ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

INFORMATION THEORY

Stanislav Palúch

Európsky sociálny fond

ŽILINA, 2008

ŽILINSKÁ UNIVERZITA V ŽILINE, Fakulta riadenia a informatiky

Dvojjazyková publikácia slovensky - anglicky
Double - language publication Slovak - English

INFORMATION THEORY
Palúch Stanislav

Podľa slovenského originálu

Palúch Stanislav
TEÓRIA INFORMÁCIE
Vyd. Žilinská univerzita vŽiline/ EDIS - vydavateľstvo ŽU, Žilina, v tlači

Translation: Doc. RNDr. Stanislav Palúch, CSc.

Slovak version reviewed by: Prof. RNDr. Jan Černý, Dr.Sc., DrHc.,
Prof. RNDr. Beloslav Riečan,Dr.Sc.,
Prof. Ing. Mikuláš Alexík, CSc.
English version reviewed by: Ing. Daniela Stredákova

Vydala Žilinská univerzita vŽiline, Žilina 2008
Issued by University of Žilina, Žilina 2008

Tlač / Printed by
ISBN
Vydané spodporou Európskeho sociálneho fondu,
projekt SOP ĽZ - 2005/NP1-007
Issued with support of European Social Foundation,
project SOP ĽZ - 2005/NP1-007

Európsky
sociálny
fond

Contents

– Where is wisdom?
– Lost in knowledge.
– Where is knowledge?
– Lost in information.
– Where is information?
– Lost in data.

T. S. Eliot

Preface

The mankind, living the third millennium, comes to what can be described as *information age*. We are (and we will be) increasingly overflown with abundance of various information. Press, radio, television with their terrestrial and satellite versions and lately Internet are sources of more and more information. A lot of information originates from activities of state and regional authorities, enterprises, banks, insurance companies, various funds, schools, medical and hospital services, police, security services and citizens themselves.

Most frequent operations with information is its transmission, storage, processing and utilization. The importance of the protecting of information against disclosure, stealing, misuse and unauthorised modification grows significantly.

Technology of transmission, storing and processing of information has a crucial impact on development of human civilisation. There are several information revolutions described in literature.

The origin of speech is mentioned as the first information revolution. Human language became a medium for handover and sharing information among various people. Human brain was the only medium for storage of information.

The invention of script is mentioned as the second information revolution. The information was transferred only verbally by tradition until it could be stored and carried forward through space and time. This had the consequence that the civilisations which invented a written script started to develop more quickly than until then similarly advanced communities – to these days there are some forgotten tribes living as in the stone age.

The third information revolution was caused by the invention of the printing press (J. Gutenberg, 1439). Gutenberg's printing technology spread rapidly throughout Europe and has made information accessible to many people. Knowledge and culture of people have risen as basic fundamentals for the industrial revolution and for the origin of modern industrial society.

The fourth information revolution is related to the development of computers and communication technique and their capability to store and process information. The separation of information from its physical carrier during transmission and enormous capacity of memory storage devices along with fast computer processing and transmitting is considered as a tool of boundless consequences.

On the other hand, organization of our contemporary advanced society is much more complicated. Globalization is one of specific features of present years. Economics of individual countries are not separated anymore – international corporations are more and more typical. Most of today's complicated products is composed from parts coming from many parts of world.

Basic problems of countries surpass through their borders and grow into worldwide issues. Protecting the environment, global warming, nuclear energy, unemployment, epidemic prevention, international criminality, marine reserves, etc., are examples of such issues.

The solving of such issues requires a coordination of governments, managements of large enterprises, regional authorities and citizens which is not possible without a transmission of information among the mentioned subjects. The construction of efficient information network and its optimal utilization is one of duties of every modern country. The development and build up of communication networks are very expensive and that is why we face very often the question whether existing communication line is exploited to its maximum capacity, or if it is possible to make use of an optimization method for increasing the amount of transferred information.

It was not easy to give a qualified answer to this question (and it is not easy up to now). The application of an optimization method involves creating a mathematical model of information source, transmission path, and processes that accompany the transmission of information. These issues appeared during the World War II and become more and more important ever since. It was not possible to include them into any established field of mathematics. Therefore a new branch of science called information theory had to be founded (by Claude E. Shannon). The information theory was initially a part of mathematical cybernetics which grew step by step into a younger scientific discipline – informatics.

The information theory distinguishes the following phases in a transfer of information:

- transmitting messages from information source
- encoding messages in encoder
- transmission through information channel
- decoding messages in decoder
- receiving messages in receiver

The first problem of the information theory is to decide which objects carry an information and how to quantify the amount of information. The idea to identify the amount of information with the corresponding data file size is wrong, since there are many ways of storing the same information resulting in various file sizes (e. g., using various software compression utilities PK-ZIP, ARJ, RAR, etc.).

We will see that it is convenient to assign information to events of some universal probability space $(\Omega, \mathcal{A}, P)$. Most of books on the information theory start with the Shannon - Hartley formula $I(A) = -\log_2 P(A)$ without any motivation. A reader of pioneering papers about the information theory can see that the way to this formula was not straightforward. In the first chapter of this book, I aim to show this motivation. In addition to the traditional way of assigning information to events of some probablility space, I show (for me extraordinary beautiful) the way suggested by Černý and Brunovský [4] of introducing information without probability.

The second chapter is devoted to the notion of entropy of a finite partition $A_1, A_2, \dots, A_n$ of universal space of elementary events Ω. This entropy should express the amount of our hesitation – uncertainty before executing an experiment with possible outcomes $A_1, A_2, \dots, A_n$. Two possible ways of defining entropy are shown, both are leading to the same result.

The third chapter studies information sources, their properties and defines the entropy of information sources.

The fourth chapter deals with encoding and decoding of messages. The main purpose of encoding is to make the alphabet of the message suitable for transmission over a channel. Other purposes of encoding are compression, ability to reveal certain errors, or to repair a certain number of errors. Compression and error-correcting property are contradictory requirements and it is not easy to comply with them. We will see that many results of algebra, finite groups, rings, and field theory, and finite linear space theory is very useful for modelling and solving encoding problems. The highlight of this chapter is the fundamental source coding theorem: The source entropy is the lower bound of the average value of length of binary compressed messages from this source.

The information channel can be modelled by means of elementary probability theory. In this book I constrain myself to the simplest memoryless stationary channel since such a channel describes common frequent channels and can be relatively easy modelled by elementary mathematical means. I introduce three definitions of channel capacity. For memoryless stationary channels all definitions lead to the same value of capacity.

It shows that messages from a source with the entropy H can be transferred through a channel with the capacity C, if $H < C$. This fact is exactly formulated in two Shannon theorems.

This book contains fundamental definitions and theorems from the fields of information theory and coding. Since this publication is targeted to engineers in informatics I skip complicated proofs – the reader can find them in cited references. All proofs in this book are finished by the symbol ■, complicated sections that can be skipped without loss of continuity are marked by the asterisk.

I wish to thank prof. J. Černý, prof. B. Riečan and prof. M. Alexík for their careful readings, suggestions and correctings many errors.

