CODING, CRYPTOGRAPHY

and

CRYPTOGRAPHIC PROTOCOLS

Prof. Jozef Gruska DrSc

Fall 2002

INTRODUCTION

- Transmission of classical information in time and space is nowadays very easy (through noiselesss channel).

  It took centuries, and many ingenious developments and discoveries (writting, book printing, photography, movies, radio transmissions, TV, sounds recording) and the idea of the digitalization of all forms of information to discover fully this property of information.

  Coding theory develops methods to protect information against a noise.

- Information is becoming an increasingly available commodity for both individuals and society.

  Cryptography develops methods how to protect information against an enemy (or an unauthorized user).

- A very important property of information is that it is often very easy to make unlimited number of copies of information.

  Steganography develops methods to hide important information in innocently looking information (and that can be used to protect intellectual properties).

HISTORY OF CRYPTOGRAPHY

The history of cryptography is the story of centuries-old battles between codemakers and codebreakers, an intellectual arms race that has had a dramatic impact on the course of history.

The ongoing battle between codemakers and codebreakers has inspired a whole series of remarkable scientific breakthroughts.

History is full of codes. They have decided the outcomes of battles and led to the deaths of kings and queens.

I. BASICS of CODING THEORY

ABSTRACT

Coding theory – theory of error correcting codes – is one of the most interesting and applied part of mathematics and informatics.

All real systems that work with digitally represented data, as CD players, TV, fax machines, internet, satelites, mobiles, require to use error correcting codes because all real channels are, to some extent, noisy.

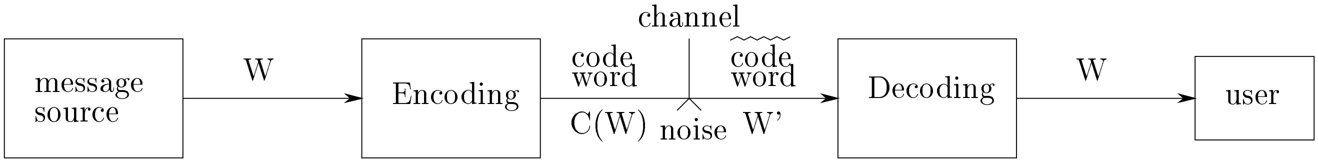
- Coding theory problems are therefore among the very basic and most frequent problems of storage and transmission of information.
- Coding theory results allow to create reliable systems out of unreliable systems to store and/or to transmit information.
- Coding theory methods are often elegant applications of very basic concepts and methods of (abstract) algebra.

Chapter presents and illustrates the very basic problems, concepts, methods and results of coding theory.

I. Coding - basic concepts

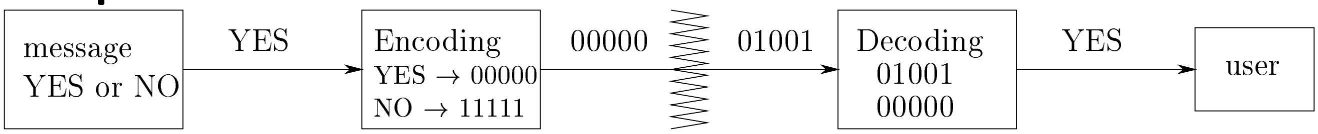
Without coding theory and error-correcting codes there would be no deep-space travel and pictures, no satelite TV, no compact disc, no . . . no . . . no . . . .

Error-correcting codes are used to correct messages when they are transmitted through noisy channels.



Error correcting framework

Example



A **code** C over an alphabet Σ is a subset of $\Sigma^\star$ — $(C \subset \Sigma^\star)$.

A **q-nary code** is a code over an alphabet of q-symbols.

A **binary code** is a code over the alphabet $\{0, 1\}$.

Examples of codes $C_1 = \{00, 01, 10, 11\}$ $C_2 = \{000, 010, 101, 100\}$
$C_3 = \{00000, 01101, 10111, 11011\}$

CHANNEL

is the physical medium through which information is transmitted.

(Telephone lines and the atmosphere are examples of channels.)

