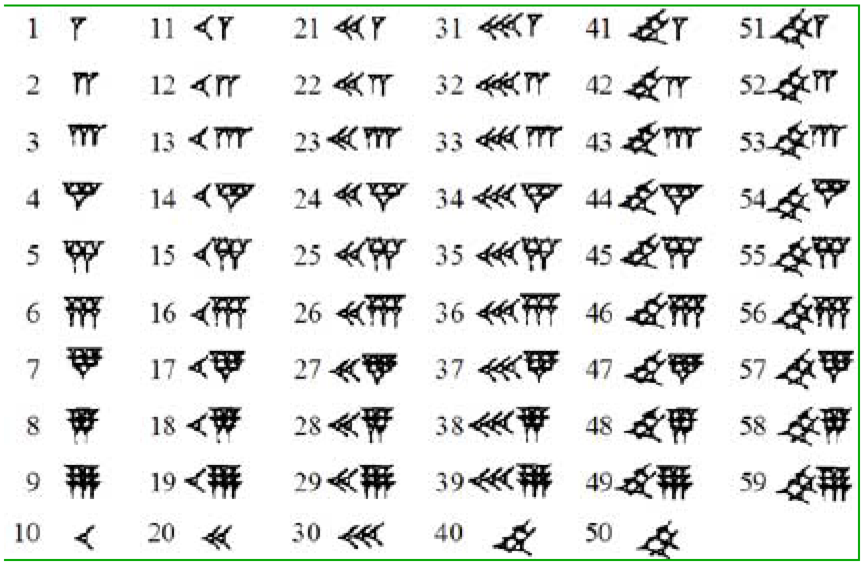
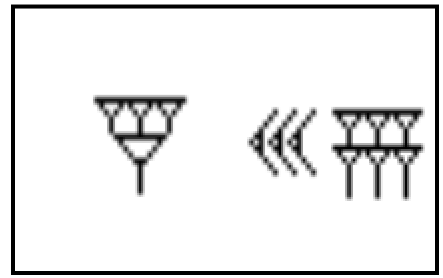
FUNDAMENTALS OF INFORMATION SCIENCE

Shandong University
2025 Spring

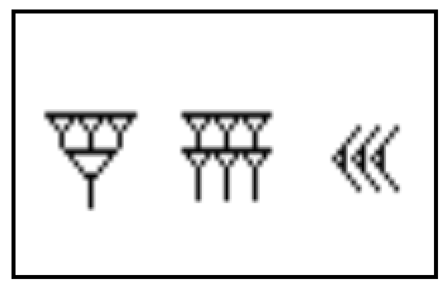
Babylonians

$$N = \sum_{i=0}^{m} d_i 60^i$$

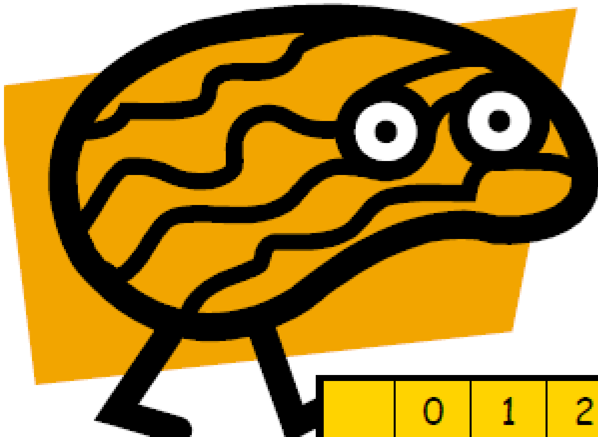


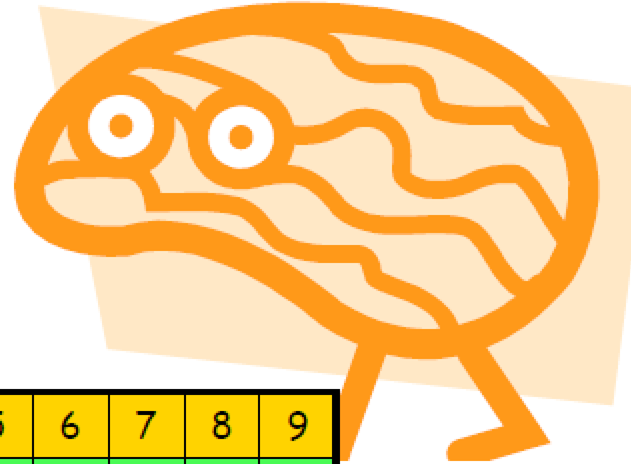


4x60 + **36**x1 = 276



4x3600 + **6**x60 + **30**x1 = 14,790





	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

Gottfried Leibniz
1646-1716

Leibniz - Binary System

Use the smallest syntax possible Binary - 0 and 1

§71 This gives me the opportunity to point out that all numbers could be written by 0 and 1 in the binary or dual progression. Thus:

1	1	10 is equal to 2
10	2	100 is equal to 4
1000	8	1000 is equal to 8
10000	16	etc.
100000	32	
1000000	64	

Gottfried Leibniz
1646-1716

Binary Addition

Addition:

*	110	6
	111	7
	••	
	1101	13

101	5
1011	11
• • • •	
10000	16

1110	14
10001	17
11111	31

carry

Gottfried Leibniz
1646-1716

Binary Multiplication

Multiplication:

	11	3
**	11	3
3	11	
6	11	
	1001	9

Addition of **shifted versions**

Gottfried Leibniz
1646-1716

Leibniz
The Founder of Information Science

Contributed to:

Mathematics	Philosophy
Physics	Politics
Logic	Law
Probability	History
Computing	Library science
...	...

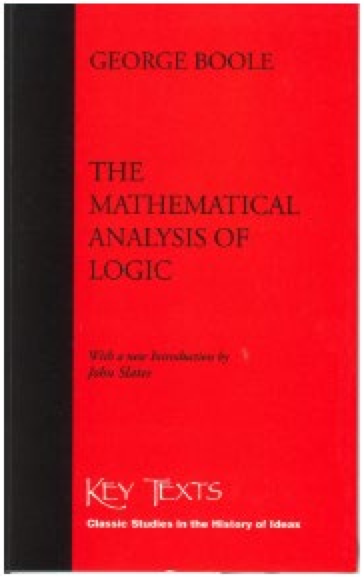
He was **the first** that thought about those four topics as one!

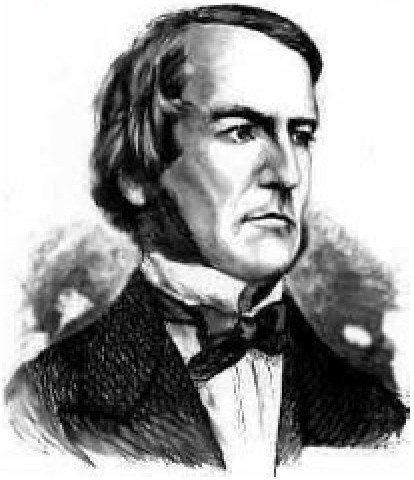
However, his work was recognized **mainly starting 1900**...

~2000 years later...

1847, Algebra of Logic

George Boole
1815 -1864





a "number system" for logic....

