Foundations of Quantum Information

Course Short Content

Introduction

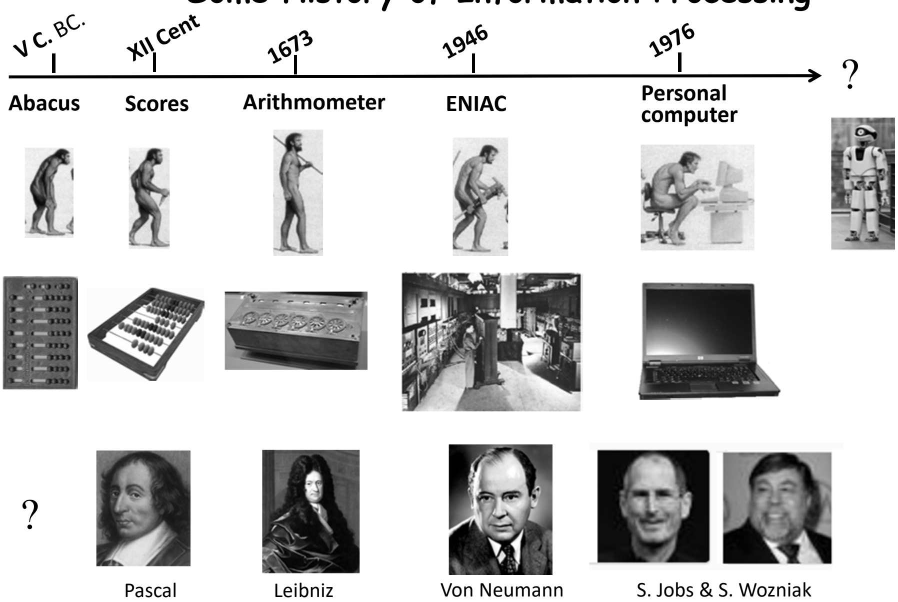
1. Classical approach to Information and Computational Complexity,
2. Quantum Approach to Information,
3. Quantum Information With Discrete Variables,
4. Quantum Communication and Algorithms,
5. Quantum hardware,
6. Quantum information with continuous variables.

Basic Literature

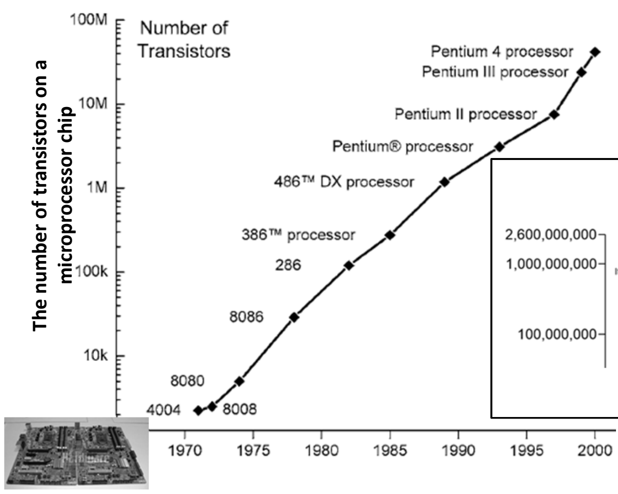
1. Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information Cambridge University Press, 2000
2. Christopher Gerry, Peter Knight, Introductory Quantum Optics, Cambridge University Press, 2005
3. L. Mandel and E. Wolf, Optical Coherence and Quantum Optics, Cambridge University Press, 1995.
4. D. F. Walls, Gerard J. Milburn, Quantum Optics Springer Science & Business Media, 2008.
5. S. M. Barnett and P. M. Radmore, *Methods in Theoretical Quantum Optics,* Oxford, 1997
6. Christopher Gerry, Peter Knight, Introductory Quantum Optics, Cambridge University Press, 2005
7. Anthony Sudbery. Quantum Mechanics and the Particles of Nature: An Outline for Mathematicians Cambridge University Press 1986
8. EMMANUEL DESURVIRE, Classical and Quantum Information Theory. An Introduction for the Telecom Scientist, Cambridge University Press 2009

+ a lot of papers, reviews, etc.

