MIT OCW Quantum Information Science I Notes

Sandesh Katakam

November 2021

Contents

1 Quantum and Classical Computing Fundamentals:

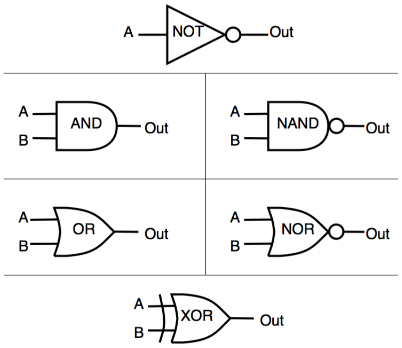
Concepts covered:

- Circuits/Reversible Circuits
- Quantum states(Vectors)
- Quantum Evolution
- Measurement
- Tensors
- Quantum Weirdness

1.1 Historical developments:

- Models for Classical Computation:
  - **Turing Machine** → Universal.
    - 1936 → Turing Universal Computation. Universal means give any computation, you can do it on a turing machine (or) you can do it with a lambda calculus
  - **Finite Automata** → Not Universal.
    - It doesn't solve all the problems.
  - **Lambda Calculus** → Universal.
    - STUFF TO BE WRITTEN.....
  - **A Coin** → Not Universal.
    - STUFF TO BE WRITTEN.....

  - **Circuit Model** → Super Universal(which can be solved with cic).
    - This Model is older than Turing Machines.



- **Why did Turing not use circuit Model?**
  - Circuits are only designed for finite size Input
  - If you want to have some kind of computation that is independant of the size of input. Circuits don't really do that.
  - **Ex:** Does Turing Machine with program(P) and Input i halt.
    This is uncomputable, but can solve it with family of circuits → circuits with $2^{p \cdot i}$ inputs.
    - Outputs:
      - 1 → If T.M. Halt.
      - 0 → If T.M. doesn't Halt.
  - There is not a single circuit for every size input(??)
  - We can hide uncomputable information in the design of circuit for size "n". This is why Turing came up with Turing Machine than Circuit Model.
  - So far no one has built a **Quantum Turing Machine**.
  - In 1994, Peter Shor came up with an algorithm called **"Shor's Algorithm"**. Shor's algorithm is a quantum algorithm for factoring a number N in O((log N)3) time and O(log N) space. The algorithm is significant because it implies that public key cryptography might be easily broken, given a sufficiently large quantum computer.
  - **Circuit Model:** The Description of the circuit should be out put by a classical computer program. We choose implement circuit model for studying Quantum computations.
  - **Classical Computer:**
    - Input n(size of input)
    - output : Quantum circuit of size "n"
    - This can be a Turing Machine.
- **Quantum Mechanics:**
  - **Einstein Podolsky Rosen (1936)** Published that Quantum Mechanics is **"Incomplete"**.
    - **Reason:**
      Given two particles with Opposite Momentum$\{+p, -p\}$ and Opposite positions$\{+x, -x\}$

You measure the **position/momentum** of one
It gives the **position/momentum** of the other particle

    - You cannot measure the position and momentum because QM says the position and momentum are conjugate variables.

    - **Schrodinger's explanation of EPR paradox** → These particles are entangled and that's the property entangled particles have.
    - **1964: John Bell** proved that no classical explanation for EPR.
    - **1981: Alain Aspect** showed Bell's predictions were correct.
    - **1982: N Herbert** Published a paper

      FLASH → First Laser Amplified Superluminal Hookup.

      The paper demonstrated faster than light communication using weird properties of EPR Pairs.
      This Paper is indeed proven to be wrong!. One of the refrees of the paper is **Arthur Peres**.
      The paper was proved wrong using **No-cloning Theorem**.

  - **No-cloning Theorem:**
    A single unknown quantum state cannot be duplicated. ....
  - **1982:** Feynman and Yuri Manin:
    - "Very difficult to simulate quantum mechanics".
    - Requires 2^n(exponential) time for "n particles". Complexity: $\mathcal{O}(2^n)$.
    - Quantum Computers would be faster.
  - **1992: Duetch-Jossa** → We can apply quantum computation to classical computing problems.
  - **1993:** Bernstein and Vazirani, Simon in 1994 came up with a problem that is exponentially faster on a quantum computer.
    - This Problem uses Periodicity.
    - Discrete log and factoring which are very important classical problems.
  - In 1994, **Shor's Algorithm** factoring a discrete log on Quantum Computer.
  - Simulating Quantum Mechanics is faster with a Quantum Computer.
  - With advances in High Performance Computing(HPC) classical computers are getting better at simulating quantum mechanics too.
  - **Are quantum computers good for classical computing?**
    - In 1995, **Lov Grover** came up with high search algorithm which searches a space of size n. Grover's algorithm has a complexity of $\mathcal{O}(\sqrt{n})$.
    - One of the disadvantage is you cannot do error correction on a quantum computer.
    - If we want to factor a real number, if every quantum operation and every quantum gate is not accurate to $\frac{1}{10^9}$. So, the errors are going to mount up and the result will be worthless.