The Algebra (Boolean Calculus)

Boole, Jevons, Peirce, Schroder (18xx)

Axiomatic System: Huntington (1904)

Algebraic system: set of elements B,
two binary operations + and ·
B has at least two elements (0 and 1)

If the following axioms are true then it is a **Boolean Algebra**:

$\{a, \bar{a}, b, c\} \in B$

A1. identity $\boxed{a + 0 = a \quad \text{and} \quad a \cdot 1 = a}$

A2. complement $\boxed{a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0}$ ← $\boxed{\forall a \quad \exists \bar{a}}$

A3. commutative $\boxed{a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a}$

A4. distributive
$$\left\{ \begin{array}{l} \boxed{a + b \cdot c = (a + b) \cdot (a + c)} \\ \boxed{a \cdot (b + c) = a \cdot b + a \cdot c} \end{array} \right.$$

Application of the the 0-1 Theorem

Elements:

0-1 vectors of length n, there are 2^n vectors

Operations:

Bitwise OR

Bitwise AND

Bitwise Complement

It is a Boolean algebra with for n=1

By the 0-1 Theorem:

It is a Boolean algebra for any finite n

Examples 'other' Boolean Algebras

- 0-1 vectors
- Algebra of subsets next
- Arithmetic Boolean algebras next

So Far so Good...

- **A1. Identities:**
  $a + 0 = a$ and $a \cdot 1 = a$
- **A2. Complements:**
  $a + \bar{a} = 1$ and $a \cdot \bar{a} = 0$
- **A3. Commutativity:**
  $a + b = b + a$ and $a \cdot b = b \cdot a$
- **A4. Distributivity:**
  $a + (b \cdot c) = (a + b) \cdot (a + c)$ and $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

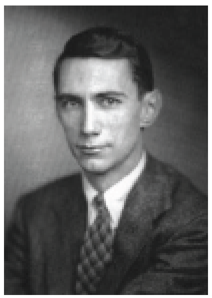
- **L1. Self Absorption:**
  $a + a = a$ and $a \cdot a = a$
- **L2. Simple Absorption:**
  $a + 1 = 1$ and $a \cdot 0 = 0$

- **T3. Associativity:**
  $(a + b) + c = a + (b + c)$
  $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- **T4. DeMorgan Laws:**
  $\overline{(a + b)} = \bar{a} \cdot \bar{b}$
  $\overline{(a \cdot b)} = \bar{a} + \bar{b}$

- **T0. Duality:**
  Correctness is maintained when interchange $+$ and $\cdot$, as well as 0 and 1.
- **T1. Distinct Complement:**
  Every element has another element that is its unique complement.
- **T2. Absorption:**
  $a + ab = a$ and $a \cdot (a + b) = a$

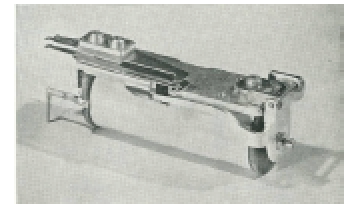
Shannon used **relays** and connected them in **series-parallel circuits**

Shannon
1916-2001

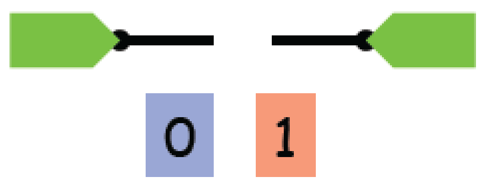


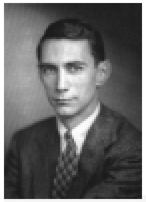
A Symbolic Analysis of Relay and Switching Circuits*

*Claude E. Shannon***



Relay on the edge **controlled** by a **0-1 variable**





A Symbolic Analysis of Relay and Switching Circuits*

synthesis of circuits

In Shannon's words:

"For the synthesis problem the **desired characteristics** are first written as a **system of equations**, and **the equations are then manipulated into the form representing the simplest circuit**. The **circuit may then be immediately drawn from the equations**."

desired characteristic

↓

system of equations

↓

simplified set of equations

↓

simple circuit

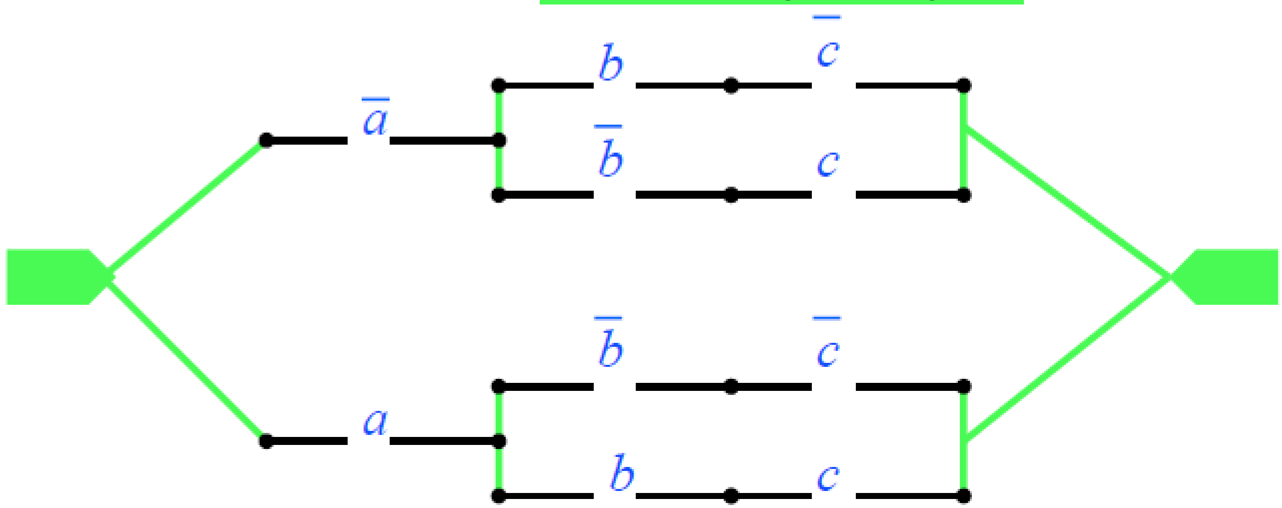
XOR of More Variables

$$XOR(x_1, x_2, \ldots, x_n) = \begin{cases} 0 & \text{if } |X| \text{ (number of 1's in } X\text{) is even} \\ 1 & \text{if } |X| \text{ is odd} \end{cases}$$