NOISE

may be caused by sunpots, lighting, meteor showers, random radio disturbance, poor typing, poor hearing, ...

TRANSMISSION GOALS

1. Fast encoding of information.
2. Easy transmission of encoded messages.
3. Fast decoding of received messages.
4. Reliable correction of errors introduced in the channel.
5. Maximum transfer of information per unit time.

METHOD OF FIGHTING ERRORS: REDUNDANCY

0 is encoded as 00000 and 1 is encoded as 11111.

Example (Codings of a path avoiding an enemy territory)

Story Alice and Bob share an identical map (Fig. 1) gridded as shown in Fig.1. Only Alice knows the route through which Bob can reach her avoiding the enemy territory. Alice wants to send Bob the following information about the route he should take.

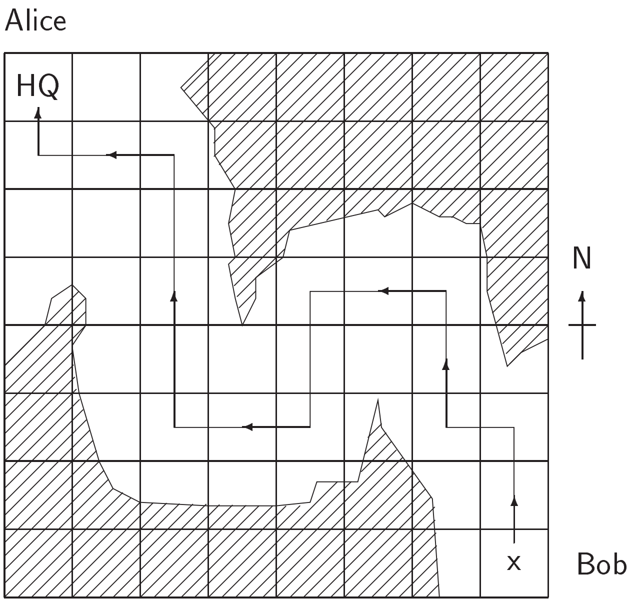


Fig. 1

NNWNNWWSSWWNNNNNWWN

Three ways to encode the safe route from Bob to Alice.

1.

$$C_1 = \{00, 01, 10, 11\}$$

Any error in the code word
000001000001011111010100000000010100
would be a disaster.

2.

$$C_2 = \{000, 011, 101, 110\}$$

A single error in encoding each of symbols N, W, S, E could be **detected**.

3.

$$C_3 = \{00000, 01101, 10110, 11011\}$$

A single error in decoding each of symbols N, W, S, E could be **corrected**.

Basic terminology

Block code – a code with all words of the same length.

Codewords – words of some code.

Basic assumptions about channels

1. Code length preservation Each output codeword of a channel has the same length as the input codeword.

2. Independence of errors The probability of any one symbol being affected in transmissions is the same.

Basic strategy for decoding

For decoding we use the so-called **maximal likehood principle**, or nearest neighbor decoding strategy, which says that the receiver should decode a word w' as that codeword w that is the closest one to w'.

Hamming distance

The intuitive concept of "closeness" of two words is well formalized through **Hamming distance** $h(x, y)$ of words x, y.

For two words x, y

$$h(x, y) = \text{the number of symbols } x \text{ and } y \text{ differ.}$$

Example

$$h(10101, 01100) = 3, h(fourth, eighth) = 4$$

Properties of Hamming distance

(1) $h(x, y) = 0 \Leftrightarrow x = y$
(2) $h(x, y) = h(y, x)$
(3) $h(x, z) \leq h(x, y) + h(y, z)$ triangle inequality

An important parameter of codes C is their **minimal distance**.

$$h(C) = \min\{h(x, y) | x, y \in C, x \neq y\},$$

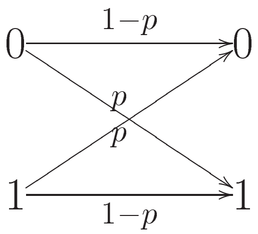
because it gives the smallest number of errors needed to change one codeword into anther.