I am fascinated by the information theory, because it puts together purposefully and logically results of continuous and discrete, deterministic and probabilistic mathematics – probability theory, measure theory, number theory, and algebra into one comprehensive, meaningful, and applicable theory. I wish the reader will have the same aesthetic pleasure, when reading this book, as I had while writing it.

Author.

Chapter 1

Information

1.1 Ways and means of introducing information

Requiring an information about the departure of IC train TATRAN from Žilina for Bratislava, we can get it exactly in the form of the following sentence: "*IC train Tatran for Bratislava departs from Žilina at 15:30.*" A friend not remembering exactly can give the following answer: "*I do not remember exactly, but the departure is surely between 15:00 and 16:00.*"

A student announces the result of an exam: "*My result of the exam from algebra is B.*" Or only shortly: "*I passed the exam from algebra.*"

At the beginning of football match a sportscaster informs: "*I estimate the number of football fans from 5 to 6 thousands.*" After obtaining the exact data from organizers he puts more exactly: "*The organizers sold 5764 tickets.*"

Each of these propositions carries a certain amount of information with it. We intuitively feel that the exact answer about the train departure (15:30) contains more information than that one of the friend (between 15:00 and 16:00) although even the second one is useful. Everyone will agree that the proposition "*Result of the exam is B*" contains more information than mere "*I passed the exam.*"

The possible departures of IC train Tatran are 00:00, 00:01, 00:03, . . . , 23:58, 23:59 – there exist 1440 possibilities. There are 6 possibilities of the result of exam (A, B, C, D, E, FX). It is easier to guess the result of an exam than the exact departure time of a train.

Our intuition says us that the exact answer about the train departure gives us more information than the exact answer about the result of an exam. The question rises how to quantify the amount of information.

Suppose that information will be defined as a real function $I : \mathcal{A} \rightarrow \mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers), assigning a non negative real number to every element from the set $\mathcal{A}$.

The first problem is in the specification of the set $\mathcal{A}$. At the first glance it could seem convenient to take the set of all propositions[1] for the set $\mathcal{A}$. Working with propositions is not very comfortable. We would rather work with more simple and more standard mathematical objects.

Most of information–carrying propositions is a sentence in the form: "*Event A occurred.*", resp., "*Event A will occur.*"

The **event** A in the information theory can be defined similarly as in the probability theory as a subset of a set Ω where Ω is the set of all possible outcomes, sometimes called **sample space**, or **universal sample space**[2].

In cases, when Ω is an infinite set, certain theoretical difficulties related to measurability of its subset $A \subseteq \Omega$ can occur[3]. As we will see later, the information of a set A is a function of its probability measure. Therefore we restrict ourselves to such a system of subsets of Ω for which we are able to assign their measure. It shows that such system of subsets of Ω contains the sample space Ω and is closed under complementation, and countable unions of its members.

[1] Proposition is a statement – a meaningful declarative sentence – for which it makes sense to ask whether it is true or not.

[2] It is convenient to imagine that the set Ω is the set of all possible outcomes for all universe and every time. However, if the reader has difficulties with the idea of such broad universal sample space, he or she can consider that Ω is the set of all possible outcomes different for every individual instance – e. g. when flipping a coin $\Omega = \{0, 1\}$, when studying rolling a die $\Omega = \{1, 2, 3, 4, 5, 6\}$, etc.
Suppose that for every $A \subseteq \Omega$ there is a function $\chi_A : \Omega \rightarrow \{0, 1\}$ such that if $\omega \in A$, then $\chi_A(\omega) = 1$, if $\omega \notin A$ then $\chi_A(\omega) = 0$.

[3] A measurable set is such subset of Ω which can be assigned a Lebesgue measure. It was shown that subsets of the set $\mathbb{R}$ of all real numbers exits that are non measurable. For such nonmeasurable sets it is not possible to assign their probability and therefore we restrict ourselves only to measurable sets.
However, the reader does not need to concern himself about nonmeasurability of sets because all instances of nonmeasurable sets were created by means of axiom of choice. Therefore all sets used in practice are measurable.

Definition 1.1. Let Ω be a nonempty set called **sample space** or **universal sample space**. **σ-algebra** of subsets of sample space Ω is such a system $\mathcal{A}$ of subsets of Ω, for which it holds:

1. $\Omega \in \mathcal{A}$
2. If $A \in \mathcal{A}$ then $A^C = (\Omega - A) \in \mathcal{A}$
3. If $A_n \in \mathcal{A}$ for $n = 1, 2, \ldots,$ then $\displaystyle\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}$.

σ-algebra $\mathcal{A}$ contains the sample space Ω. Furthermore, it contains with any finite or infinite sequence of sets, their union, and with every set it contains its complement, too. It can be easily shown that σ-algebra contains the empty set $\emptyset$ (complement of Ω) and with any finite or infinite sequence of sets it contains their intersection, too.

Now our first problem is solved. We will assign information to all elements of σ-algebra of some sample space Ω.

The second problem is how to define a real function $I : \mathcal{A} \to \mathbb{R}$ (where $\mathbb{R}$ is the set of all real numbers) in such a way that the value $I(A)$ for $A \in \mathcal{A}$ expresses the amount of information contained in the message "*The event A occurred.*"

We were in analogical situation when we introduced the probability on σ-algebra $\mathcal{A}$. There are three ways how to define the probability – the elementary way, the axiomatic way and the way making use of the notion of normalized measure on measurable space $(\Omega, \mathcal{A})$.

The analogy of elementary approach will do. This approach can be characterised as follows:

Suppose that the sample space is the union of finite number n mutually disjoint events:

$$\Omega = A_1 \cup A_2 \cup \cdots \cup A_n \ .$$

Then the probability of each of them is $\frac{1}{n}$ – i. e., $P(A_i) = \frac{1}{n}$ for every $i = 1, 2, \ldots, n$.
σ-algebra $\mathcal{A}$ will contain the empty set $\emptyset$ and all finite unions of the type

$$A = \bigcup_{k=1}^{m} A_{i_k}, \tag{1.1}$$

where $A_{i_k} \neq A_{i_l}$ for $k \neq l$. Then every set $A \in \mathcal{A}$ of the form 1.1 is assigned the probability $P(A) = \frac{m}{n}$. This procedure can be used also in more general case when the sets $A_1, A_2, \dots, A_n$ are given arbitrary probabilities $p_1, p_2, \dots, p_n$ where $p_1 + p_2 + \dots + p_n = 1$. In this case the probability of the set A from 1.1 is defined as $P(A) = \sum_{k=1}^{m} p_{i_k}$.

Additivity is an essential property of probability – for every $A, B \in \mathcal{A}$ such that $A \cap B = \emptyset$ it holds $P(A \cup B) = P(A) + P(B)$. However, for information $I(A)$ we expect that if $A \subseteq B$ then $I(B) \leq I(A)$, i. e., that information of "smaller" event is greater or equal than the information of the "larger" one. This implies that if $I(A \cup B) \leq I(A)$, $I(A \cup B) \leq I(B)$, and therefore for non-zero $I(A)$, $I(B)$ it cannot hold $I(A \cup B) = I(A) + I(B)$.