$$a \oplus b \oplus c = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$$

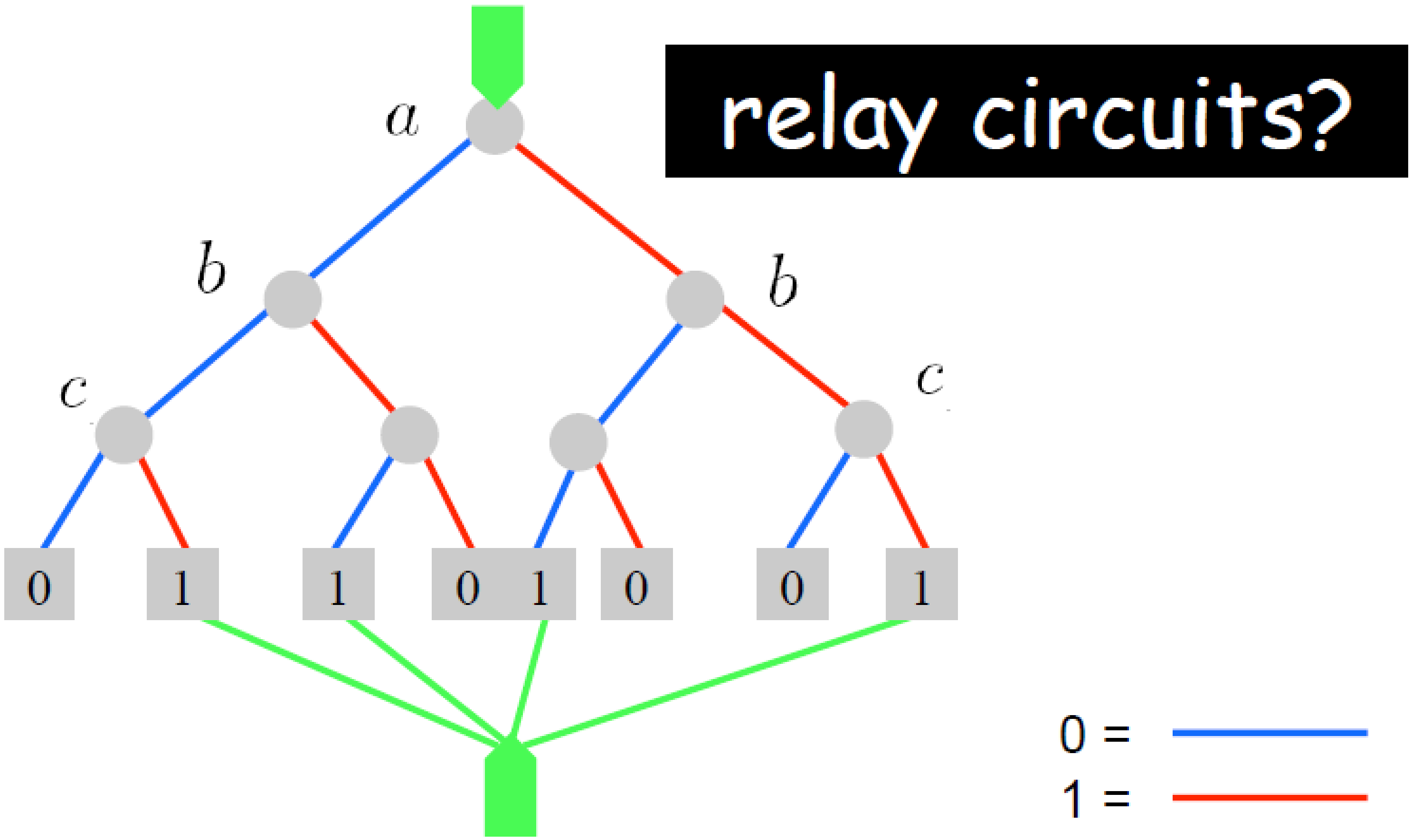
How many relays?

10

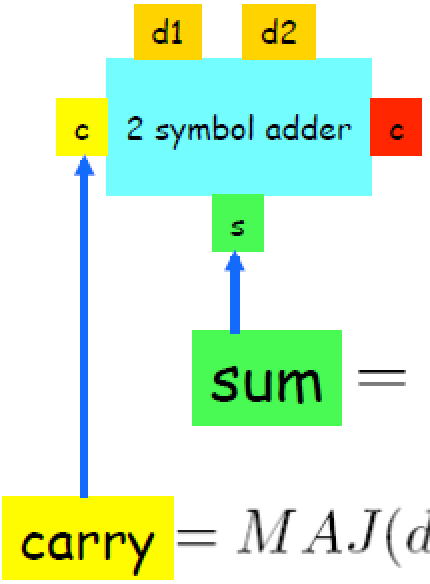


Trees and Relay Circuits

$$a \oplus b \oplus c = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$$



MAJ and *XOR* are **symmetric** Boolean functions



$$\text{sum} = XOR(d_1, d_2, c)$$

$$\text{carry} = MAJ(d_1, d_2, c)$$

Symmetric Functions

Definition: A Boolean function f is symmetric if

$$f(X) = f(\pi(X))$$

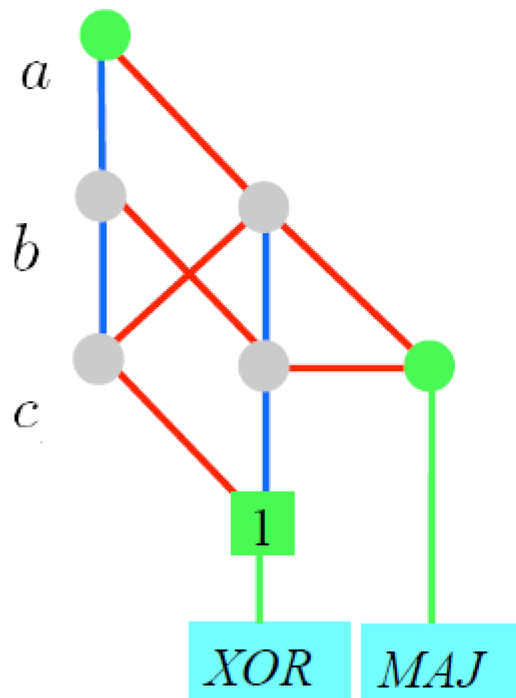
for an arbitrary permutation π

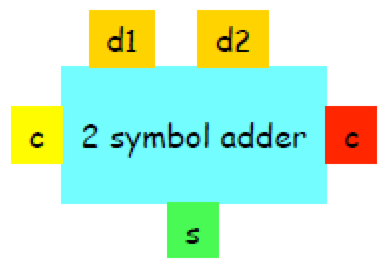
Theorem: A Boolean function $f(X)$ is symmetric if and only if it is a function of the number of 1's in X, namely $|X|$

Relay Circuits for the Sum and the Carry Functions

sum: $a \oplus b \oplus c = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$

carry: $MAJ(a,b,c) = a \cdot b + a \cdot c + b \cdot c$





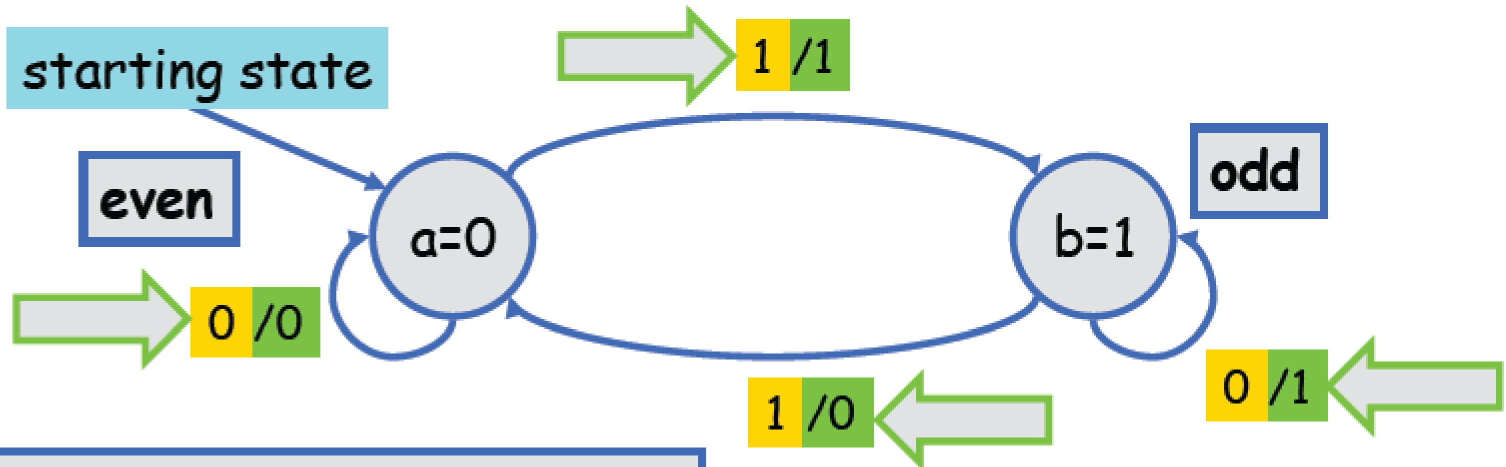
How many relays for a 2 symbol adder?

