmatrix}2/3 & 1/3\end{pmatrix}$. More generally, we can represent it using the vector $|p\rangle = \begin{pmatrix}p\\q\end{pmatrix}$, for $q = 1 - p$ and $0 \leq p \leq 1$.
- We can consider larger (discrete) probability spaces. For example: $|\mathbf{p}\rangle = \begin{pmatrix}p_1\\ \ddots \\ p_N\end{pmatrix}$, $p_i \geq 0$ and $\sum_i p_i = 1$.

3.3 Quantum bits

We can encode a quantum bit within the degrees of freedom of a physical system: Electron spin up or down, photon polarization being clockwise or counter clockwise. Mathematically we have.

- Vector notation $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$, $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$.
- Superposition: $|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix}$ satisfying $|\alpha|^2 + |\beta|^2 = 1$.

Figure 1: The probability line q = 1- p



- **Example:** $|+\rangle := \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ and $|-\rangle := \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$
- We can have more quantum states by adding complex numbers. E.g. $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$
- **Measurement:** When we measure the quantum state $|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$ in the standard basis we obtain the value 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.
- **Question:** Can we distinguish $|+\rangle$ from $|-i\rangle$ in the standard basis?
- **Higher dimensional states:** $|\psi\rangle = \begin{pmatrix}\alpha_0\\\alpha_1\\\dots\\\alpha_{N-1}\end{pmatrix}$ then $\sum_i |\alpha_i|^2 = 1$

Conjugate vectors: We can define a conjugate vector $\langle\psi| = (\alpha_0^*, \alpha_1^*, \dots)$. Based on this definition $\langle\psi,\psi\rangle = \sum_i |\alpha_i|^2$ which is $= 1$ for a quantum state. We can also define inner product $\langle\psi,\phi\rangle$.

- $\langle 0,1\rangle = 0, \langle +,-\rangle = 0$.

4 Reversible operations on classical bits

It was known since the early days of quantum mechanics that quantum mechanics is reversible by nature. The question, hence, was whether we can perform computation using reversible elements. For instance the AND gate is not reversible. It dissipates part of the energy to the environment. During 1980's Ed Fredkin and Toffoli proposed a reversible model based on billiard balls that could simulate arbitrary computations.

Figure 2: The amplitude circle $\alpha^2 + \beta^2 = 1$

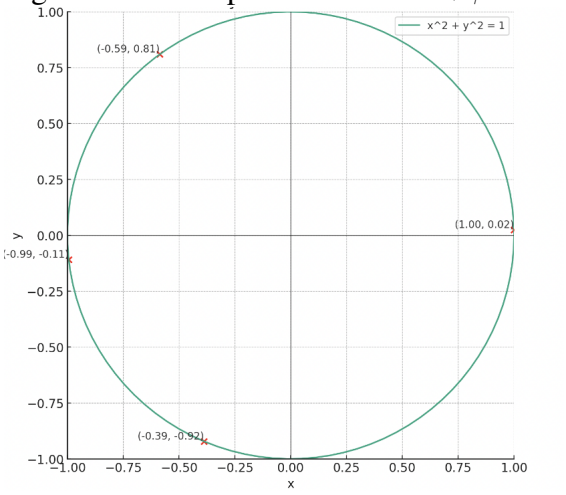


Figure 3: The Bloch sphere $|\alpha|^2 + |\beta|^2 = 1$

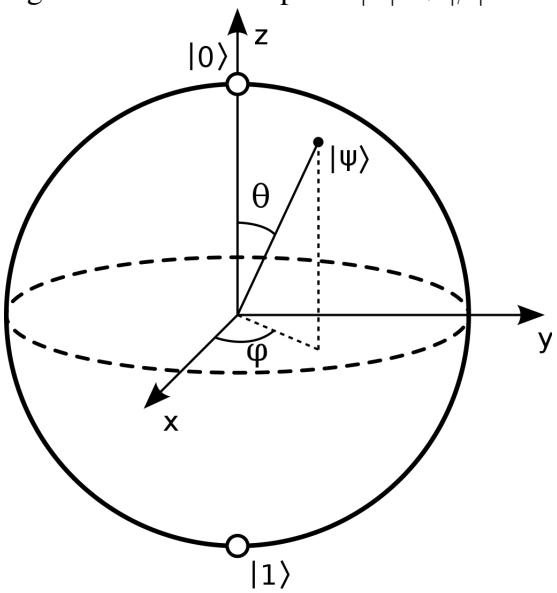


Figure 4: The truth table for functions NOT, AND and OR

x	$\bar{x}$
0	1
1	0

$f(x) = NOT(x)$

x	y	$AND(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$OR(x,y)$
0	0	0
0	1	1
1	0	1
1	1	1