Here is the idea of further procedure:
Since binary operation

$$+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

is not suitable for calculation the information of the disjoint union of two sets using their informations we try to introduce other binary operation:

$$\oplus : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+,$$

(where $\mathbb{R}_0^+$ is the set of all non-negative real numbers) which expresses the information of disjoint union of two sets A, B as follows:

$$I(A \cup B) = I(A) \oplus I(B).$$

We do not know, of course, whether such an operation $\oplus$ even exists and, if yes, whether there are more such operations and, if yes, how one such operation differs from the another.

Note that the domain of operation $\oplus$ is $\mathbb{R}_0^+ \times \mathbb{R}_0^+$ (it suffices that $\oplus$ is defined only for pairs of non negative numbers).
Let us make a list of required properties of information:

1. $I(A) \geq 0$ for all $A \in \mathcal{A}$ (1.2)
2. $I(\Omega) = 0$ (1.3)
3. If $A \in \mathcal{A},\ B \in \mathcal{A},\ A \cap B = \emptyset,$ then $I(A \cup B) = I(A) \oplus I(B)$ (1.4)
4. If $A_n \nearrow A = \bigcup_{i=1}^{\infty} A_i,$ or $A_n \searrow A = \bigcap_{i=1}^{\infty} A_i,$ then $I(A_n) \to I(A).$ (1.5)

Property 1. says that the amount of information is non-negative number, property 2. says that the message "*Event Ω occurred.*" carries none information. Property 3. states how the information of disjoint union of events can be calculated using informations of both events and operation $\oplus$, and the last property 4. says[4] that the information is in certain sense "continuous" on $\mathcal{A}$.

Let A, B be two events with informations $I(A)$, $I(B)$. It can happen, that the occurence of one of them gives no information about the other. In this case the information $I(A \cap B)$ of the event $A \cap B$ equals to the sum of informations of both events. This is the motivation for the following definition.

Definition 1.2. The events A, B are **independent** if it holds

$$I(A \cap B) = I(A) + I(B)\ . \tag{1.6}$$

Let us make a list of required properties of operation $\oplus$:
Let $x,\ y,\ z \in \mathbb{R}_0^+$.

1. $x \oplus y = y \oplus x$ (1.7)
2. $(x \oplus y) \oplus z\ = x \oplus (y \oplus z)$ (1.8)
3. $I(A) \oplus I(A^C) = 0$ (1.9)
4. $\oplus : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a continuous function of two variables (1.10)
5. $(x + z) \oplus (y + z) = (x \oplus y) + z$ (1.11)

Properties 1 and 2 follow from the commutativity and the associativity of set operation union. Property 3 can be derived form the requirement $I(\Omega) = 0$ by the following sequence of identities:

$$0 = I(\Omega) = I(A \cup A^C) = I(A) \oplus I(A^C)$$

The property 4 – continuity – is a natural requirement following from the requirement (1.5).

It remains to explain the requirement 5. Let A, B, C are three events such that A, B are disjointm, and A, C are independent, and B, C are independent.

If the message "*Event A occurred.*" says nothing about the event C and the message "*Event B occurred.*" says nothing about the event C then also the message "*Event $A \cup B$ occurred.*" says nothing about the event C. Thus events $A \cup B$ and C are independent.

[4] The notation $A_n \nearrow A$ means that $A_1 \subseteq A_2 \subseteq A_3, \dots$ and $A = \bigcup_{i=1}^{\infty} A_i$. Similarly $A_n \searrow A$ means that $A_1 \supseteq A_2 \supseteq A_3, \dots$ and $A = \bigcap_{i=1}^{\infty} A_i$. $I(A_n) \to I(A)$ means that $\lim_{n\to\infty} I(A_n) = I(A)$.

Denote $x = I(A)$, $y = I(B)$, $z = I(C)$ and calculate the information $I\left[(A \cup B) \cap C)\right]$

$$I\left[(A \cup B) \cap C)\right] = I(A \cup B) + I(C) = I(A) \oplus I(B) + I(C) = x \oplus y + z \quad (1.12)$$

$$\begin{aligned} I\left[(A \cup B) \cap C)\right] = I\left[(A \cap C) \cup (B \cap C)\right] = I(A \cap C) \oplus I(B \cap C) = \\ = \left[I(A) + I(C)\right] \oplus \left[I(B) + I(C)\right] = (x + z) \oplus (y + z) \quad (1.13) \end{aligned}$$

The property 5 follows from comparing of right hand sides of (1.12), (1.13).

Theorem 1.1. *Let a binary operation $\oplus$ on the set $\mathbb{R}_0^+$ fulfills axioms* (1.7) *till* (1.11)*. Then*

$$\begin{aligned} &\text{either} \quad \forall x, y \in \mathbb{R}_0^+ && x \oplus y = \min\{x, y\}, && (1.14) \\ &\text{or} \quad \exists k > 0\ \forall x, y \in \mathbb{R}_0^+ && x \oplus y = -k \log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right). && (1.15) \end{aligned}$$

Proof. The proof of this theorem is complicated, the reader can find it in [4]. ■

It is interesting that (1.14) is the limit case of (1.15) for $k \to 0+$.
First let $x = y$ and then $\min\{x, y\} = x$. Then

$$\begin{aligned} -k \log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right) = -k \log_2\left(2.2^{-\frac{x}{k}}\right) = \\ = -k \log_2\left(2^{(-\frac{x}{k}+1)}\right) = -k.(-\frac{x}{k} + 1) = x - k \end{aligned}$$

Now it is seen that the last expression converges toward x for $k \to 0^+$. Let $x > y$ then $\min\{x, y\} = y$. It holds:

$$-k \log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right) = -k \log_2\left(2^{-\frac{y}{k}}.(2^{\frac{y-x}{k}} + 1)\right) = y - k. \log_2\left(2^{\frac{y-x}{k}} + 1\right)$$

To prove the theorem it suffices to show that the second term of the last difference tends to 0 for $k \to 0^+$. The application of l'Hospital rule gives

$$\begin{aligned} \lim_{k \to 0^+} k. \log_2\left(2^{\frac{y-x}{k}} + 1\right) = \lim_{k \to 0^+} \frac{\log_2\left(2^{\frac{y-x}{k}} + 1\right)}{\frac{1}{k}} = \\ = \lim_{k \to 0^+} \frac{\frac{2^{(y-x)/k}.\ln(2).(y-x)}{(2^{(y-x)/k}+1)/k^2}}{\frac{1}{k^2}} = \ln(2)(y - x). \lim_{k \to 0^+} \frac{2^{(y-x)/k}}{2^{(y-x)/k} + 1} = 0 \end{aligned}$$

since $(y - x) < 0$, $(y - x)/k \to -\infty$ for $k \to 0^+$, and thus $2^{(y-x)/k} \to 0$.
Therefore $\lim_{k \to 0^+} -k \log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right) = \min\{x, y\}$.