10

State Machines

synthesis

State Machine for XOR



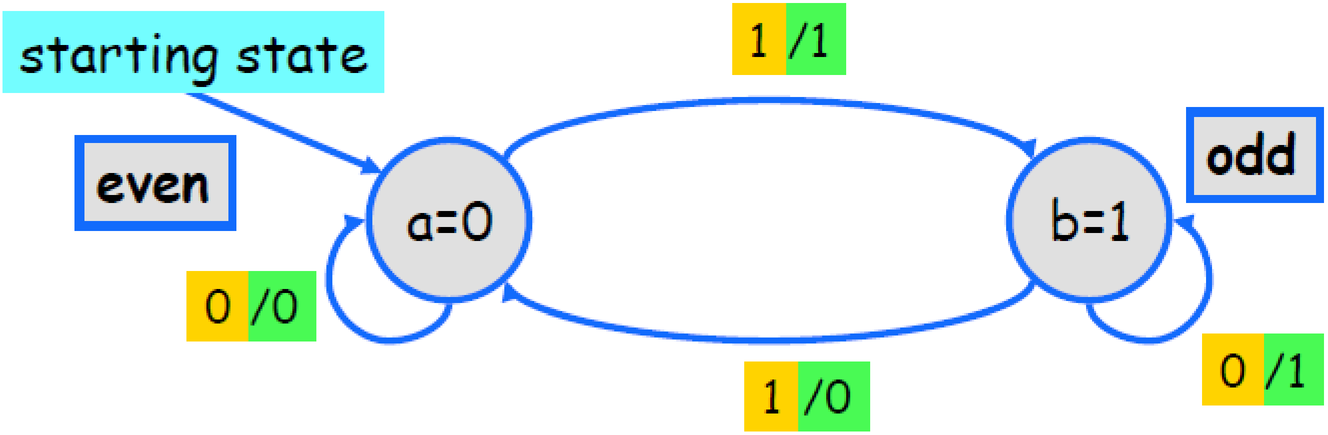
Computing the next state?

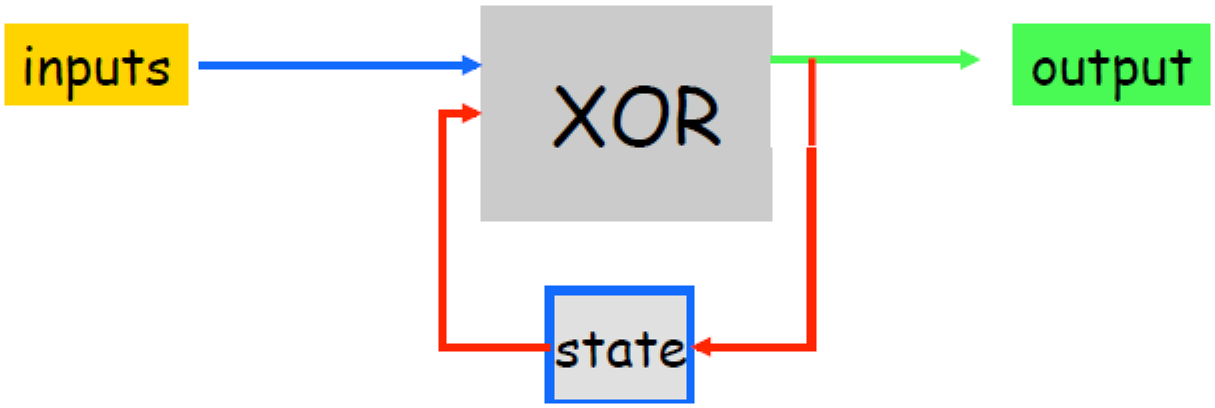
current state \ inputs	0	1
0	0	1
1	1	0

What is the function?

XOR

State Machine for XOR





Equal number of red and blue balls?

Can we compute / recognize any sequence?

NO

Finite state is a limitation...

It is also our is a limitation...

Turing Machines

Alan Turing (1912-1954)

The Turing Machine

- A Turing machine consists of three parts:
  - A **finite-state control** that issues commands,
  - an **infinite tape** for input and scratch space, and
  - a **tape head** that can read and write a single tape cell.
- At each step, the Turing machine
  - writes a symbol to the tape cell under the tape head,
  - changes state, and
  - moves the tape head to the left or to the right.

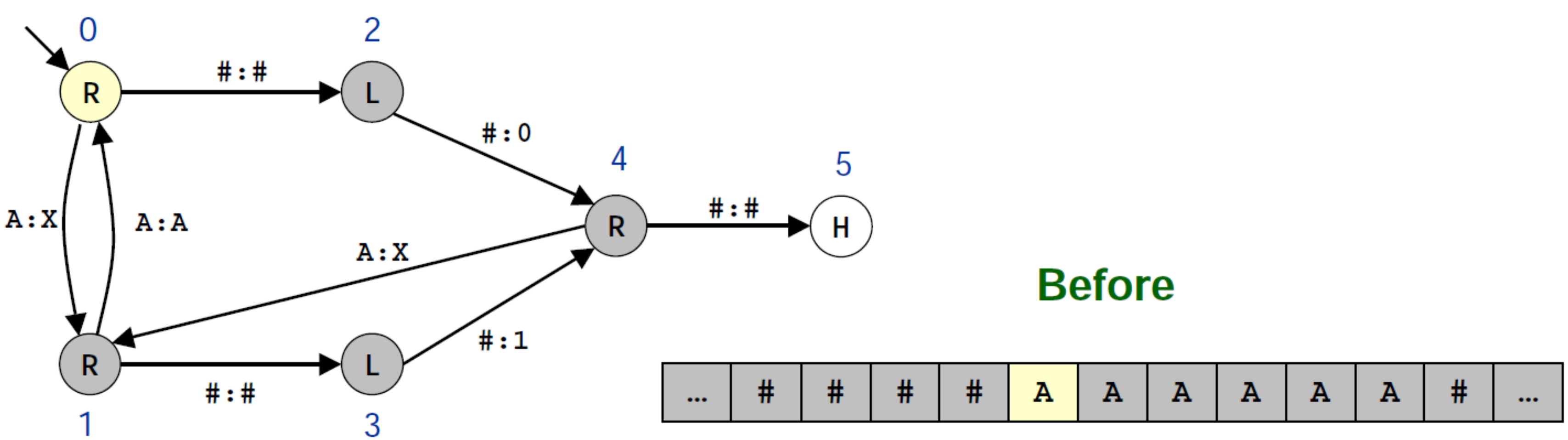
Fetch, Execute

States.

- Finite number of possible machine configurations.
- Determines what the machine does to the active cell and in which way the tape head moves afterwards.

State transition diagram.