4.1 Boolean logic

Let's first introduce Boolean logic. Introduce:

- **Boolean function:** A Boolean function is a function that takes a value in $\{0,1\}^*$ and outputs a value in $\{0,1\}^*$. For example the XOR function is the following function

$$XOR(x) = \begin{cases} 0 & x \text{ has even number of ones} \\ 1 & \text{otherwise} \end{cases}$$

  is a Boolean function. We often fix an integer n and work with functions $f : \{0,1\}^n \to \{0,1\}$. In other words $f(x) = x_0 + \ldots + x_{n-1} \mod 2$.

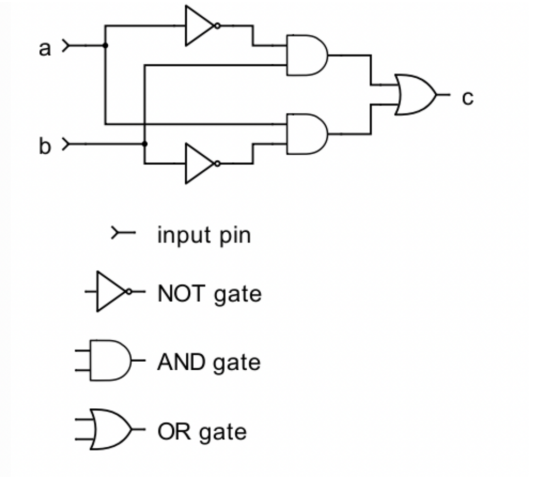
- **Truth table:** For a Boolean function the truth table for this function is a table which puts inputs and outputs next to each other. In particular, for each input bit $x \in \{0,1\}^n$ we have a row and for each row of the table has two parts $(x, f(x))$. You can see the example of the truth table for AND, OR and NOT in Figure 4. A truth table has therefore 2^n rows.

- **Question"** How many Boolean functions are there?

  Answer: 2^{2^n}

- **Boolean gate-set:** An important observation is that arbitrarily large Boolean circuits can be constructed using basic Boolean functions that take one or two inputs. A gate set is a fixed set of Boolean functions that take a certain (say $1, 2, 3, \ldots$) number of input bits and output a certain number of output bits. We use gate-sets to produce larger gate-sets.

Figure 5: Circuit for XOR .



- **Boolean circuit:** A Boolean circuit is a diagram that represents compositions of gates from a gate set. You can think of it as a directed acyclic graph. Each edge encodes a Boolean value and each node is a gate. See for example the circuit producing XOR in Figure 5 (From here).
- **Universal gate-set:** A gateset is universal if it can produce all Boolean functions. **Example:** $\{AND, OR, NOT\}$, $NAND$, NOR

Exercise: How many gates is sufficient to produce an arbitrary function?

4.2 Simple Boolean gates

Now let's study a few Boolean gates

- We already saw, AND, OR, NOT, and XOR gates.
- **Question:** Is AND reversible? How about OR?
- SWAP
- Controlled Not operation
- Toffoli

Exercise: Prove $SWAP = CNOT_{12}CNOT_{21}CNOT_{12}$.

4.3 Universal classical computation via reversible gates

Toffoli gate: $T(x, y, z) = (x, y, (x \wedge y) \oplus z)$.

Below we show that Toffoli gate is universal for classical computation

- **AND:** a.k.a. controlled-controlled NOT $T(x, y, 0) = (x, y, x \wedge y)$
- **NOT:** $T(x, 1, 1) = (x, 1, \bar{x})$
- **OR:** $T(\bar{x}, \bar{y}, 1) = (\bar{x}, \bar{y}, NOT(\bar{x} \wedge \bar{y})) = (\bar{x}, \bar{y}, x \vee y)$
- De Morgan's Law

Fredkin gate: Controlled SWAP

$$F(x, y, z) = \begin{cases} (x, y, z) & x = 0 \\ (x, z, y) & x = 1. \end{cases}$$

Exercise: Prove the Fredkin gate is Universal.