Theorem 1.2. *Let $x \oplus y = -k\log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right)$ for all nonnegative real x, y. Let $x_1, x_2, \dots, x_n$ are nonnegative real numbers. Then*

$$\bigoplus_{i=1}^{n} x_i = x_1 \oplus x_2 \oplus \cdots \oplus x_n = -k\log_2\left(2^{-\frac{x_1}{k}} + 2^{-\frac{x_2}{k}} + \cdots + 2^{-\frac{x_n}{k}}\right) \tag{1.16}$$

Proof. The proof by mathematical induction on n is left for the reader. ■

1.2 Elementary definition of information

Having defined the operation $\oplus$ we can try to introduce the information in similar way as in the case of elementary definition of probability.

Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ be a partition of the sample space Ω, into n events with equal information, i. e., let

$$1.\quad \Omega = \bigcup_{i=1}^{n} A_i, \text{ where } A_i \cap A_j = \emptyset \qquad \text{for } i \neq j \tag{1.17}$$

$$2.\quad I(A_1) = I(A_2) = \cdots = I(A_n) = a \quad \text{for } i \neq j \tag{1.18}$$

We want to evaluate the value of a. It follows from (1.17), (1.18):

$$0 = I(\Omega) = I(A_1) \oplus I(A_2) \oplus \cdots \oplus I(A_n) = \underbrace{a \oplus a \oplus \cdots \oplus a}_{n-\text{times}} = \bigoplus_{i=1}^{n} a \tag{1.19}$$

$$0 = \bigoplus_{i=1}^{n} a = \\ = \begin{cases} \min\{a, a, \dots, a\} = a & \text{if } x \oplus y = \min\{x, y\} \\ -k\log_2\left(\underbrace{2^{-\frac{a}{k}} + \cdots + 2^{-\frac{a}{k}}}_{n-\text{times}}\right) & \text{if } x \oplus y = -k\log_2\left(2^{-\frac{x}{k}} + 2^{-\frac{y}{k}}\right) \end{cases} \tag{1.20}$$

For the first case $\bigoplus_{i=1}^{n} = a = 0$ and hence the information of every event of the partition $\{A_1, A_2, \dots, A_n\}$ is zero. This is not an interesting result and there is no reason to deal with it further.

For the second case

$$\bigoplus_{i=1}^{n} a = -k\log_2\left(\underbrace{2^{-\frac{a}{k}} + \cdots + 2^{-\frac{a}{k}}}_{n-\text{times}}\right) = -k\log_2\left(n.2^{-a/k}\right) = a - k\log_2(n) = 0$$

From the last expression it follows:

$$a = k.\log_2(n) = -k.\log_2\left(\frac{1}{n}\right) \tag{1.21}$$

Let the event A be union of m mutually different events $A_{i_1}, A_{i_2}, \dots, A_{i_m}$, $A_{i_k} \in \mathbf{A}$ for $k = 1, 2, \dots m$. Then

$$\begin{aligned} I(A) &= I(A_{i_1}) \oplus I(A_{i_2}) \oplus \cdots \oplus I(A_{i_m}) = \underbrace{a \oplus a \oplus \cdots \oplus a}_{m\text{–times}} = \\ &= -k.\log_2\left(\underbrace{2^{-a/k} + 2^{-a/k} + \cdots + 2^{-a/k}}_{m\text{–times}}\right) = -k\log_2\left(m.2^{-a/k}\right) = \\ &= -k.\log_2(m) - k.\log_2\left(2^{-a/k}\right) = -k.\log_2(m) - k.\left(-a/k\right) = \\ &= -k.\log_2(m) + a \ = -k.\log_2(m) + k.\log_2(n) = \\ &= k.\log_2\left(\frac{n}{m}\right) = -k.\log_2\left(\frac{m}{n}\right) \end{aligned} \tag{1.22}$$

Theorem 1.3. *Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ be a partition of the sample space Ω into n events with equal information. Then it holds for the information $I(A_i)$ of every event A_i $i = 1, 2, \dots, n$:*

$$I(A_i) = -k\log_2\frac{1}{n}. \tag{1.23}$$

Let $A = A_{i_1} \cup A_{i_2} \cup \cdots \cup A_{i_m}$ be an union of m mutually different events of partition $\mathbf{A}$, i. e., $A_{i_k} \in \mathbf{A}$, $A_{i_k} \neq A_{i_l}$ for $k \neq l$. Let $I(A)$ be the information of A. Then:

$$I(A) = -k\log_2\frac{m}{n}. \tag{1.24}$$

Let us focus our attention to an interesting analogy with elementary definition of probability. If the sample space Ω is partitioned into n disjoint events $A_1, A_2, \dots, A_n$ with equal probability p then this probability can be calculated from the equation $\sum_{i=1}^{n} p = n.p = 1$ and hence $P(A_i) = p = 1/n$. If a set A is a disjoint union of m sets of partition $\mathbf{A}$ then its probability is $P(A) = m/n$.

When introducing the information, information $a = I(A_i)$ of every event A_i is calculated from the equation (1.20) from where we obtain $I(A_i) = a = -k.\log_2(1/n)$. The information of a set A which is a disjoint union of m events of the partition $\mathbf{A}$ is $I(A) = -k.\log_2(m/n)$.

Now it is necessary to set up the constant k. This depends on the choice of the unit of information. Different values of parameter k correspond to different units of information. (The numerical value of distance depends on chosen units of length – meters, kilometers, miles, yards, etc.)

When converting logarithms to base a to logarithms to base b we can use the following well known formula:

$$\log_b(x) = \log_b(a).\log_a(x) = \frac{1}{\log_a(b)}.\log_a(x). \tag{1.25}$$

So the constant k and the logarithm to base 2 could be replaced by the logarithm to arbitrary base in formulas (1.21), (1.22). This was indeed used by several authors namely in the older literature on the information theory where sometimes decimal logarithm appears in evaluating the information.

The following reasoning can by useful for determining the constant k. Computer technique and digital transmission technique use for data transfer in most cases binary digits 0 and 1. It would be natural if such a digit would carry one unit of information. Such unit of information is called 1 bit.

Let $\Omega = \{0, 1\}$ be the set of values of a binary digit, let $A_1 = \{0\}$, $A_2 = \{1\}$. Let both the sets A_1, A_2 carry information a. We want that $I(A_1) = I(A_2) = a = 1$. It holds $1 = a = k.\log_2(2) = k$ according to (1.21).

If we want that (1.21) expresses the amount of information in bits we have to set $k = 1$. We will suppose from now on that information is measured in bits and hence $k = 1$.

1.3 Information as a function of probability

When introducing the information in elementary way, we have shown that the information of an event A which is disjoint union of m events of a partition $\Omega = A_1 \cup A_2 \cup \cdots \cup A_n$ is $I(A) = -\log_2(m/n)$ while the probability of the event A is $P(A) = m/n$. In this case we could write $I(A) = -\log_2(P(A))$. In this section we will try to define the information from another point of view by means of probability.