- Ex. If Turing machine is in state 0 and the input symbol is `A` then:
  - overwrite the `A` with `X`
  - move to state 1
  - move tape head to right



C Program to Simulate Turing Machine

Three character alphabet (0 is 'blank').

Position on tape.

- `head`

Input: description of machine (9 integers per state s).

- `next[i][s] = t` : if currently in state s and input character read in is i, then transition to state t.
- `out[i][s] = w` : if currently in state s and input character read in is i, then write w to current tape position.
- `move[i][s] = ±1` : if currently in state s and input character is i, then move head one position to left or right.
- `tape[i]` is i^{th} character on tape initially.

Details missing:

- Might run off end of tape.

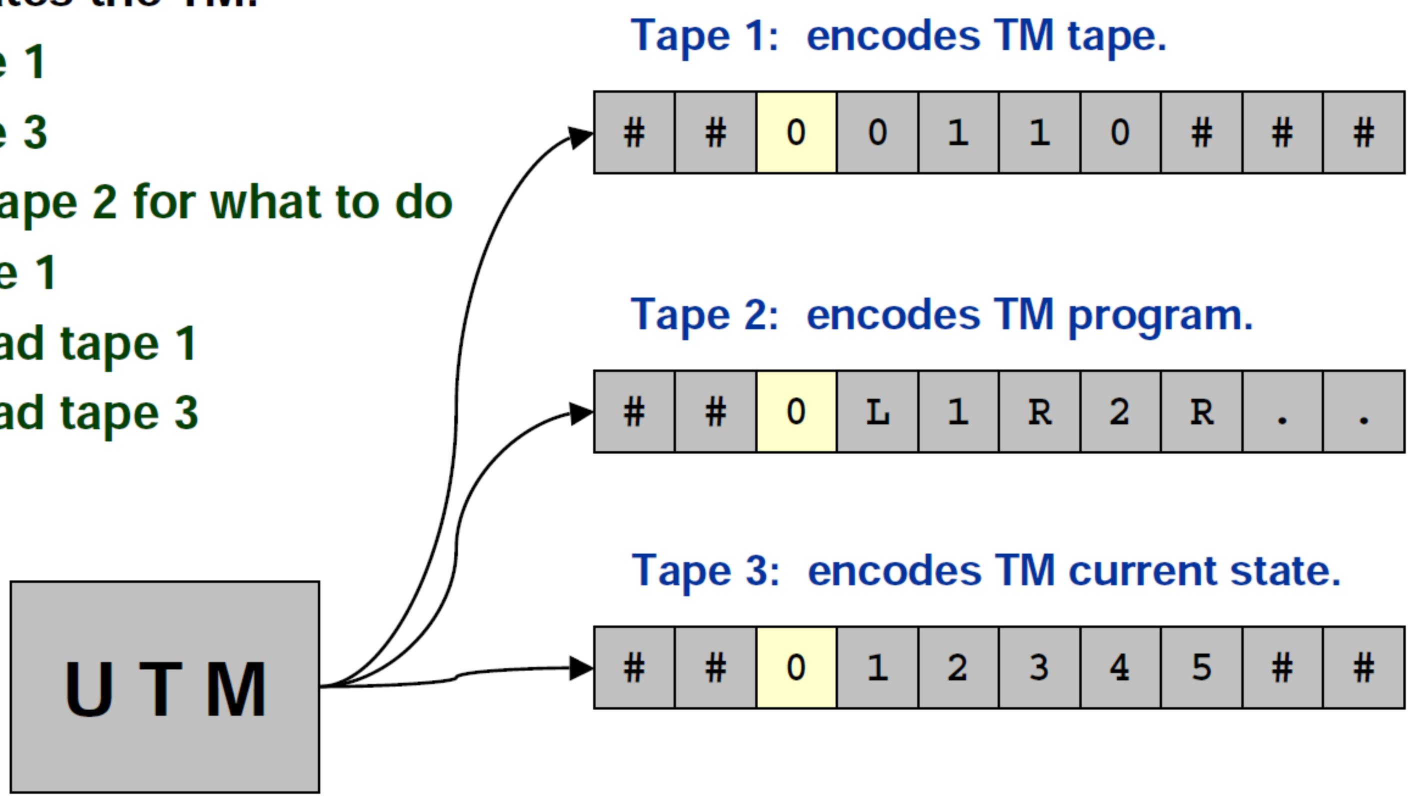
Universal Turing Machine

UTM.

- A specific TM that simulates operations of any TM.

How to create.

- Encode 3 ingredients of TM using 3 tapes.
- UTM simulates the TM.
  - read tape 1
  - read tape 3
  - consult tape 2 for what to do
  - write tape 1
  - move head tape 1
  - move head tape 3



Turing machine is foundation of all modern computers.

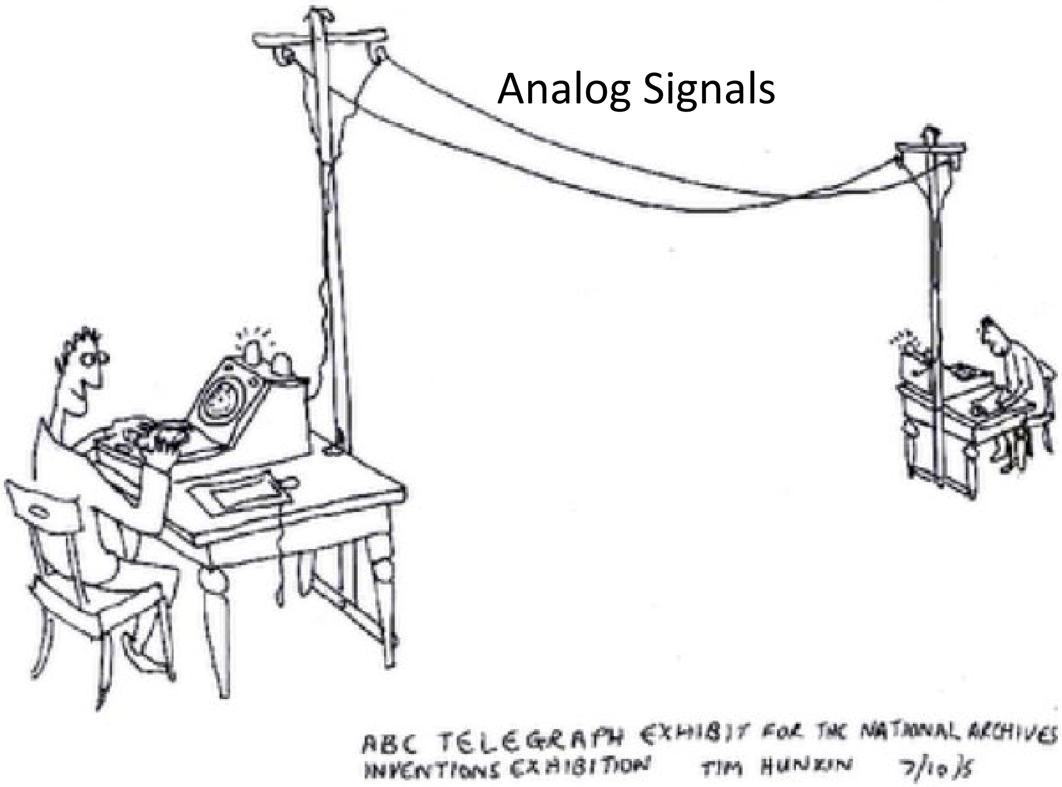
- Simple computational model is easier to analyze.
- Leads to deeper understanding of computation.