Theorem Basic error correcting theorem
(1)A code C can detected up to s errors if $h(C) \geq s + 1$.
(2) A code C can correct up to t errors if $h(C) \geq 2t + 1$.

Proof (1) Trivial.
(2) Suppose $h(C) \geq 2t + 1$. Let a codeword x is transmitted and a word y is recceived with $h(x, y) \leq t$. If $x' \neq x$ is a codeword, then $h(x', y) \geq t + 1$ because otherwise $h(x', y) < t + 1$ and therefore $h(x, x') \leq h(x, y) + h(y, x') < 2t + 1$ what contradicts the assumption $h(C) \geq 2t + 1$.

Binary symmetric channel

Consider a transmition of binary symbols such that each symbol has probability of error $p < \frac{1}{2}$.



Binary symmetric channel

If n symbols are transmitted, then the probability of t errors is

$$p^t(1-p)^{n-t}\binom{n}{t}.$$

In the case of binary symmetric channels the "nearest neighbour decoding strategy" is also "maximum likehood decoding strategy".

Example Consider $C = \{000, 111\}$ and the nearest neighbour decoding strategy.

Probability that the received word is decoded correctly

as 000 is $(1-p)^3 + 3p(1-p)^2$,
as 111 is $(1-p)^3 + 3p(1-p)^2$.

Therefore

$$P_{err}(C) = 1 - ((1-p)^3 + 3p(1-p)^2)$$

is the so-called **word error probability**.

Example If $p = 0.01$, then $P_{err}(C) = 0.000298$ and only one word in 3555 will reach the user with an error.

Addition of one parity-check bit

Example Let all 2^{11} of binary words of length 11 be codewords.
Let the probability of an error be 10^{-8}.
Let bits be transmitted at the rate 10^7 bits per second.

The probability that a word is transmitted incorrectly is approximately

$$11p(1-p)^{10} \approx \frac{11}{10^8}.$$

Therefore $\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1$ of words per second are transmitted incorrectly.

One wrong word is transmitted every 10 seconds, 360 erroneous words every hour and 8640 words every day without being detected!

Let one parity bit be added.

Any single error can be detected.

The probability of at least two errors is:

$$1-(1-p)^{12} - 12(1-p)^{11}p \approx \binom{12}{2}(1-p)^{10}p^2 \approx \frac{66}{10^{16}}$$

Therefore approximately $\frac{66}{10^{16}} \cdot \frac{10^7}{12} \approx 5.5 \cdot 10^{-9}$ words per second are transmitted with an undetectable error.

Corollary One undetected error occurs only every 2000 days!
$(2000 \approx 10^9/(5.5 \times 86400).)$

TWO-DIMENSIONAL PARITY CODE

The two-dimensional parity code arranges the data into a two-dimensional array and then to each row (column) parity bit is attached.

Example Binary string

$$10001011000100101111$$

is represented and encodded as follows

$$\begin{matrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{matrix}$$

Question How much better is two-dimensional encoding than one-dimensional encoding?

Notation and Examples

Notation: An (n, M, d) – code C is a code such that

- n – is the length of codewords.
- M – is the number of codewords.
- d – is the minimum distance in C.

Example $C_1 = \{00, 01, 10, 11\}$ is a $(2, 4, 1)$-code.
$C_2 = \{000, 011, 101, 110\}$ is a $(3, 4, 2)$-code.
$C_3 = \{00000, 01101, 10110, 11011\}$ is a $(5, 4, 3)$-code.

Comment: A good (n, M, d) code has small n and large M and d.

Example (Transmission of photographs from the deep space)

- In 1965=69 **Mariner 4-5** took the first photographs of another planet – 22 photos. Each photo was divided into 200×200 elementary squares – pixels. Each pixel was assigned 6 bits representing 64 levels of brightness. Hadamard code was used.

  Transmission rate: 8.3 bits per second.

- In 1970-72 **Mariners 6 – 8** took such photographs that each picture was broken into 700×832 squares. Reed-Muller $(32, 64, 16)$ code was used.

  Transmission rate was 16200 bits per second. (Much better pictures)

HADAMARD CODE

In Mariner 5, 6-bit pixels were encoded using 32-bit long Hadamard code that could correct up to 7 errors.