Suppose that the information $I(A)$ of an event A depends only on its probability $P(A)$, i. e., $I(A) = f(P(A))$ and that the function f does not depend on the corresponding probability space $(\Omega, \mathcal{A}, P)$.

We will study now what functions are eligible to stand in expression $I(A) = f(P(A))$. We will show that the only possible function is the function $f(x) = -k.\log_2(x)$. We will use the method from [5].

First, we will give a generalized definition of independence of finite o infinite sequence of events.

Definition 1.3. The finite or infinite sequence of events $\{A_n\}_n$ is called **sequence of (informational) independent events** if for every finite subsequence $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ holds

$$I\left(\bigcap_{k=1}^{m} A_{i_k}\right) = \sum_{k=1}^{m} I\left(A_{i_k}\right). \tag{1.26}$$

In order that information may have "reasonable" properties, it is necessary to postulate that the function f is continuous, and that events which are independent in probability sense are independent in information sense, too, and vice versa.

This means that for a sequence of independent events $A_1, A_2, \ldots, A_n$ it holds

$$I(A_1 \cap A_2 \cap \cdots \cap A_n) = f(P(A_1 \cap A_2 \cap \cdots \cap A_n)) = f\left(\prod_{i=1}^{n} P(A_i)\right) \tag{1.27}$$

and at the same time

$$I(A_1 \cap A_2 \cap \cdots \cap A_n) = \sum_{i=1}^{n} I(A_i) = \sum_{i=1}^{n} f\left(P(A_i)\right) \tag{1.28}$$

Left hand sides of both last expressions are the same, therefore

$$f\left(\prod_{i=1}^{n} P(A_i)\right) = \sum_{i=1}^{n} f\left(P(A_i)\right) \tag{1.29}$$

Let the probabilities of all events $A_1, A_2, \ldots, A_n$ are the same, let $P(A_i) = x$. Then $f(x^n) = n.f(x)$ for all $x \in \langle 0, 1\rangle$. For $x = 1/2$ we have

$$f(x^m) = f\left(\frac{1}{2^m}\right) = m.f\left(\frac{1}{2}\right). \tag{1.30}$$

For $x = \dfrac{1}{2^{1/n}}$ it is $f(x^n) = f\left(\left(\dfrac{1}{2^{1/n}}\right)^n\right) = f\left(\dfrac{1}{2}\right) = n.f(x) = n.f\left(\dfrac{1}{2^{1/n}}\right)$, from which we have

$$f\left(\frac{1}{2^{1/n}}\right) = \frac{1}{n}.f\left(\frac{1}{2}\right) \tag{1.31}$$

Finally, for $x = \dfrac{1}{2^{1/n}}$ it holds

$$f(x^m) = f\left(\frac{1}{2^{m/n}}\right) = m.f(x) = m.f\left(\frac{1}{2^{1/n}}\right) = \frac{m}{n}.f\left(\frac{1}{2}\right),$$

and hence

$$f\left(\frac{1}{2^{m/n}}\right) = \frac{m}{n}.f\left(\frac{1}{2}\right) \tag{1.32}$$

Since (1.32) holds for all positive integers m, n and since the function f is continuous it holds

$$f\left(\frac{1}{2^x}\right) = x.f\left(\frac{1}{2}\right) \text{ for all real numbers } x \in \langle 0, \infty).$$

Let us create an auxiliary function g: $g(x) = f(x) + f\left(\dfrac{1}{2}\right).\log_2(x)$.

Then it holds:

$$\begin{aligned} g(x) &= f(x) + f\left(\frac{1}{2}\right).\log_2(x) = f\left(2^{\log_2(x)}\right) + f\left(\frac{1}{2}\right).\log_2(x) = \\ &= f\left(\frac{1}{2^{-\log_2(x)}}\right) + f\left(\frac{1}{2}\right).\log_2(x) = \end{aligned}$$

$$= \quad -\log_2(x).f\left(\frac{1}{2}\right)+f\left(\frac{1}{2}\right).\log_2(x)=0$$

Function $g(x)=f(x)+f\left(\frac{1}{2}\right).\log_2(x)$ is identically 0, and that is why

$$f(x)=-f\left(\frac{1}{2}\right).\log_2(x)=-k.\log_2(x) \tag{1.33}$$

Using the function f from the last formula (1.33) in the place of f in $I(A)=f(P(A))$ we get the famous **Shannon – Hartley formula:**

$$I(A)=-k.\log_2(P(A)) \tag{1.34}$$

The coefficient k depends on the chosen unit of information similarly as in the case of elementary way of introducing information.

Let $\Omega=\{0,1\}$ be the set of possible values of binary digit, $A_1=\{0\}$, $A_2=\{1\}$, let the probability of both sets is the same $P(A_1)=P(A_2)=1/2$. From the Shannon – Hartley formula it follows that both sets carry the same amount of information – we would like that this amount is the unit of information. That is why it has to hold:

$$1=f\left(\frac{1}{2}\right)=-k.\log_2\left(\frac{1}{2}\right)=k,$$

and hence $k=1$. We can see that this second way leads to the same result as the elementary way of introducing information.

Most of textbooks on the information theory start with displaying the Shannon-Hartley formula from which many properties of information are derived. The reader may ask the question why the amount of information is defined just by this formula and whether it is possible to measure information using another expression. We have shown that several ways of introducing information leads to the same unique result and that there is no other way how to do it.

Chapter 2

Entropy

2.1 Experiments

If we receive the message "*Event A occurred.*", we get with it $-\log_2 P(A)$ bits of information, where $P(A)$ is the probability of the event A. Let $(\Omega, \mathcal{A}, P)$ be a probability space. Imagine that the sample space Ω is partitioned to a finite number n of disjoint events $A_1, A_2, \dots, A_n$. Perform the following experiment: Choose at random $\omega \in \Omega$ and determine A_i such that $\omega \in A_i$, i. e., determine which event A_i occurred.

We have an uncertainty about its result before executing the experiment. After executing the experiment the result is known and our uncertainty disappears. Hence we can say that the amount of uncertainty before the experiment equals to the amount of information delivered by execution of the experiment.

We can organize the experiment in several cases – we can determine the events of the partition of the sample space Ω. We can do it in order to maximize the information obtained after the execution of the experiment.

We choose to partition the set Ω into such events that every one corresponds to one result of the experiment, according to possible outcomes of available measuring technique. A properly organized experiment is one of crucial prerequisites of success in many branches of human activities.

Definition 2.1. Let $(\Omega, \mathcal{A}, P)$ be a probability space. **Finite measurable partition of the sample space** Ω is a finite set of events $\{A_1, A_2, \dots, A_n\}$ such that $A_i \in \mathcal{A}$ for $i = 1, 2, \dots, n$, $\bigcup_{i=1}^{n} A_i = \Omega$ a $A_i \cap A_j = \emptyset$ for $i \neq j$.

The finite measurable partition $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$ of the sample space Ω is also called **experiment**.