- **Errors:**
  - **Classical Fault Tolerance:**
    - **Check Pointing:** Write down state of system. If something goes wrong, you don't have to start at the beginning.
      Note: The Advanced microdevices had check pointing. The Techniques are called **Little Hammer** and **Big Hammer**. If something went wrong during a computation, they start again at the checkpoint.

    - You can only use this when you have error detecting method. You can use error detecting method from other technique together with his to ensure the system has error detection capability.

  - **Error Correcting Codes:**
    Map K bites to N bits and you can recover from some number of errors. All computer memory and communication devices use this technique.
    These have Error detection capabilities.

  - **Massive Redundancy:**
    You keep many copies of your computation, if one of them goes wrong, you can compare them with others. If one system is different you throw it out and use or copy the state of one of the other computation. You can do massive redundancy for each bit.

- **Which or the techniques are compatible with "No-Cloning" theorem?**

    - ∗ Massive Redundancy: It is fine to start out with n copies of computation. It's fine to compare them with each other to see if they are wrong. If one goes wrong, you have to copy the state of another to this new one or else you will be left with (n-1) copies of computation. For every error detected our no. of copies goes down, so for large computation our no. of copies keep going down at every step of the process. So, **This doesn't work**.

    - ∗ Check Pointing : Check pointing involves writing down the state which is impossible according to No-cloning theorem.

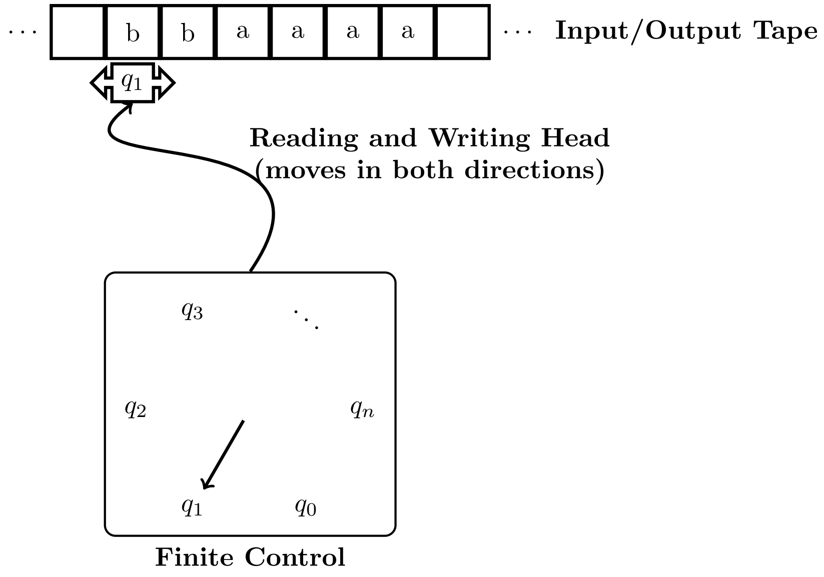
    - ∗ Error Correcting Codes: They work(compatible) but the reason nobody does computation with them in classical computing is that it requires a lot more overhead than using them for communication and Memory. Classically, the other techniques are less resource intensive than E.C.C.

- **Physical and Conceptual Models of Classical Computation:**

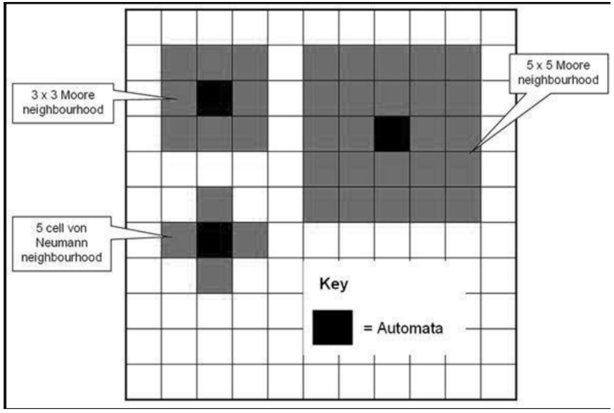
    - **Models Universality:**

        1. **Mechanical:** Examples are Abacus, Old rotary curta-type calculators, thermostats on walls.
           One of the famous mechanical model is Charles Babbage's **"Difference Machine"**.
           Electrical : computations out of electrons.
           Optical : Computations out of Photons.
           Biological : Computations out of Neurons.