Goal: simplest machine "as powerful" as conventional computers.

- TM = software.
- UTM = general purpose computer.

Lecture 2.1: Bits and Entropy

Communication System



Introduce a lot of noise

↓

Digital Communication

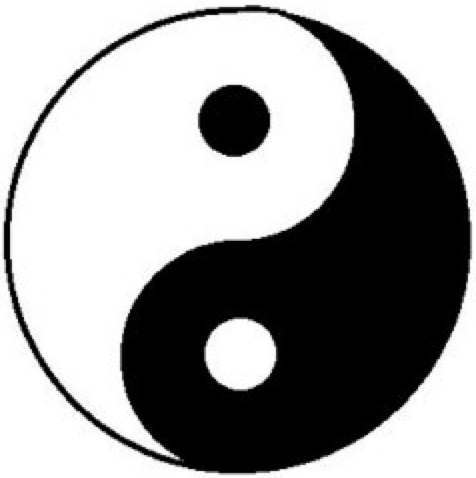
↓

What is the limit?

How to achieve the limit?

Bits

Information is measured in bits (0 and 1)
How is information quantified?



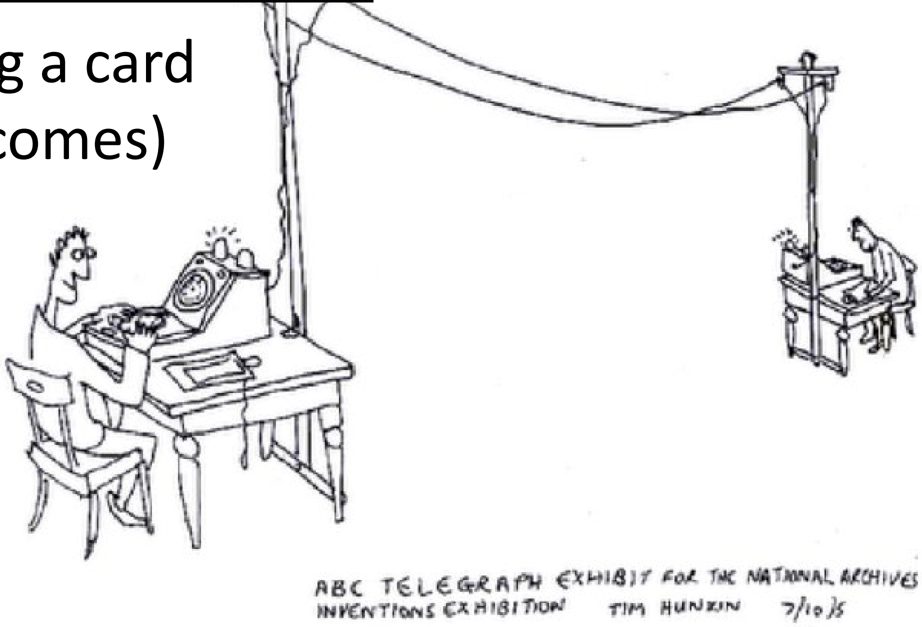
Small information content

Large information content

How is information qualified?

flipping a coin
(2 outcomes)

selecting a card
(52 outcomes)



How compactly tell the outcome of such an experiment?

Which has more information?

I am telling you:

1. “temperature is 40 today”

2. “temperature is 5 today”

Coin tosses with probability 99% (head)

1. The outcome is head

2. The out come is tail

How is information qualified?

Hartley proposed the following definition of the information associated with an event whose probability of occurrence is p

$$I \equiv \log(1/p) = -\log(p).$$

Following Shannon's convention, *we will use base 2*,1 in which case the unit of information is called a **bit**.

Why use the logarithm?

How is information qualified?

One reason of using the logarithm is **additivity**.

Event A = It rained in Qingdao yesterday
Event B = 3 girls in our class
If Event A and B are independent:

Additivity: I(A, B) = I(A) + I(B)

The logarithmic definition provides us with the desired additivity because

$$I_A + I_B = \log(1/p_A) + \log(1/p_B) = \log\frac{1}{p_A p_B} = \log\frac{1}{P(A \text{ and } B)}.$$

How is information qualified?

1 bit

$\log_2(52)$ bits

a randomly chosen decimal digit is even

Amount of information = log2(10/5) = 1 bit.

Definition of Entropy

Information of a random variable X?

Let X be a random variable taking on a finite number M of different values $x_1, \cdots, x_M$

With probability $p_1, \cdots, p_M$, $p_i > 0$, $\sum_{i=1}^{M} p_i = 1$

Information of X = Expected Information of All Outcomes (Entropy)

$$H(p_1, \cdots, p_M) = -\sum_{i=1}^{M} p_i \log_2 p_i$$

Entropy

- Uncertainty in a single random variable
- Can also be written as:

$$H(X) = \mathbb{E}\left\{\log \frac{1}{p(X)}\right\}$$

- Intuition: $H = \log(\#\text{of outcomes/states})$
- Entropy is a functional of $p(x)$
- Entropy is a lower bound on the number of bits need to represent a RV. E.g.: a RV that that has uniform distribution over 32 outcomes

Entropy

- $H(X) \geq 0$

- Definition, for Bernoulli random variable, $X = 1$ w.p. p, $X = 0$ w.p. $1 - p$

$$H(p) = -p \log p - (1 - p) \log(1 - p)$$

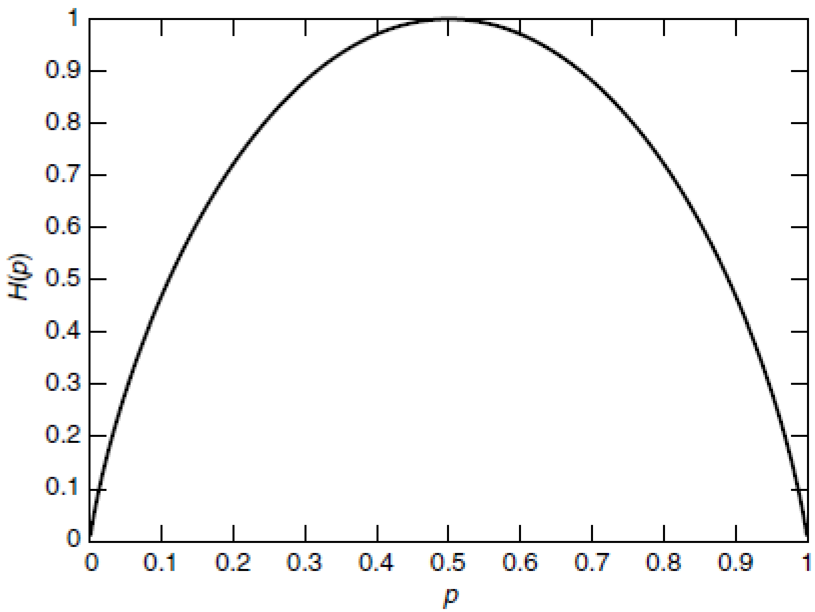


FIGURE 2.1. $H(p)$ vs. p.