Some literature requires weaker assumptions on the sets $\{A_1, A_2, \dots, A_n\}$ of experiment $\mathbf{P}$, namely $P\left(\bigcup_{i=1}^{n} A_i\right) = 1$ and $P(A_i \cap A_j) = 0$ for $i \neq j$. Both the approaches are the same and their results are equivalent.

Every experiment should be designed in such a way that its execution gives as much information as possible. If we want to know the departure time of IC train Tatran, we can get more information from the answer to the question "*What is the hour and the minute of departure of IC train Tatran from Žilina to Bratislava?*" than from the answer to the question "*Does IC train Tatran depart from Žilina to Bratislava before noon or after noon?*". The first question partitions the space Ω into 1440 possible events, the second to only 2 events.

Both questions define two experiments $\mathbf{P}_1$, $\mathbf{P}_2$. Suppose that all events of the experiment $\mathbf{P}_1$ have the same probability equal to 1/1440 and both events of the experiment $\mathbf{P}_2$ have probability 1/2. Every event of $\mathbf{P}_1$ carries with it $-\log_2(1/1440) = 10.49$ bits of information, both events of $\mathbf{P}_2$ carry $-\log_2(1/2) = 1$ bit of information.

Regardless of the result of the experiment $\mathbf{P}_1$ performing this experiment gives 10.49 bits of information while experiment $\mathbf{P}_2$ gives 1 bit of information.

We will consider the amount of information obtained by executing an experiment to be a measure of its uncertainty also called entropy of the experiment.

2.2 Shannon's definition of entropy

In this stage we know how to define the uncertainty – entropy $H(\mathbf{P})$ of an experiment $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$ if all its events A_i have the same probability $1/n$ – in this case:

$$H(\mathbf{P}) = -\log_2(1/n).$$

But what to do in the case when events of the experiment have different probabilities? Imagine that $\Omega = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$, $P(A_1) = 0.1$, $P(A_2) = 0.9$.

If A_1 is the result we get $I(A_1) = -\log_2(0.1) = 3.32$ bits of information, but if the outcome is A_2 we get only $I(A_2) = -\log_2(0.9) = 0.15$ bits of information. Thus the obtained information depends on the result of the experiment. In the case of A_1 the obtained amount of information is large but it happens only in 10% of trials – in 90% of trials the outcome is A_2 and the gained information is small.

Imagine now that we execute the experiment many times – e. g., 100 times. Approximately in 10 trials we get 3.32 bits of information, and approximately in 90 trials we get 0.15 bits of information. The total amount of information can be calculated as

$$10 \times 3.32 + 90 \times 0.15 = 33.2 + 13.5 = 46.7$$

bits.

The average information (per one execution of the experiment) is $46.7/100 = 0.467$ bits. One possibility how to define the entropy of experiment in general case (case of different probabilities of events of the experiment) is to define it as the mean value of information.

Definition 2.2. Shannon's definition of entropy. Let $(\Omega, \mathcal{A}, P)$ be a probability space. Let $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$ be an experiment. The **entropy** $H(\mathbf{P})$ **of the experiment P** is the mean of discrete random variable X whose value is $I(A_i)$ for all $\omega \in A_i$,[1] i. e.:

$$H(\mathbf{P}) = \sum_{i=1}^{n} I(A_i)P(A_i) = -\sum_{i=1}^{n} P(A_i).\log_2 P(A_i) \tag{2.1}$$

A rigorous reader could now ask what will happen if there is an event A_i in the experiment $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$ with $P(A_i) = 0$. Then the expression $-P(A_i).\log_2 P(A_i)$ is of the type $0 \log_2 0$ – and such an expression is not defined. Nevertheless it holds:

$$\lim_{x \to 0+} x \log_2(x) = 0,$$

and thus it is natural to define the function $\eta(x)$ as follows:

$$\eta(x) = \begin{cases} -x.\log_2(x) & \text{if } x > 0 \\ 0 & \text{if } x = 0. \end{cases}$$

Then the Shannon entropy formula should be in the form:

$$H(\mathbf{P}) = \sum_{i=1}^{n} \eta(P(A_i)).$$

[1] The random variable X could be defined exactly

$$X(\omega) = -\sum_{i=1}^{n} \chi_{A_i}(\omega).\log_2 P(A_i),$$

where $\chi_{A_i}(\omega)$ is the set indicator of A_i, i. e., $\chi_{A_i}(\omega) = 1$ if and only if $\omega \in A_i$, otherwise $\chi_{A_i}(\omega) = 0$.

However, the last notation slightly conceals the form of nonzero terms of formula and that is why we will use the form (2.1) with the following convention:

Agreement 2.1. From now on, we will suppose that the expression $0.\log_2(0)$ is defined and that

$$0.\log_2(0) = 0.$$

The terms of the type $0.\log_2(0)$ in the formula (2.1) express the fact that adding a set with zero probability to an experiment $\mathbf{P}$ results in a new experiment $\mathbf{P}'$ which entropy is the same as that of $\mathbf{P}$.

2.3 Axiomatic definition of entropy

The procedure of introducing the Shannon's formula in preceding section was simple and concrete. However, not all authors were satisfied with it. Some authors would like to introduce the entropy without the notion of information $I(A)$ of individual event A. This section will follow the procedure of introducing the notion of entropy without making use of that of information.

Let $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$ be an experiment, let $p_1 = P(A_1)$, $p_2 = P(A_2)$, $\dots$, $p_n = P(A_n)$, let H be a (in this stage unknown) function expressing the uncertainty of $\mathbf{P}$. Suppose that the function H does not depend on any particular type of the probability space $(\Omega, \mathcal{A}, P)$, but it depends only on numbers $p_1, p_2, \dots, p_n$:

$$H(\mathbf{P}) = H(p_1, p_2, \dots, p_n).$$

Function $H(p_1, p_2, \dots, p_n)$ should have several natural properties arising from its purpose. It is possible to formulate these properties as axioms from which it is possible to derive another properties and even the particular form of the function H.

There are several axiomatic systems for this purpose, we will work with that of Fadejev from year 1956:

AF0: Function $y = H(p_1, p_2, \dots, p_n)$ is defined for all n and for all $p_1 \geq 0,\ p_2 \geq 0, \dots,\ p_n \geq 0$ such that $\sum_{i=1}^{n} p_i = 1$ and takes real values.

AF1: $y = H(p, 1-p)$ is a function of one variable continuous on $p \in \langle 0, 1 \rangle$.

AF2: $y = H(p_1, p_2, \dots, p_n)$ is a symmetric function, i. e., it holds:

$$H(p_{\pi[1]}, p_{\pi[2]}, \dots, p_{\pi[n]}) = H(p_1, p_2, \dots, p_n). \tag{2.2}$$

for an arbitrary permutation π of numbers $1, 2, \dots, n$.

AF3: Branching principle.