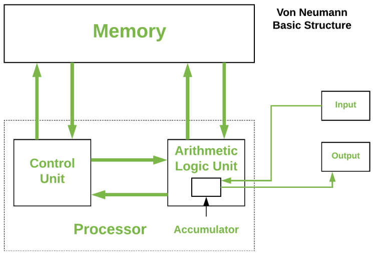
        2. **Conceptual:**
           (a) **Turing Machine:**
            - ∗ Probabilistic Turing Machines
            - ∗ Universal Turing Machines
            - ∗ Minsky Machines(named after Marvin Minsky)
            - ∗ **Components of a TM:**
                - · A Finite state Machine
                - · A representation of different states and transition between states, which happen depending on what is read at what time and the previous state it existed in.
                - · A Head → It can read/write from the Tape.
                - · Infinite length Tape (Kind of acts like a Memory).



(b) **Cellular Automata:** The Model of Computation is a world which is a grid in 2-D and N-dim, where the point is that you have some kind of state in a local cell of this grid, and it undergoes transitions based on the state of its neighbours.
Ex : Conway's Game of Life.


3 x 3 Moore neighbourhood
5 x 5 Moore neighbourhood
5 cell von Neumann neighbourhood
Key
= Automata

(c) **Von Neumann Architecture:**
STUFF TO BE WRITTEN...



(d) **DNA-Based:**
Computation based on the idea that uses DNA strands.
You have strands of bases AGCT, A and T associate each other and this is called **Ligation**.
When you have two different strands of DNA, they will pattern match other strands in right locations which matched the sequence of bases to produce Base-pair ligations.
These Bio-operations include primitive operations:

∗ Melting of strands → dividing strands.

∗ Appending/adding two strands end to end.

∗ cutting, restriction, flipping, PCR(Polymerase chain reaction).

- we are now starting to reconsider what it means to do computation.
- we need to generalize the idea of compuatation, because we are at the end of silicon road.
- we cannot rely on Moore's law much longer to provide an increasing scaling that's exponential of capability.
- we need to look at different physical mechanisms to build computations.

- **Big Questions The Physics of Computaion?**
  - How do we thinkn of physical mechanisms as doing computation?
  - How do we exploit physical and bio mechanisms that exist to realize the computations tha we want to achieve?

- **Universality:**
  All Boolean circuits can be composed of **AND** and **NOT** gates.

  - **Boolean functions:**

$$f(X_0, X_1, ..., X_{n-1}) = 0 \, or \, 1$$

$$\text{N bits of Input}$$

$$f(X_1, X_2) = X_1 * X_2 \rightarrow AND \, gate$$

$$f(X) = \overline{X} \rightarrow NOT \, gate$$

- **Complexity:**
  - Some math problems are harder than other.

  - The point is their scaling w.r.t size of problem.

  - **Strong Church-Turing Hypothesis:**
    * Any model of computation can be simulated on a probabilistic Turing Machine.

    * This simulation cost comes with almost a polynomial increase in No. of elementary operations required.

  - **Motivating factors of quantum computing:**
    * The difference between two of the most important classes of Mathematical Problems.
    * Many problems can be expressed as decision problems.
      Ex: Is "m" prime? → "PRIMALITY". It was not known for many years. You can answer the question with some randomness, but wouldn't know it for certain. This is **Rabin's Primality testing algorithm**. Today there is a deterministic primality testing Algorithm.

  - **FACTORING:**
    Given a composite integer m, and another integer $l(l < m)$. Does m have a factor that is non-trivial? (Y?N)

    * If the time needed to answer the question is polynomial w.r.t to the size of question. It is **Polynomial** in complexity.
      If time ∼ poly(size):

$$Problem \in P$$

    * If the "YES" instances of the problem are easily verified with the aid of a witness, then the problem is in the Class **NP**.

$$Problem \in NP$$

      And for the sake of completness we have another parallel to this, which is a mirror image.
    * If the "NO" instances of the problem are easily verified:

$$Problem \in CO - NP$$

    * **Where does QC fits in this landscape?**
      QC sits in a different kind of landscape. It has a complexity class **"BQP"**.
      Reason: Model of computation is different. It doesn't fit directly into any of this classes, because sometimes the output is Quantum-Mechanical.