- Concave
- Maximizes at $p = 1/2$

Example

Example 2.1.2 Let

$$X = \begin{cases} a & \text{with probability} \frac{1}{2}, \\ b & \text{with probability} \frac{1}{4}, \\ c & \text{with probability} \frac{1}{8}, \\ d & \text{with probability} \frac{1}{8}. \end{cases} \tag{2.6}$$

The entropy of X is

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} = \frac{7}{4} \text{ bits.} \tag{2.7}$$

Joint Entropy

- Extend the notion to a pair of discrete RVs (X, Y)
- Nothing new: can be considered as a single vector-valued RV
- Useful to measure dependence of two random variables

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x,y)$$

$$H(X,Y) = -\mathbb{E}\log p(X,Y)$$

Conditional Entropy

- Conditional entropy: entropy of a RV given another RV. If $(X, Y) \sim p(x, y)$

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X = x)$$

- Various ways of writing this

$$H(Y|X) = -\mathbb{E} \log p(Y|X)$$

Chain Rule for Entropy

Entropy of a pair of RVs = entropy of one + conditional entropy of the other:

$$H(X,Y) = H(X) + H(Y|X)$$

Proof:

- $H(Y|X) \neq H(X|Y)$
- $H(X) - H(X|Y) = H(Y) - H(Y|X)$

Example

Example 2.2.1 Let (X, Y) have the following joint distribution:

$Y \backslash X$	1	2	3	4
1	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$
2	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$
3	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
4	$\frac{1}{4}$	0	0	0

The marginal distribution of X is $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$ and the marginal distribution of Y is $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$, and hence $H(X) = \frac{7}{4}$ bits and $H(Y) = 2$ bits. Also,

$$H(X|Y) = \sum_{i=1}^{4} p(Y = i)H(X|Y = i) \tag{2.22}$$

$$= \frac{1}{4}H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) + \frac{1}{4}H\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}\right)$$
$$+ \frac{1}{4}H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) + \frac{1}{4}H(1, 0, 0, 0) \tag{2.23}$$

$$= \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times 2 + \frac{1}{4} \times 0 \tag{2.24}$$

$$= \frac{11}{8} \text{ bits.} \tag{2.25}$$

Similarly, $H(Y|X) = \frac{13}{8}$ bits and $H(X, Y) = \frac{27}{8}$ bits.

Relative Entropy

- Measure of distance between two distributions

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

- Also known as Kullback-Leibler distance in statistics: expected log-likelihood ratio

- A measure of inefficiency of assuming that distribution is q when the true distribution is p

- If we use distribution is q to construct code, we need $H(p) + D(p||q)$ bits on average to describe the RV

Example 2.3.1 Let $\mathcal{X} = \{0, 1\}$ and consider two distributions p and q on $\mathcal{X}$. Let $p(0) = 1 - r$, $p(1) = r$, and let $q(0) = 1 - s$, $q(1) = s$. Then

$$D(p||q) = (1-r)\log\frac{1-r}{1-s} + r\log\frac{r}{s} \tag{2.31}$$

and

$$D(q||p) = (1-s)\log\frac{1-s}{1-r} + s\log\frac{s}{r}. \tag{2.32}$$

If $r = s$, then $D(p||q) = D(q||p) = 0$. If $r = \frac{1}{2}$, $s = \frac{1}{4}$, we can calculate

$$D(p||q) = \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{3}{4}} + \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{1}{4}} = 1 - \frac{1}{2}\log 3 = 0.2075 \text{ bit}, \tag{2.33}$$

whereas

$$D(q||p) = \frac{3}{4}\log\frac{\frac{3}{4}}{\frac{1}{2}} + \frac{1}{4}\log\frac{\frac{1}{4}}{\frac{1}{2}} = \frac{3}{4}\log 3 - 1 = 0.1887 \text{ bit}. \tag{2.34}$$

Note that $D(p||q) \neq D(q||p)$ in general.

Mutual Information

- Measure of the amount of information that one RV contains about another RV

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = D(p(x,y)||p(x)p(y))$$

- Reduction in the uncertainty of one random variable due to the knowledge of the other

- Relationship between entropy and mutual information

$$I(X;Y) = H(Y) - H(Y|X)$$

Proof:

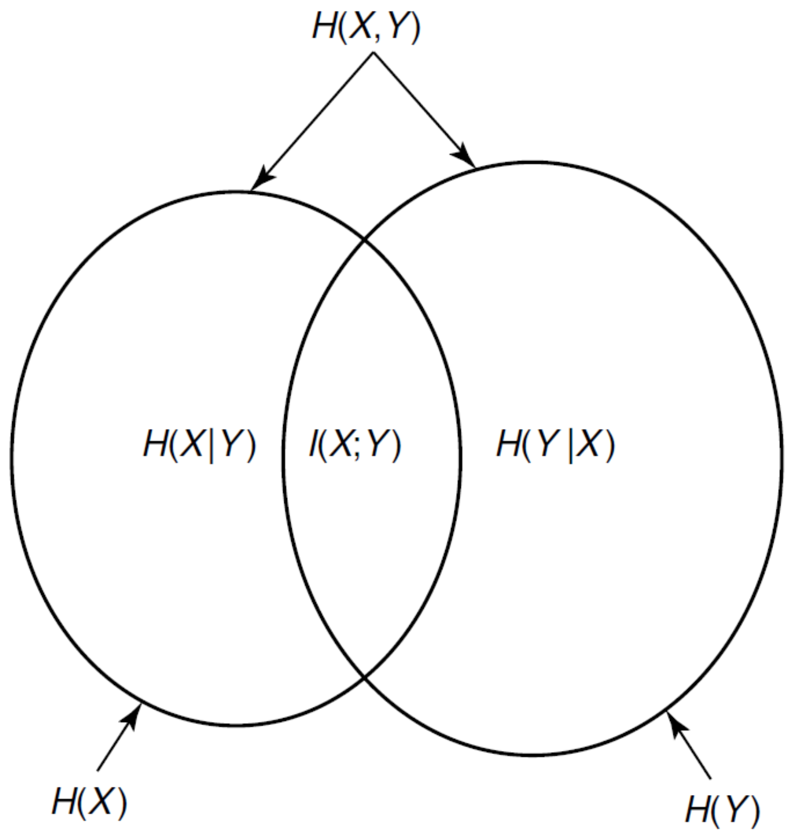
Mutual Information

- $I(X;Y) = H(Y) - H(Y|X)$
- $H(X,Y) = H(X) + H(Y|X) \rightarrow I(X;Y) = H(X) + H(Y) - H(X,Y)$
- $I(X;X) = H(X) - H(X|X) = H(X)$
  Entropy is "self-information"

Example: calculating mutual information

Vien Diagram

Vien diagram



$I(X;Y)$ is the intersection of information in X with information in Y

X: blood type

Y: chance for skin cancer

	A	B	AB	O
Very Low	1/8	1/16	1/32	1/32
Low	1/16	1/8	1/32	1/32
Medium	1/16	1/16	1/16	1/16
High	1/4	0	0	0

Conditional entropy: $H(X|Y) = 11/8$ bits, $H(Y|X) = 13/8$ bits

$$H(Y|X) \neq H(X|Y)$$

Mutual information: $I(X; Y) = H(X) - H(X|Y) = 0.375$ bit

What is the Entropy of English?