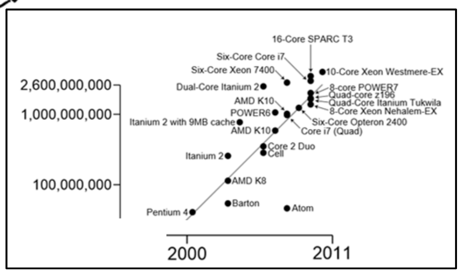
Some History of Information Processing



Moor's Law



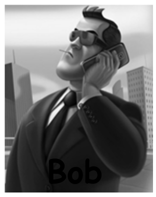
G. Moore
Intel Co



Big Data Problem

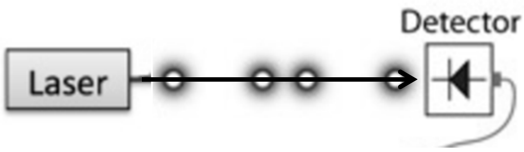
Ahmed K. Noor, Open Eng. 2015; 5:75–88

What We Recognize as Quantum Technologies



- Quantum Cryptography — **Protection of communications, based on the principles of quantum physics**
- Quantum Communication — **Information transfer on the principles of qu-antum physics, e.g. Quantum teleportation**
- Quantum Metrology — **Measurement of physical quantities beyond standard quantum limit, e.g. registration of gravitational waves**
- Quantum Computing — **Universal computing which uses quantum phenomena for data processing.**
- Quantum Simulators — **Computing device for solving only one problem and uses the phenomena of quantum physics.**
- Quantum Machine Learning — **Exploits quantum algorithms for AI purposes and vise versa**

Classical probabilities and distributions



Two Theories of Probability

- Classical (**Kolmogorovian**) prob.theory;
- Quantum theory

1930, **Kolmogorov**:
Economics, finance, statiscis …

A.N. Kolmogorov: "Grund begriffe der Wahrscheinlichkeitsrechnung", Springer, Berlin , 1933.

CLASSICAL THEORY

Andrei Kolmogorov (1933)

QUANTUM THEORY

John von Neumann (1932)

Von Neumann, Mathematical Foundations of Quantum Mechanics, 1932

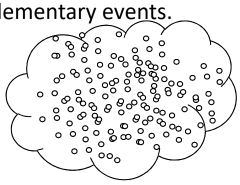
1920 – Bohr, Heisenberg, Schrodinger, et al,
Quantum mechanics

1930 – von Newmann, -
Quantum logic,
quantum measurement,
quantum probabilities

1950, Feynman,
Quantum computing,
1990Ekert, Brassard , et al
1991 Quantum information

Classical probability

Kolmogorov approach: representation of random events by subsets of some basic set. This set is considered as sample space - the collection of all possible realizations of some experiment. Points of sample space are called elementary events.



$$\Omega = \{\omega_1, \ldots, \omega_N\}$$

Example. n- time coin tossing

We have vector $\omega = \{x_1, \ldots, x_n\}$, where

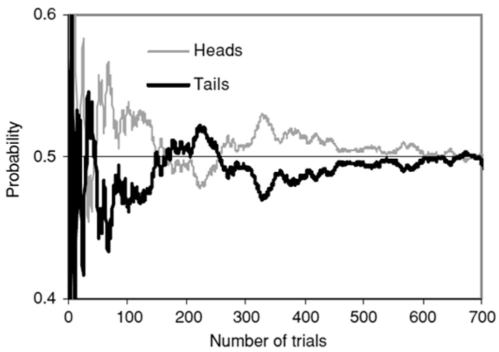
$x_j =$ **«*Heads*» , «Tails». Set Ω contains 2^n points**

For each event A one possible to match some probability of measurement . $A \mapsto P(A)$

$$\begin{cases} p(\text{heads}) = \dfrac{\text{number of heads counts}}{\text{number of trials}} \\ p(\text{tails}) = \dfrac{\text{number of tails counts}}{\text{number of trials}}. \end{cases} \qquad 0 \le P(\omega_j) \le 1, \ \sum_j P(\omega_j) = 1.$$

According to Kolmogorov, a set of events on which the probabilities are determined is very large

Experimental determination of probabilities for coin flipping by means of 700 successive trials



Q. What is the probability that a flipped coin lands three times on the same side?

Answer: The probability of getting either three tails or three heads is q = $(1/2)^3$ = 0.125. But ! Since either succession of events is possible (i.e., getting three heads or three tails), the total probability is actually 0.125 + 0.125 = 0.25 = 1/4.