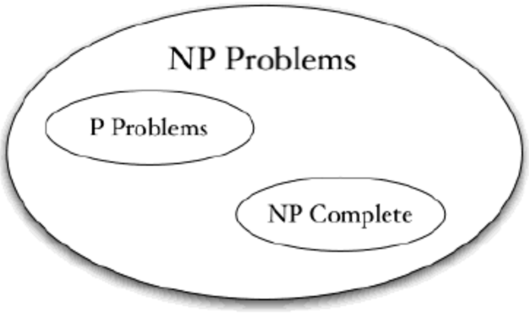


Figure 1: Diagram of complexity classes provided that P NP. If P = NP, then all three classes are equal.

- Example of NP-Complete Problems:
  - **3-SAT problem:** This is about the satisfiability of Boolean functions.

    $$f(x_0,....,x_{n-1}) = (x_1 + x_3 + x_9)(x_4 + \overline{x_7} + x_1 1)........$$

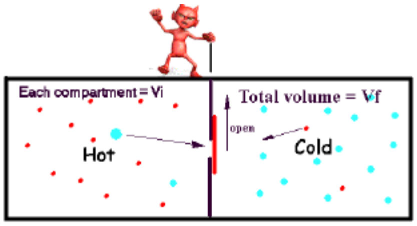
  - Given N inputs for a boolean function. Each term in the RHS has 3 bits.
  - **Goal:** Does there exist an assignment of 0s and 1s to the 'x' inputs such that the output is equal to 1. If we can solve the problem in polynomial time with N. Then you can solve all the rest of NP-Complete problems.
- Another class of problems that encompasses all these classes is called **Sharp**. The problems of the kind "counting the no. of solutions".

- **Universality of AND and NOT:**
  - AND and NOT are universal for boolean.
  - AND and NOT → OR (**"Demorgan's Theorem**).
  - **Proof:** Let $f : (n+1)\,bits \rightarrow \{0,1\}$

    $$f_0(x_1,.....,x_n) = f(0,x_1,x_2,.....x_n)$$
    $$f_1(x_1,.....,x_n) = f(1,x_1,x_2,......x_n)$$
    $$f(y,x_1,x_2,....,x_n) = f(1,x_1,x_2......,x_n)$$
    $$\rightarrow recurse....$$

- **Thermodynamics and computation(Maxwell's Daemon):**
  - **Maxwell's Daemon:** The experiment came out from a correspondence letter. Maxwell's daemon tries to achieve a configuration where he/she puts all cold molecules in the left and hot molecules to the right side. Then create an Imbalance of temperature. This is actually is impossible!

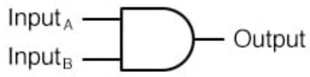
Is it because of energy of energy required to move the wall?
(or)
Measurement?

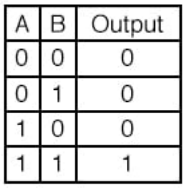


- **Szilard's Engine(1929):** There are 8 steps:
  1.
  2.
  3.
  4.
  5.
  6.
  7.
  8.
- **Rolf Landauer(1961):**
  - n-bit register → (2^n configuration)
  - Like n-boxes of scratch pad in configuration.
  - Clearing it in standard state is like compressing a gas:
    - all configuration go to final state.
    - Entropy(randomness of register) must go to environment (due to 2nd law of thermodynamics→ entropy is conserved.)
    - erasure must cause some energy($kT\log_e 2$) Per Bit.

      $$k \rightarrow Boltzmann's\, Constant$$
    - Heat pumped ($\delta Q = T\delta S$)
    - This result is known as "Exorcism of Maxwell's Daemon".
    - This is a beautiful connection between fundamental physics and computation.
    - **Could we build a compute that doesn't need to erase?**
      If we could build a computer that did not need to erase information, may be it would not need to dissipate energy to do the computation. In principle, we can do → "Reversible Computing".
    - **Modern Computers:** $\sim\ 500kT\log_e 2$ Joules/Bit.
    - **DNA Computer:** $\sim 1.5 \cdot 10^{-19}\ Joule/operation.$
  - **Reversible Computing:**
    - Is $kT\log_e 2$ Fundamental?
      - This is irreversible(can't go backwards).
      - That means if you had random XY coming in, you have less randomness of the output, so some entropy is changed.