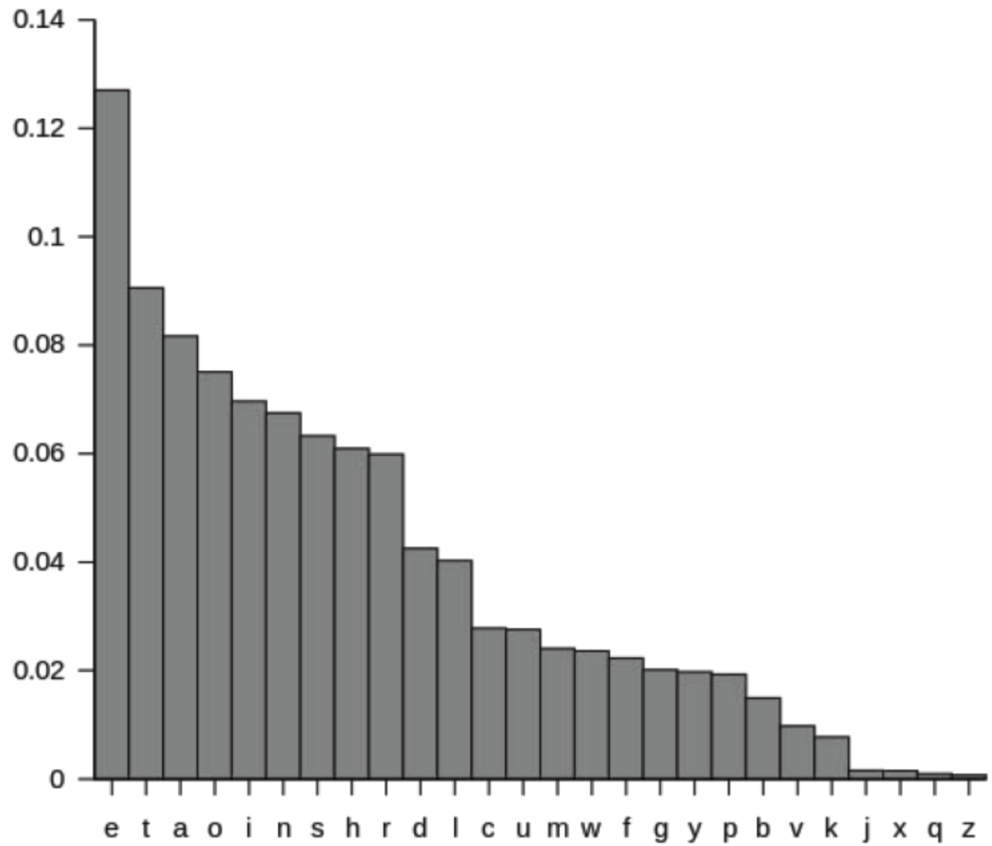


Image in the public domain. Source: Wikipedia.

Taking account of actual individual symbol probabilities, but not using context, entropy = 4.177 bits per symbol

English has Lots of Context

- Write down the next letter (or next 3 letters!) in the snippet

*Nothing can be said to be certain, except death and ta_*

But x has a very low occurrence probability (0.0017) in English words

  - Letters are not independently generated!

- Shannon (1951) and others have found that the entropy of English text is a lot lower than 4.177
  - Shannon estimated 0.6-1.3 bits/letter using human expts.
  - More recent estimates: 1-1.5 bits/letter

What we want to determine?

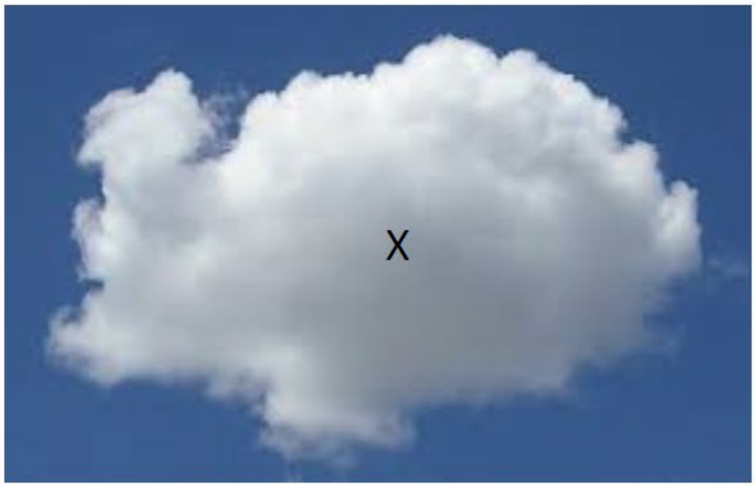
- Average per-symbol entropy over long sequences:

$$\underline{H} = \lim_{K->\infty} H(S_1, S_2, S_3, \ldots, S_K)/K$$

where S_j denotes the symbol in position j in the text.

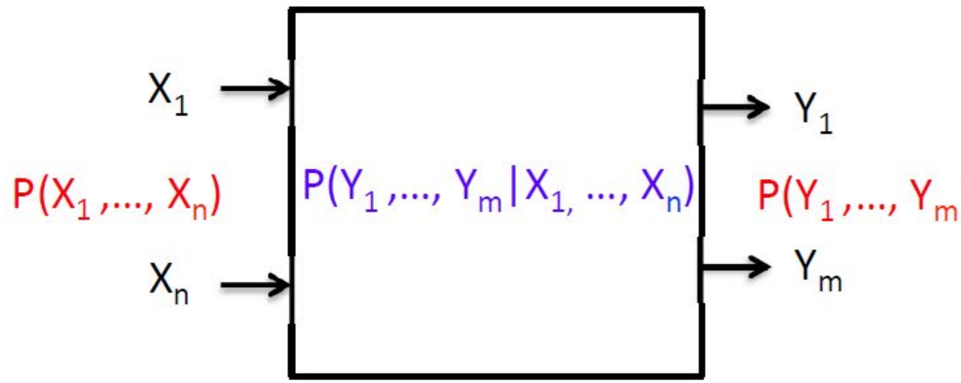
Summary

Entropy



H(X)

Mutual Information

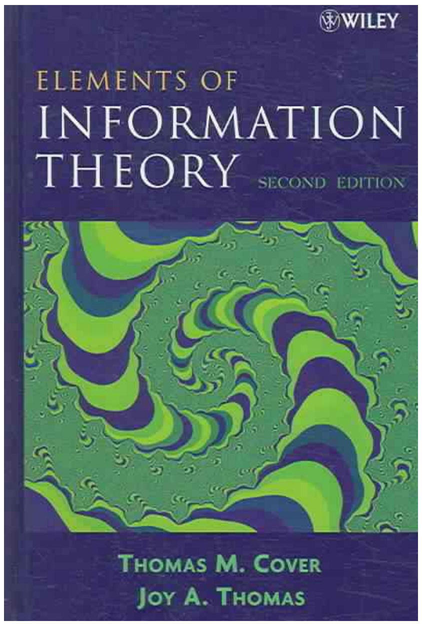


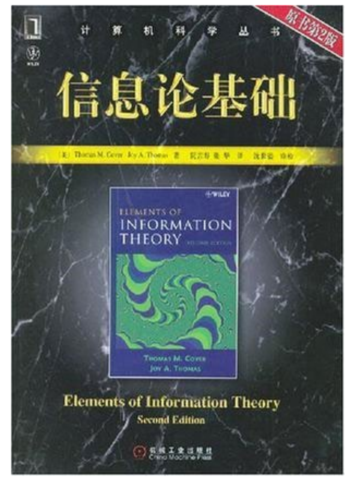
$I(X_1, \ldots, X_n; Y_1, \ldots, Y_m)$

Conditional Entropy

K-L Distance

Reference Book





Elements of Information Theory
Thomas M. Cover