Classical probability distribution (PDF)

Consider then the *discrete* case, as defined by the sample/event space $S = \{x_1, x_2, \ldots, x_N\}$, where N can be infinite (notation being $N \to \infty$). The associated PDF is called $p(x)$, which is a function of the random variable x, which takes the discrete values x_i ($i = 1, \ldots, N$). When writing $p(x_i)$, this conceptually means "the probability that event x takes the value x_i." The *mean*, which is noted $<x>$, or also $\bar{x}$, or $E(x)$, is given by the weighted sum

$$<x> = \sum_{i=1}^{N} x_i p(x_i).$$

Q. Lets consider rolling dice problem. What is $p(x_i)$. ? Pls, calculate $<x>$

Dice

Answer

As an illustration, take the event space $S = \{1, 2, 3, 4, 5, 6\}$ corresponding to the outcomes of rolling a single die. As we know, the PDF is $p(x) = 1/6$ for all events x. The mean value is, therefore,

$$<x> = \sum_{i=1}^{6} x_i p(x_i) = \sum_{i=1}^{6} i \frac{1}{6} = \frac{1}{6}(1+2+3+4+5+6) = 3.5.$$

PDF variance $\sigma^2 = <(x - <x>)^2>$

Q. Pls, calculate $\sigma^2 = <(x - <x>)^2>$

Answer $\sigma^2 = <(x - <x>)^2> = <x^2 - 2x<x> + <x>^2>$

$$= <x^2> - 2<x><x> + <<x>^2> = <x^2> - 2<x>^2 + <x>^2 = \boxed{<x^2> - <x>^2}$$

PDF standard deviation

$$\sigma = \sqrt{\sigma^2} \equiv \sqrt{<x^2> - <x>^2}.$$

Q. Pls, calculate σ^2 for one-dice case

Calculus $<x^2> = \sum_{i=1}^{6} i^2 \frac{1}{6} = \frac{1}{6}(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) = 15.166.$

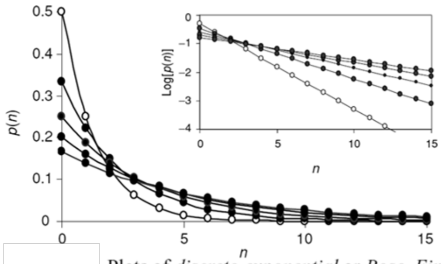
$$\sigma^2 = <x^2> - <x>^2 = 15.166 - 3.5^2 = 2.916.$$

PDE distributions

Discrete-exponential, or, Bose–Einstein (BE) distribution

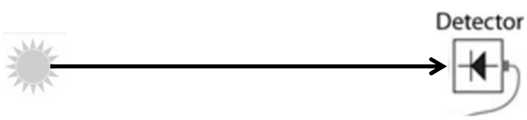
$$p(n) = \frac{1}{N+1}\left(\frac{N}{N+1}\right)^n$$

where $<n> = N$ is the mean value. This PDF variance is simply $\sigma^2 = N + N^2$



$$N = \frac{1}{e^{\hbar\omega/k_BT} - 1}$$

Plots of *discrete-exponential* or *Bose–Einstein* probability distribution corresponding to mean values $<n> = N = 1, 2, 3, 4, 5$ (open symbols corresponding to the case $<n> = 1$). The inset shows the same plots in decimal logarithmic scale.