If $p_n = q_1 + q_2 > 0$, $q_1 \geq 0$, $q_2 \geq 0$, then

$$H(p_1, p_2, \dots, p_{n-1}, \underbrace{q_1, q_2}_{p_n}) =$$

$$= H(p_1, p_2, \dots, p_{n-1}, p_n) + p_n.H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}\right) \quad (2.3)$$

We extend the list of these axioms with so called Shannon's axiom. Denote:

$$F(n) = H\left(\underbrace{\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}}_{n\text{–times}}\right) \quad (2.4)$$

The Shannon's axiom says:

AS4: If $m < n$, then $F(m) < F(n)$.

The axiom AF0 is natural – we want the entropy to exist and to be a real number for all possible experiments. The axiom AF1 expresses a natural requirement that small changes of probabilities of an experiment with two outcomes result in small changes of the uncertainty of this experiment. The axiom AF2 says that the uncertainty of an experiment does not depend on the order of its events.

The axiom AF3 needs more detailed explanation. Suppose that the experiment $\mathbf{P} = \{A_1, A_2, \dots, A_{n-1}, A_n\}$ with probabilities $p_1, p_2, \dots, p_n$ is given. We define a new experiment $\mathbf{P}' = \{A_1, A_2, \dots, A_{n-1}, B_1, B_2\}$ in such a way that we divide the last event A_n of $\mathbf{P}$ into two disjoint parts B_1, B_2. Then it holds $P(B_1) + P(B_2) = P(A_n)$ for the corresponding probabilities. Denote $P(B_1) = q_1$, $P(B_2) = q_2$, then $p_n = q_1 + q_2$.

Let us try to express the increment of uncertainty of the experiment $\mathbf{P}'$ compared to uncertainty of $\mathbf{P}$. If the event A_n occurs then the question about the result of experiment $\mathbf{P}$ is fully answered but we have some additional uncertainty about the result of experiment $\mathbf{P}'$ – namely which of events B_1, B_2 occurred.

Conditional probabilities of events B_1, B_2 given A_n are $P(B_1 \cap A_n)/P(A_n) = P(B_1)/P(A_n) = q_1/p_n$, $P(B_2 \cap A_n)/P(A_n) = P(B_2)/P(A_n) = q_2/p_n$, Hence if the outcome is the event A_n the remaining uncertainty is

$$H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}\right).$$

Nevertheless, the event A_n does not occur always but only with probability p_n. That is why the division of the event A_n into two disjoint events B_1, B_2 increases the total uncertainty of $\mathbf{P}'$ compared to the uncertainty of $\mathbf{P}$ by the amount:

$$p_n.H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}\right).$$

Fadejev's axioms AF0 – AF3 are sufficient for deriving all properties and the form of the function H. The validity of Shannon's axiom can also be proved from AF0 – AF3.

The corresponding proofs using only AF0 – AF3 are slightly complicated and that is why we will use the natural Shannon's axiom. This says that if $\mathbf{P}_1$, $\mathbf{P}_2$ are two experiments, the first having m events all with probability $1/m$, the second n events all with probability $1/n$ and $m < n$ then the uncertainty of $\mathbf{P}_1$ is less then that of $\mathbf{P}_2$

Theorem 2.1. *Shannon's entropy*

$$H(\mathbf{P}) = \sum_{i=1}^{n} I(A_i)P(A_i) = -\sum_{i=1}^{n} P(A_i)\log_2 P(A_i)$$

fulfils the axioms AF0 *till* AF3 *and Shannon's axiom* AS4.

Proof. Verification of all axioms is simple and straightforward and the reader can do it easily himself. ■

Now we will prove several affirmations arising from axioms AF0 – AF3 and AS4. These affirmations will show us several interesting properties of the function H provided this function fulfills all mentioned axioms. The following theorems will lead step by step to Shannon's entropy formula. Since Shannon's entropy (2.1) fulfills all axioms by theorem 2.1, these theorems hold also for it.

Theorem 2.2. *Function $y = H(p_1, p_2, \dots, p_n)$ is continuous on the set*

$$\mathcal{Q}_n = \left\{(x_1, x_2, \dots, x_n) \mid x_i \geq 0 \text{ for } i = 1, 2 \dots, n,\ \sum_{i=1}^{n} x_i = 1\right\}.$$

Proof. Mathematical induction on m. The statement for $m = 2$ is equivalent with axiom AF1. Let the function $y = H(x_1, x_2, \dots x_m)$ be continuous on $\mathcal{Q}_m$. Let $(p_1, p_2, \dots, p_m, p_{m+1}) \in \mathcal{Q}_{m+1}$. Suppose that at least one of the numbers p_m, p_{m+1} is different from zero (otherwise we change the order of numbers p_i). Using axiom AF3 we have:

$$H(p_1, p_2, \dots, \underbrace{p_m, p_{m+1}}) = H\big(p_1, p_2, \dots, p_{m-1}, (p_m + p_{m+1})\big) + \\ + (p_m + p_{m+1}).H\left(\frac{p_m}{(p_m + p_{m+1})}, \frac{p_{m+1}}{(p_m + p_{m+1})}\right) \quad (2.5)$$

The continuity of the first term of (2.5) follows from the induction hypothesis, the continuity of the second term follows from axiom A1. ■

Theorem 2.3. $H(1, 0) = 0$.

Proof. Using axiom AF3 we can write:

$$H\left(\frac{1}{2}, \underbrace{\frac{1}{2}, 0}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}.H(1, 0) \quad (2.6)$$

Applying first axiom AF2 and then axiom AF3:

$$H\left(\frac{1}{2}, \frac{1}{2}, 0\right) = H\left(0, \underbrace{\frac{1}{2}, \frac{1}{2}}\right) = \\ = H(0, 1) + H\left(\frac{1}{2}, \frac{1}{2}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + H(1, 0) \quad (2.7)$$

Comparing left hand sides of (2.6), (2.7) we get $\frac{1}{2}.H(1, 0) = H(1, 0)$ what implies $H(1, 0) = 0$. ■

Let $\mathbf{P} = \{A_1, A_2\}$ be the experiment consisting from two events one of which is certain and the other impossible. Theorem (2.3) says, that such experiment has zero uncertainty.

Theorem 2.4. $H(p_1, p_2, \dots, p_n, 0) = H(p_1, p_2, \dots, p_n)$

Proof. At least one of numbers $p_1, p_2, \dots, p_n$ is positive. Let $p_n > 0$ (otherwise we change the order). Then using axiom AF3:

$$H(p_1, p_2, \dots, \underbrace{p_n, 0}) = H(p_1, p_2, \dots, p_n) + p_n. \underbrace{H(1, 0)}_{0} \quad (2.8)$$

■

Again one good property of entropy – it does not depend on events with zero probability.