Hadamard code had 64 codewords. 32 of them were represented by the 32×32 matrix $H = \{h_{ij}\}$, where $0 \leq i, j \leq 4$ and

$$h_{ij} = (-1)^{a_0 b_0 + a_1 b_1 + \ldots + a_4 b_4}$$

where i and j have binary representations

$$i = a_4 a_3 a_2 a_1 a_0, j = b_4 b_3 b_2 b_1 b_0.$$

The remaing 32 codewords were represented by the matrix $-H$.

Decoding was quite simple.

CODE RATE

For q-nary (n, M, d)-code we define **code rate**, or **information rate**, R, by

$$R = \frac{\lg_q M}{n}.$$

The code rate represents the ratio of the number of input data symbols to the number of transmitted code symbols.

Code rate ($\frac{6}{12}$ for Hadamard code), is an important parameter for real implementations, because it shows what fraction of the bandwidth is being used to transmit actual data.

The ISBN-code

Each recent book has International Standard Book Number which is a 10-digit codeword produced by the publisher with the following structure:

l	-	p	-	m	-	w	$=$	x_1	$\ldots$	x_{10}
language	-	publisher	-	number	-	weighted check sum				
0	-	07	-	709503	-	0				

such that

$$\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$$

The publisher has to put X into the 10-th position if $x_{10} = 10$.

The ISBN code is designed to detect: (a) any single error (b) any double error created by a transposition

Single error detection

Let $X = x_1 \ldots x_{10}$ be a correct code and let

$$Y = x_1 \ldots x_{j-1} y_j x_{j+1} \ldots x_{10} \text{ with } y_j = x_j + a, \; a \neq 0$$

In such a case:

$$\sum_{i=1}^{10} i y_i = \sum_{i=1}^{10} i x_i + ja \not\equiv 0 \pmod{11}$$

Transposition detection

Let x_j and x_k be exchanged.

$$\begin{aligned} \sum_{i=1}^{10} i y_i &= \sum_{i=1}^{10} i x_i + (k-j)x_j + (j-k)x_k \\ &= (k-j)(x_j - x_k) \not\equiv 0 \bmod 11 \qquad \text{if } k \neq j \text{ and } x_j \neq x_k. \end{aligned}$$

Equivalence of codes

Definition Two q-ary codes are called **equivalent** if one can be obtained from the other by a combination of operations of the following type:

(a) a permutation of the positions of the code.
(b) a permutation of symbols appering in a fixed position.

Question: Let a code be displayed as an $M \times n$ matrix. To what correspond opperations (a) and (b)?

Claim: Distances between codewords are unchanged by operations (a), (b). Consequently, equivalent codes have the same parameters (n, M, d) (and correct the same number of errors).

Examples of equivalent codes

$$(1) \left\{ \begin{matrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{matrix} \right\} \qquad \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{matrix} \right\}$$

$$(2) \left\{ \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{matrix} \right\} \qquad \left\{ \begin{matrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{matrix} \right\}$$

Lemma Any q-ary (n, M, d)-code over an alphabet $\{0, 1, \ldots, q-1\}$ is equivalent to an (n, M, d)-code which contains the all-zero codeword $00\ldots0$.

Proof Trivial.

The main coding theory problem

A good (n, M, d)-code has small n, large M and large d.

The main coding theory problem is to optimize one of the parameters n, M, d for given values of the other two.

Notation: $A_q(n, d)$ is the largest M such that there is an q-nary (n, M, d)-code.

Theorem (a) $A_q(n, 1) = q^n$; (b) $A_q(n, n) = q$.

Proof (a) obvios; (b) Let C be an q-nary (n, M, n)-code. Any two distinct codewords of C differ in all n positions. Hence symbols in any fixed position of M codewords have to be different $\Rightarrow A_q(n, n) \leq q$. Since the q-nary repetition code is (n, q, n)-code, we get $A_q(n, n) \geq q$.