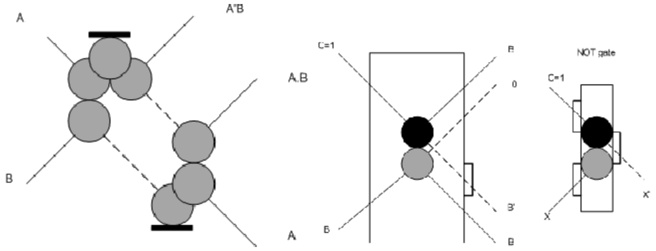
2 - input AND gate





A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

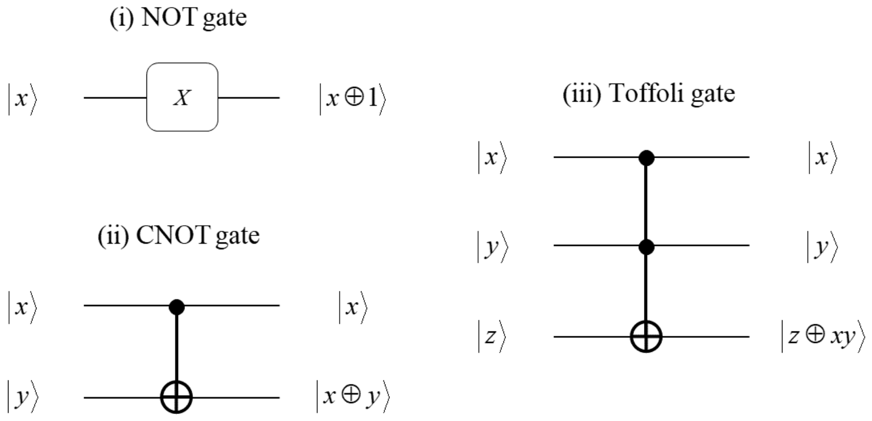
- **Billiard ball Model(Mechanical model of computation:)**
  We formed a gate by using the Model. The Model was not adapated because any small error can lead to bigger and bigger error.
  r0.5



Fredkin and Toffoli → In a paper ” conservative logic” [FT82]
fun fact:

- Ed Fredkin → only person who's become a faculty at MIT with No PhD, No Bachelor's, No High school Diploma.

- **Reversible Gate:**



Theorem 1. *Toffoli gate is Universal:*

Theorem 2. *Fred Universal*

Theorem 3. *Amount of garbage needed to reversibly simulate any boolean circuit ∼ O(width of circuit) and* **Not depth!**

Theorem 4. *Any Boolean circuit of N gates can be simulated by a reversible circuit of O(polynomial(n)) new gates.*

1.2 Mon, Sept 9: Review of Newtonian Mechanics

- A *Newtonian trajectory* $\mathbf{x}(t)$ $(t \in \mathbb{R})$ is given by solutions of the second order ODE

  $$m\,\ddot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)),$$

  where $m > 0$ is a basic parameter associated with a given Newtonian particle, called its *mass*.

- The force field $\mathbf{F}(\mathbf{x})$ — which we take to be static (i.e., not intrinsically dependent on time) for simplicity — is said to be *conservative* if there is a *potential function* $V(\mathbf{x})$ such that

  $$\mathbf{F}(\mathbf{x}) = -\nabla V(\mathbf{x}).$$

Here, '∇' denotes the *gradient operator*,

$$\nabla V = \left(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z}\right).$$

- For a conservative force field, we can find a *conserved quantity* along the Newtonian trajectories, namely the *total mechanical energy.*

  $$E = H(\mathbf{x}, \mathbf{p}) := \frac{1}{2m}\,\mathbf{p}^2 + V(\mathbf{x}).$$

  Here, $\mathbf{p}^2 := \mathbf{p} \cdot \mathbf{p} = \|\mathbf{p}\|^2$, and $\mathbf{p} := m\mathbf{v} := m\dot{\mathbf{x}}$ is the *momentum.*

1.3 Tue, Sept 10: Alternative Formulations of Newtonian Mechanics

- The **Hamiltonian formulation**:

  $$\dot{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{p}}, \qquad \dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{x}}.$$

- The **Lagrangian formulation**:

  $$\delta S[\mathbf{x}(t)] = 0,$$

  where the *action* on the time interval $[t_a, t_b]$ is given by

  $$S[\mathbf{x}(t)] := \int_{t_a}^{t_b} \left[\frac{m}{2}\,\dot{\mathbf{x}}(t)^2 - V(\mathbf{x}(t))\right] dt.$$

- Etc.

Definition 1. *The* Name of the term here... *is given by*

$$MathEquationhere..$$

2 Section 2 Name here..

References

[FT82] Edward Fredkin and Tommaso Toffoli. "Conservative logic". In: *International Journal of Theoretical Physics* 21.3 (Apr. 1982), pp. 219–253. ISSN: 1572-9575. DOI: 10.1007/BF01857727. URL: https://doi.org/10.1007/BF01857727.