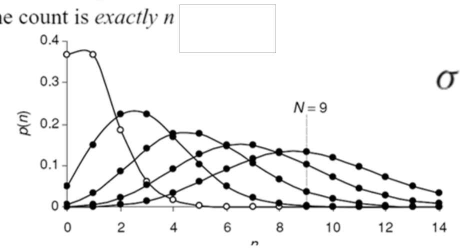


PDE distributions

Poisson Distribution

$$p(n) = e^{-N}\frac{N^n}{n!}$$

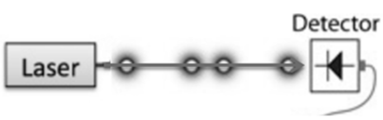
PDF is used to predict the number of occurrences of a discrete event over a fixed time interval. If N is the expected number of occurrences over that time interval, the probability that the count is *exactly n*



$$\sigma^2 = N$$

Plots of the *Poisson* distribution corresponding to mean values $<n> = N = 1, 3, 5, 7, 9$ (open symbols corresponding to the case $<n> = 1$).

In laser physics, the Poisson PDF corresponds to the count of photons emitted by a coherent light source, or laser.



•

Continuous Distributions

Lets consider $p(x)$ as ***continuous* probability distribution**. It is consider the possibility that the events form a continuous and infinite suite of real numbers, which, in the physical world, represent the unbounded set of measurements of a physical quantity.

$$0 \leq p(x) \leq 1$$

for all values x belonging to the event space $\lfloor x_{\min}, x_{\max} \rfloor$

Properties

$$\int_{x=x_{\min}}^{x=x_{\max}} p(x)\mathrm{d}x = 1$$

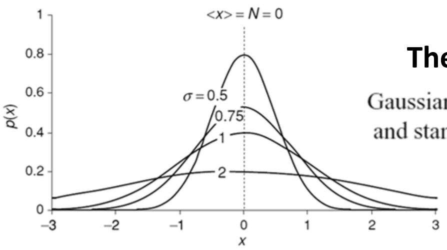
$$<x> = \int_{x=x_{\min}}^{x=x_{\max}} x p(x)\mathrm{d}x \qquad <x^2> = \int_{x=x_{\min}}^{x=x_{\max}} x^2 p(x)\mathrm{d}x,$$

•

PDE distributions

Gaussian Distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-N)^2}{2\sigma^2}\right]$$



The mean value is $<x> = N$

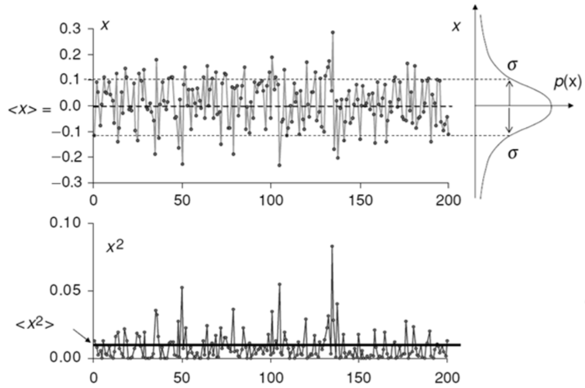
Gaussian probability distribution with mean $<x> = N = 0$ and standard deviations $\sigma = 0.5, 0.75, 1$, and 2.

Applications

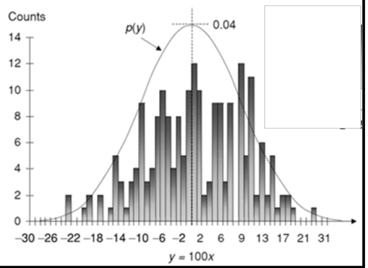
- ***Experimental measurement errors (the mean <x> being taken as the value to be retained);***
- ***Photonics, to approximate the transverse or spatial distribution of light intensity in optical fibers or in laser beams;***
- ***Information theory, when analyzing continuous channels with noise.***
- ***Telecommunications, in the distribution of 1/0 bit errors in digital receivers;***
- ***Education and training, in the distribution of intelligence (IQ) test scores, or professional qualifications and performance ratings;***

•

Example of discrete samplings (200 events) of a Gaussian distribution (<*x*> = 0, *σ* = 0.1), showing the outcome for random variables



Histogram of the 200 sampling points x as converted into $y = 100x$, and corresponding Gaussian distribution envelope $p(y)$.



•

The Entropy

What is a bit?