Example Proof that $A_2(5, 3) = 4$.
(a) Code C_3 is a $(5, 4, 3)$-code, hence $A_2(5, 3) \geq 4$.
(b) Let C be a $(5, M, 3)$-code with $M \geq 4$.
• By previous lemma we can assume that $00000 \in C$.
• C contains at most one codeword with at least four 1's.
(otherwise $d(x, y) \leq 2$ for two such codewords x, y)
• Since $00000 \in C$ there can be no codeword in C with one or two 1.
• Since $d = 3$ C cannot contain three codewords with three 1's. • Since $M \geq 4$ there have to be in C two codewords with three 1's.
(say 11100, 00111), the only possible codeword with four or five 1's is then 11011.

Theorem Suppose d is odd. Then a binary (n, M, d)-code exists iff a binary $(n+1, M, d+1)$-code exists.

Proof **Only if case**: Let C be a binary code (n, M, d)-code. Let

$$\hat{C} = \{x_1 \dots x_n x_{n+1} | x_1 \dots x_n \in C,\ x_{n+1} = (\sum_{i=1}^{n} x_i) \mod 2\}$$

Since parity of all codewords in $\hat{C}$ is even, $d(\hat{x}, \hat{y})$ is even for all

$$\hat{x}, \hat{y} \in \hat{C}.$$

Hence $d(\hat{C})$ is even. Since $d \le d(\hat{C}) \le d+1$ and d is odd,

$$d(\hat{C}) = d + 1.$$

Hence $\hat{C}$ is an $(n+1, M, d+1)$-code.

If case: Let D be an $(n+1, M, d+1)$-code. Choose code words x, y of D such that $d(x, y) = d + 1$.
Find a position in which x, y differ and delete this position from all codewords of D. Resulting code is an (n, M, d)-code.

Corollary: If d is odd, then $A_2(n, d) = A_2(n+1, d+1)$.
If d is even, then $A_2(n, d) = A_2(n-1, d-1)$.

Example $A_2(5, 3) = 4 \Rightarrow A_2(6, 4) = 4$
$(5, 4, 3)$-code $\Rightarrow$ $(6, 4, 4)$-code

$$\begin{matrix} 00000 \\ 01101 \\ 10110 \\ 11000 \end{matrix}$$

by adding check.

A general upper bound on $A_q(n, d)$

Notation: F_q^n – are set of all words of length n over alphabet $\{0, 1, 2, \ldots, q-1\}$

Definition For any codeword $u \in F_q^n$ and any integer $r \geq 0$ the **sphere** of radius r and centre u is denoted by

$$S(u, r) = \{v \in F_q^n | d(u, v) \leq r\}.$$

Theorem A sphere of radius r in F_q^n, $0 \leq r \leq n$ contains

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \ldots + \binom{n}{r}(q-1)^r$$

words.

Proof Let u be a fixed word in F_q^n. The number of words that differ from u in m position is

$$\binom{n}{m}(q-1)^m.$$

Theorem **(The sphere-packing or Hamming bound)**
If C is a q-nary $(n, M, 2t+1)$-code, then

$$M\left\{\binom{n}{0} + \binom{n}{1}(q-1) + \ldots + \binom{n}{t}(q-1)^t\right\} \leq q^n \tag{1}$$

Proof Any two spheres of radius t centered on distinct codewords have no codeword in common. Hence the total number of words in M spheres of radius t centered on M codewords is given by the left side (1). This number has to be less or equal to q^n.

A code which achieves the sphere-packing bound from (1), i.e. such that equality holds in (1), is called a **perfect code**.

Example An $(7, M, 3)$-code is perfect if

$$M\left(\binom{7}{0}+\binom{7}{1}\right)=2^7$$

i.e. $M = 16$

An example of such a code:
$C_4 = \{0000000, 1111111, 1000101, 1100010, 0110001, 1011000, 0101100, 0010110, 0001011, 0111010, 0011101, 1001110, 0100111, 1010011, 1101001, 1110100\}$

Table of $A_2(n, d)$ from 1981

n	$d = 3$	$d = 5$	$d = 7$
5	4	2	–
6	8	2	–
7	16	2	2
8	20	4	2
9	40	6	2
10	72–79	12	2
11	144–158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2560–3276	256–340	36–37

For current best results see http://www.win.tue.nl/math/dw/voorlincod.html

LOWER BOUND for $A_q(n,d)$

The following lower bound for $A_q(n,d)$ is known as **Gilbert-Varshanov bound**:

Theorem Given $d \leq n$, there exists a q-ary (n,M,d)-code with

$$M \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j}$$

and therefore

$$A_q(n,d) \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j}$$

General coding problem

The basic problems of information theory are how to define formally such concepts as information and how to store or transmit information efficiently.