Bit is binary digit defined as the information entropy of a ***binary random variable*** that is «0» (false) or «1» (true) with equal probability

In 1948 Claude Shannon uses «bit» (от binary digit) for definition of minimal unit of information

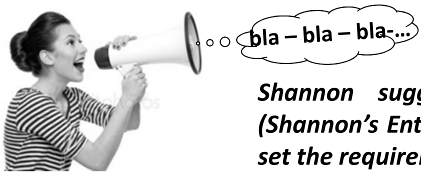
Measuring information

Lets define the amount of information, which we could use as an objective measure reference ***I*(*a*)** for each possible event ***a***. We may postulate that ***I*(*x*)** should approach zero for events close to absolute certainty (***P*(*a*) → 1**), and infinity for events reaching impossibility (***P*(*a*) → 0**).

$$\boxed{i(a) = -\log_2[\mathrm{P(a)}] = \log_2 \frac{1}{\mathrm{P(a)}}}$$

The information that one gets from tossing a coin is $\boxed{i(0.5) = \log_2(2) = 1}$

The property **I = 1** *bit means that the message used to transmit the information only requires a single symbol, out of a source of* $2^I = 2$ *possible symbols*



Shannon's Entropy

Shannon suggested that the increase in information (Shannon's Entropy* H*) is equal to the lost uncertainty, and set the requirements for its measurement.

Assuming a random source with an event space, X, comprising N elements or symbols with probabilities p_i $(i = 1 \ldots N)$, the unknown function H should meet three conditions:

(1) $H = H(p_1, p_2, \ldots, p_N)$ is a continuous function of the probability set p_i;
(2) If all probabilities were equal (namely, $p_i = 1/N$), the function H should be monotonously increasing with N;
(3) If any occurrence breaks down into two successive possibilities, the original H should break down into a weighed sum of the corresponding individual values of *H*

Definition

$$H = -\sum_{i=1}^{N} p_i \log p_i.$$ **For discrete PDF**

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \equiv \sum_{x \in X} p(x) I(x)$$ **For contunuos PDF**

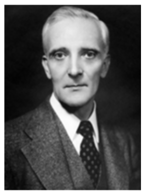
where x is a symbol from the source X and $I(x)$ is the associated information measure

Shannon's Entropy

The entropy of a source is the average amount of information per source symbol

$$H = \langle I \rangle = -\langle \log p \rangle$$

where ***I*** **(*x*)** is the *information* measure associated with a symbol ***x*** of probability ***p*(*x*)**.



Léon Brillouin

If all symbols are equiprobable ($p(x) = 1/N$), the source entropy is given by

$$H = -\sum_{x \in X} p(x) \log p(x) = -\sum_{i=1}^{N} \frac{1}{N} \log \frac{1}{N} \equiv \log N,$$

which is equal to the information $I = \log N$ of all individual symbols.

The $N = 2^q$ equiprobable symbols can be represented by $\log_2 2^q = q$ bits. It means that all symbols from this source are made of exactly q bits, $H = q$ and $I = q$,

Entropy in Dice

Q. How much information we need for dice outcomes?

__Answer.__ For a single die roll, the six outcomes are equiprobable with probability $p(x) = 1/6$

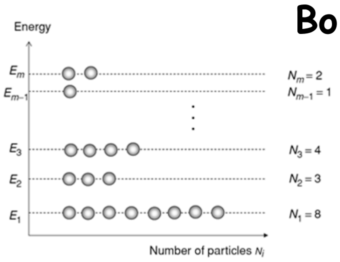
Source entropy is

$$H = -\sum_{i=1}^{6} \frac{1}{6} \log \frac{1}{6} \equiv \log 6 = 2.584 \text{ bit/symbol}.$$

We also have for the information: $I = \log 6 = 2.584$ bits. This result means that it takes 3 bits (as the nearest upper integer) to describe any of the die-roll outcomes with the same symbol length, i.e., in binary representation:

$$x = 1 \rightarrow 100 \quad x = 2 \rightarrow 010 \quad x = 3 \rightarrow 110$$
$$x = 4 \rightarrow 001, \quad x = 5 \rightarrow 101, \quad x = 6 \rightarrow 011.$$

Boltzmann Entropy



Energy-level diagram showing how a set of *N* identical **non-interacting** particles in a physical macro-system can be distributed to occupy, by subsets of number N_i , different microstates of energy $E_i \ (i = 1 \dots m)$. $N = \sum_{i=1}^{m} N_i$.