Let X be a random variable (source) which takes a value x with probability $p(x)$. The **entropy** of X is defined by

$$S(X) = -\sum_x p(x) \lg p(x)$$

and it is considered to be the **information content** of X.

The maximum information which can be stored by an n-value variable is $\lg n$.

In a special case of a binary variable X which takes on the value 1 with probability p and the value 0 with probability $1 - p$

$$S(X) = H(p) = -p \lg p - (1-p) \lg(1-p)$$

Problem: What is the minimal number of bits we need to transmit n values of X?

Basic idea: To encode more probable outups of X by shorter binary words.

Example (Morse code)

a	.-	b	-...	c	-.-.	d	-..	e	.	f	..-.	g	--.
h	....	i	..	j	.---	k	-.-	l	.-..	m	--	n	-.
o	---	p	.--.	q	--.-	r	.-.	s	...	t	-	u	..-
v	...-	w	.--	x	-..-	y	-.--	z	--..				

Shannon's noisless coding theorem

In a simple form Shannon's noisless coding theorem says that in order to transmit n values of X we need $nS(X)$ bits.

More exactly, we cannot do better and we can reach the bound $nS(X)$ as close as desirable.

Example Let a source X produce the value 1 with probability $p = \frac{1}{4}$
Let the source X produce the value 0 with probability $1 - p = \frac{3}{4}$

Assume we want to encode blocks of the outputs of X of length 4.

By Shannon's theorem we need $4H\left(\frac{1}{4}\right) = 3.245$ bits per blocks (in average)

A simple and practical methods known as **Huffman's code** requires in this case 3.273 bits per message.

mess.	code	mess.	code	mess.	code	mess.	code
0000	10	0100	010	1000	011	1100	11101
0001	000	0101	11001	1001	11011	1101	111110
0010	001	0110	11010	1010	11100	1110	111101
0011	11000	0111	1111000	1011	111111	1111	1111001

Observe that this is a **prefix code** – no codeword is a prefix of another codeword.

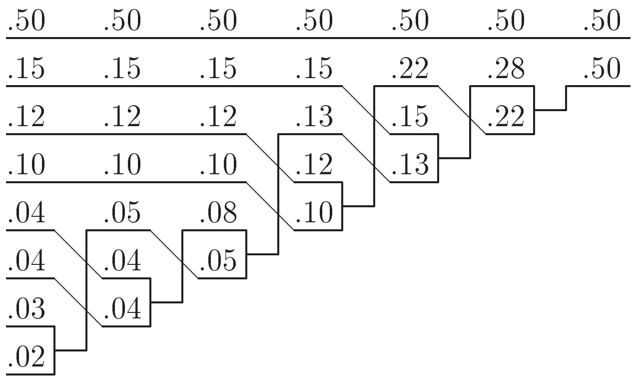
Design of Huffman code

Given a sequence of n objects, $x_1, \ldots, x_n$ with probabilities $p_1 \geq \ldots \geq p_n$.

Stage 1 – shrinking of the sequence.

(•) Replace x_{n-1}, x_n with a new object y_{n-1} with probability $p_{n-1} + p_n$ and rearrange sequence so one has again nonincreasing probabilities.

(•) Keep doing the above step till the sequence shrinks to two objects.



Stage 2 – extending the code – Apply again and again the following method.