Now let's perform some combinatorics. We have N particles each with m possible energy states, with each state having a population N_i. The number of ways W to arrange the N particles into these m boxes of populations N_i is given by:

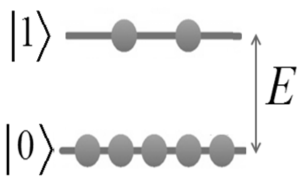
$$W = \frac{N!}{N_1! N_2! \dots N_m!}$$

Pls, proof this formula!

Boltzmann H-theorem is $H = \lim_{N\to\infty} \frac{\log W}{N} = -\sum_{i=1}^{m} p_i \log p_i \qquad dH = 0$

where $\boxed{p_i = \frac{N_i}{N} = \frac{e^{-E_i/k_BT}}{\sum_i e^{-E_i/k_BT}}}$ is the probability of finding the particle in the microstate of energy E_i.

Particle distribution for two-level system



$$\boxed{\frac{N_0}{N} = \frac{1}{1+e^{-E/k_BT}},}$$

$$\boxed{\frac{N_1}{N} = \frac{e^{-E/k_BT}}{1+e^{-E/k_BT}}.}$$

$$\frac{p_1}{p_0} = \frac{N_1}{N_0} = \frac{e^{-E_1/k_BT}}{e^{-E_0/k_BT}} = e^{-(E_1-E_0)/k_BT} \equiv e^{-E/k_BT}$$

Maximum Entropy Principle

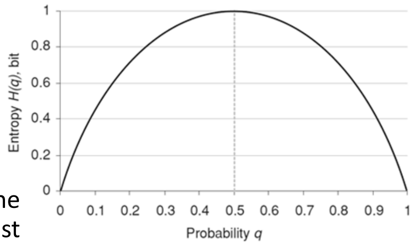
We know that entropy is the measure of the average information concerning a set of events . Can it be maximized, and to which event would the maximum correspond?

An example. *We have seen that the information related to any event **x** having probability p(x) is defined as* I(x) = −log p(x). ***Thus, information increases as the probability decreases or as the event becomes less likely.*** *The information eventually becomes infinite (*I (x)→∞*) in the limit where the event becomes "impossible" (*p(x) → 0*).*

An information is unbounded, but its infinite limit is reached only for impossible events that cannot be observed

Assume first two complementary events x_1, x_2 with probabilities $p(x_1) = q$ and $p(x_2) = 1 - q$, respectively. By definition, the entropy of the source $X = \{x_1, x_2\}$ is

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$
$$= -x_1 \log p(x_1) - x_2 \log p(x_2)$$
$$= -q \log q - (1-q)\log(1-q) \equiv f(q)$$



Principle of maximum entropy

Probability distribution which best represents the current state of system is the one with largest entropy.

Classical computation

Computational Complexity

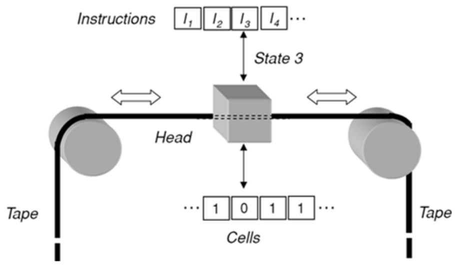
Goal of computational complexity theory.

- To provide a method of quantifying problem difficulty in an absolute sense.
- To provide a method for comparing the relative difficulty of two different problems.
- We would rigorously define the meaning of "efficient" algorithm...
- ...and we would like to state that one algorithm is "better" than another.
- Complexity theory is built on a basic set of assumptions called the model of computation (Turing Machine).

Turing Machine as an Abstract Computer

Turing machine is a mathematical model of computation that defines an abstract machine, which manipulates symbols on a strip of tape according to a table of rules.