If $C = \{c_1, \ldots, c_r\}$ is a prefix optimal code for a sourse S_r, then $C' = \{c'_1, \ldots, c'_{r+1}\}$ is an optimal code for S_{r+1}, where

$$c'_i = c_i \qquad 1 \leq i \leq r-1$$

$$c'_r = c_r 1$$

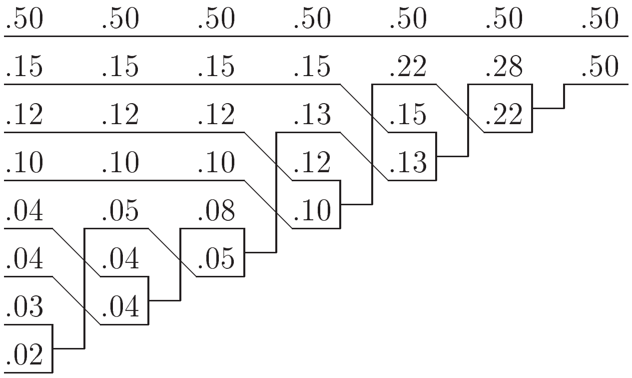
$$c'_{r+1} = c_r 0.$$

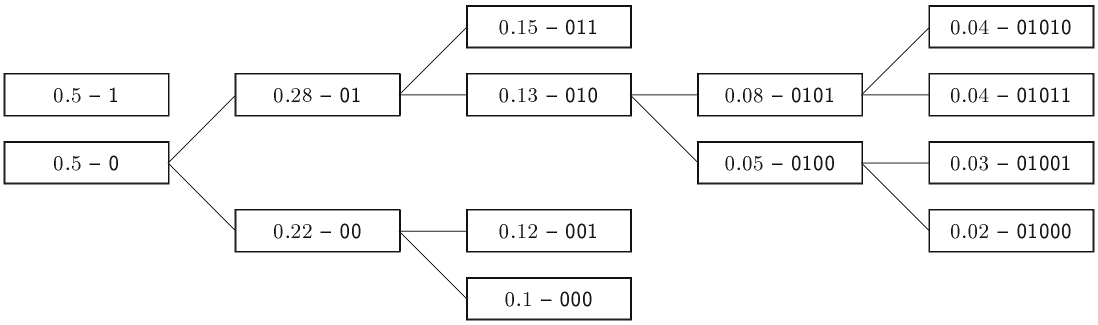
Stage 2 Apply again and again the the following method:
If $C = \{c_1, \ldots, c_r\}$ is a prefix optimal code for a source S_r, then $C' = \{c'_1, \ldots, c'_{r+1}\}$ is an optimal code for S_{r+1}, where

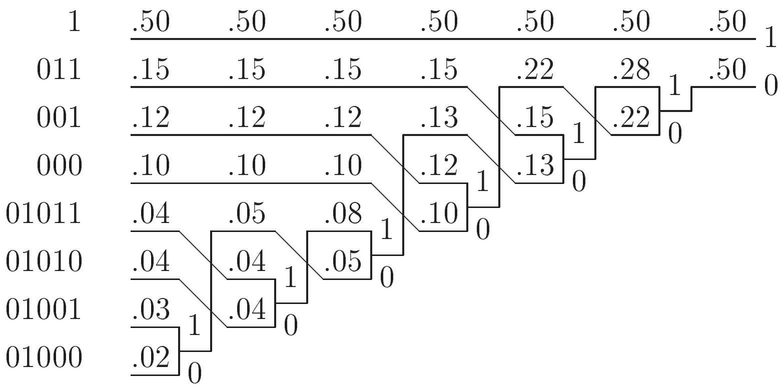
$$c'_i = c_i \qquad 1 \le i \le r-1$$

$$c'_r = c_r 1$$

$$c'_{r+1} = c_r 0$$







A BIT OF HISTORY

The subject of error-correcting codes arose originally as a response to practical problems in the reliable communication of digitally encoded information.

The discipline was initiated in the paper

Claude Shannon: A mathematical theory of communication, Bell Syst. Tech. Journal V27, 1948, 379-423, 623-656

Shannon's paper started the scientific discipline **information theory** and **error-corecting codes** are its part.

Originally, information theory was a part of electrical engineering. Nowadays, it is an important part of mathematics and also of informatics.