Alan Turing



Deterministic Turing machine (DTM), the set of rules prescribes **at most one action** to be performed **for any given situation**. DTM has a *transition function* that, for a given state and symbol under the tape head, specifies three things:

- ✓ ***the symbol to be written to the tape,***
- ✓ ***the direction (left, right or neither) in which the head should move, and***
- ✓ ***the subsequent state of the finite control.***

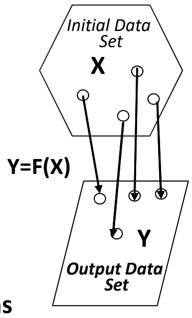
Non-deterministic Turing machine (NTM), the set of rules may prescribe more than one action to be performed for any given situation.

Kurt Gödel

Computability

Computable functions are the formalized analogue of the intuitive notion of **algorithm.**

A function is computable if there exists an algorithm that returns the corresponding output for given an input of the function domain .



There exist computable (recursive) and non-computable functions

Example of non-computable function (algorithm) realization

Halting problem *is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running (i.e., halt) or continue to run forever.*

Church–Turing thesis

Any computable algorithm can be realized by using TM.

Computational Complexity

Time complexity ***describes the amount of time it takes to run an algorithm. Time complexity is estimated by counting the number of elementary operations performed by the algorithm.***

Space complexity ***f*(*n*)** ***is the amount of memory space required to solve an instance of the computational problem as a function of the size of the input n (bits) .***

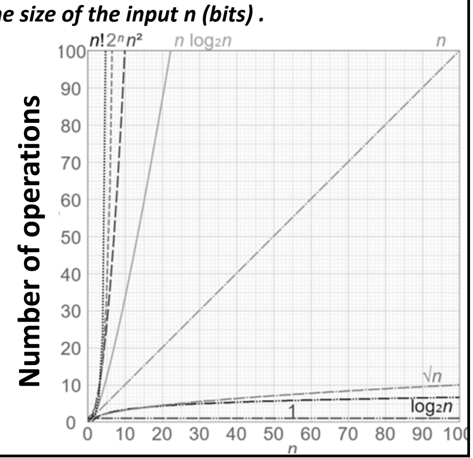
Asymptotic Analysis

Determining the exact function ***f*(*n*),** is still problematic at best.

We will only really be interested in approximately how quickly the function grows "in the limit" of **n**

To determine this, we use asymptotic analysis, aka "big *O* notation":

$$O(n),\, O(n\log n),\, O(n^\alpha),\, O(2^n)$$

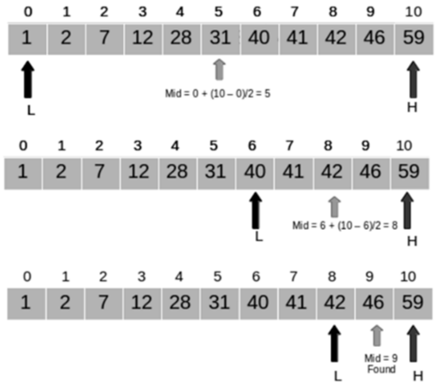


P-Class Computational Complexity

An algorithm is said to be of **Polynomial time, P-class** if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, i.e., **$f(n) = O(n^k)$** for some positive constant ***k***

Example 1: Binary search algorithm ***(The element that we are looking for is the number 46)***

Worst-case performance	$O(\log n)$
Best-case performance	$O(1)$
Average performance	$O(\log n)$

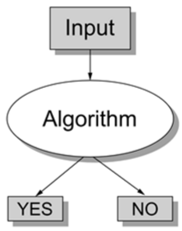


Example 2: Linear programming

Linear programming (**linear optimization**) ***is a method to achieve the best outcome (such as lowest cost) in a mathematical model whose requirements are represented by linear relationships.***

NP-Class Computational Complexity

NP (**nondeterministic polynomial time**) is the set of **decision problems** *solvable* in **polynomial time** by a **NDT machine.**



Decision problem

NP-Complete and NP-Hard Classes

A decision problem L is NP-complete if:

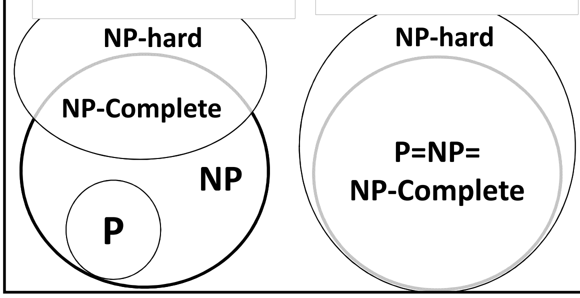
1) Any given solution for NP-complete problems can be verified quickly, but there <u>is no efficient known solution for NP-problem in polynomial time</u>.

2) Every problem in NP is reducible to L in polynomial time.

- **A problem is NP-Hard if it follows property 2 mentioned above, doesn't need to follow property 1.**

Hypothesis

$\mathrm{P} \neq \mathrm{NP}$ Or, $\mathbf{P} = \mathbf{NP}$ **?**



Examples

- ✓ **Travelling salesman problem.**
- ✓ **Factorizing problem,**
- ✓ **Graph coloring problem,**

....

Exponentially hard problems for calculus

What is an Efficient Algorithm?

Is an O(n) algorithm efficient?

How about O(n log n)?

$O(n^2)$?

$O(n^{10})$?

polynomial time

$O(n^c)$ for some constant c

$O(n^{\log n})$?

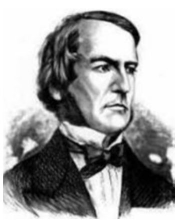
$O(2^n)$?

O(n!) ?

non-polynomial time

Classical gates

Boolean Algebra



George Boole

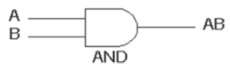
A **Boolean algebra** is a six-tuple consisting of a set *A*, equipped with two binary operations **∧ ("and"), ∨ ("or"),** a unary operation **¬ ("not")** and two elements 0 and 1.

Axiomatics

$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	associativity
$a \vee b = b \vee a$	$a \wedge b = b \wedge a$	commutativity
$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$	absorption
$a \vee 0 = a$	$a \wedge 1 = a$	identity
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	distributivity
$a \vee \neg a = 1$	$a \wedge \neg a = 0$	complements

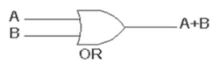
Garrett Birkhoff, 1967. Lattice Theory, 3rd ed. Vol. 25 of AMS Colloquium Publications.

Classical Logic gates



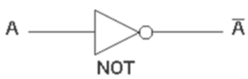
Truth table

2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1



Truth table

2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



Truth table

NOT gate	
A	$\bar{A}$
0	1
1	0

Truth table

2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Real (quantum) device has 2x2 ports

Landauer principle

Rolf Landauer, IBM J. of Res. And Development, 5 (3), 183 (1961)

Any logically irreversible manipulation of information must be accompanied by a corresponding entropy increase $\Delta S = k \ln 2$.

We lost $kT \ln 2$ ***energy (for heating)***

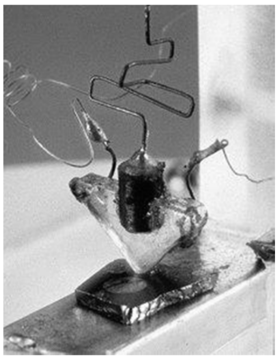
Outlook: Classical gates

- Implement Boolean functions.
- Are not reversible (invertible) due to thermodynamics. We cannot recover the input knowing the output.
- This means that there is an irretrievable loss of information.

First Solid State Transistor

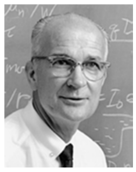
John Bardeen

Bardeen, Brattain & Shockley, 1947



Physical Realization

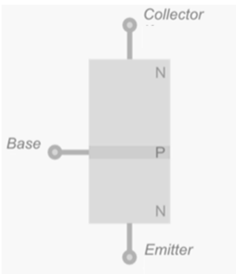
Classical transistors are devices used to amplify or switch electronic (photonic, or, some other) signals and electrical power.



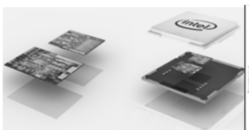
William Shockley

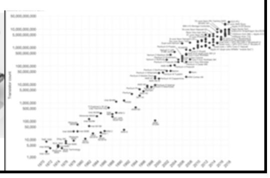
<u>**Example**</u>***: bipolar (semiconductror) transistor***

Physical realization



Currently available processor Intel i5





Transistor size is 22 nm (2015), novel transistors -8 nm

Questions?