Theorem 2.5. *Let $p_n = q_1 + q_2 + \cdots + q_m > 0$. Then*

$$H(p_1, p_2, \dots, p_{n-1}, \underbrace{q_1, q_2, \dots, q_m}_{p_n}) =$$

$$= H(p_1, p_2, \dots, p_n) + p_n.H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}, \dots, \frac{q_m}{p_n}\right) \qquad (2.9)$$

Proof. Mathematical induction on m. The statement for $m = 2$ is equivalent to the axiom AF3.
Let the statement hold for $m \geq 2$.
Set $p' = q_2 + q_3 + \cdots + q_{m+1}$, suppose that $p' > 0$ (otherwise change the order of $q_1, q_2, \dots, q_{m+1}$). By the induction hypothesis

$$H(p_1, p_2, \dots, p_{n-1}, q_1, \underbrace{q_2, \dots, q_{m+1}}_{p'=\sum_{k=2}^{m} q_k}) =$$

$$= H(p_1, p_2, \dots, p_{n-1}, \underbrace{q_1, p'}_{p_n}) + p'.H\left(\frac{q_2}{p'}, \dots, \frac{q_{m+1}}{p'}\right) =$$

$$= H(p_1, p_2, \dots, p_n) + p_n.\left[H\left(\frac{q_1}{p_n}, \frac{p'}{p_n}\right) + \frac{p'}{p_n}H\left(\frac{q_2}{p'}, \dots, \frac{q_{m+1}}{p'}\right)\right]. \qquad (2.10)$$

Again by induction hypothesis:

$$H\left(\frac{q_1}{p_n}, \underbrace{\frac{q_2}{p_n}, \dots, \frac{q_{m+1}}{p_n}}_{\frac{p'}{p_n}}\right) = H\left(\frac{q_1}{p_n}, \frac{p'}{p_n}\right) + \frac{p'}{p_n}H\left(\frac{q_2}{p'}, \dots, \frac{q_{m+1}}{p'}\right). \qquad (2.11)$$

We can see that the right hand side of (2.11) is the same as the contents of big square brackets on the right hand side of (2.10). Replacing the contents of big square brackets of (2.10) by the left hand side of (2.11) gives (2.9). ■

Theorem 2.6. *Let $q_{ij} \geq 0$ for all pairs of integers (i, j) such that $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m_i$, let $\sum_{i=1}^{n} \sum_{j=1}^{m_i} = 1$.
Let $p_i = q_{i1} + q_{i2} + \cdots + q_{im_i} > 0$ for $i = 1, 2, \dots, n$. Then*

$$H(q_{11}, q_{12} \dots q_{1m_1}, q_{21}, q_{22}, \dots, q_{2m_2}, \dots, q_{n1}, q_{n2}, \dots, q_{nm_n}) =$$

$$= H(p_1, p_2, \dots, p_n) + \sum_{i=1}^{n} p_i.H\left(\frac{q_{i1}}{p_i}, \frac{q_{i2}}{p_i}, \dots, \frac{q_{im_i}}{p_i}\right) \qquad (2.12)$$

Proof. The proof can be done by repeated application of theorem 2.5. ■

Theorem 2.7. *Denote* $F(n) = H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$. *Then* $F(mn) = F(m) + F(n)$.

Proof. From theorem 2.6 it follows:

$$\begin{aligned}
F(mn) &= H\Bigg(\underbrace{\underbrace{\frac{1}{mn}, \dots, \frac{1}{mn}}_{m\text{-times}}, \dots \underbrace{\frac{1}{mn}, \dots, \frac{1}{mn}}_{m\text{-times}}}_{n\text{-times}}\Bigg) = \\
&= H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) + \sum_{i-1}^{n} \frac{1}{n} H\left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right) = \\
&= H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) + H\left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right) = F(n) + F(m)
\end{aligned}$$

■

Theorem 2.8. *Let* $F(n) = H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$. *Then* $F(n) = c.\log_2(n)$.

Proof. We show by mathematical induction that it holds $F(n^k) = k.F(n)$ for $k = 1, 2, \dots$. By theorem 2.7 it holds: $F(m.n) = F(m) + F(n)$. Especially for $m = n$ is $F(n^2) = 2.F(n)$, $F(n^k) = F(n^{k-1}.n) = F(n^{k-1}).F(n) = (k-1).F(n) + F(n) = k.F(n)$. Therefore we can write:

$$F(n^k) = k.F(n) \quad \text{for } k = 1, 2, \dots \tag{2.13}$$

Formula (2.13) has several consequences:

1. $F(1) = F(1^2) = 2.F(1)$ What implies $F(1) = 0$, and hence
   $F(1) = c.\log_2(1)$ for every real c.

2. Since the function F is strictly increasing by axiom AS4, it holds for every integer $m > 1$ $F(m) > F(1) = 0$.

Let us have two integers $m > 1$, $n > 1$ and an arbitrary large integer $K > 0$. Then there exists an integer $k > 0$ such that

$$m^k \le n^K < m^{k+1}. \tag{2.14}$$

Since F is an increasing function

$$F(m^k) \leq F(n^K) < F(m^{k+1}).$$

Applying (2.13) gives:

$$k.F(m) \leq K.F(n) < (k+1).F(m).$$

Divide the last inequality by $K.F(m)$ ($F(m) > 0$, therefore this division is allowed and it does not change inequalities):

$$\frac{k}{K} \leq \frac{F(n)}{F(m)} < \frac{k+1}{K}. \tag{2.15}$$

Since (2.14) holds we can get by the same reasoning:

$$\log_2(m^k) \leq \log_2(n^K) < \log_2(m^{k+1})$$

$$k.\log_2(m) \leq K.\log_2(n) < (k+1).\log_2(m),$$

and hence (remember that $m > 1$ and therefore $\log_2(m) > 0$)

$$\frac{k}{K} \leq \frac{\log_2(n)}{\log_2(m)} < \frac{k+1}{K}. \tag{2.16}$$

Comparing (2.15) and (2.16) we can see that both fractions $\dfrac{F(n)}{F(m)}, \dfrac{\log_2(n)}{\log_2(m)}$ are elements of interval $\left\langle \dfrac{k}{K}, \dfrac{k+1}{K} \right)$ whose length is $\dfrac{1}{K}$ and then

$$\left| \frac{F(n)}{F(m)} - \frac{\log_2(n)}{\log_2(m)} \right| < \frac{1}{K}. \tag{2.17}$$

The left hand size of (2.17) does not depend on K. Since the whole procedure can be repeated for arbitrary large integer K, the formula (2.17) holds for arbitrary K from which it follows:

$$\frac{F(n)}{F(m)} = \frac{\log_2(n)}{\log_2(m)},$$

and hence

$$F(n) = F(m).\frac{\log_2(n)}{\log_2(m)} = \left(\frac{F(m)}{\log_2(m)} \right) \log_2(n). \tag{2.18}$$

Fixate m and set $c = \dfrac{F(m)}{\log_2(m)}$ in (2.18). We get $F(n) = c.\log_2(n)$. ■

Theorem 2.9. *Let* $p_1 \geq 0,\ p_2 \geq 0, \dots,\ p_n \geq 0,\ \sum_{i=1}^{n} p_i = 1$. *Then there exists a real number* $c > 0$ *such that*

$$H(p_1, p_2, \dots, p_n) = -c. \sum_{i=1}^{n} p_i. \log_2(p_i). \tag{2.19}$$

Proof. We will prove (2.19) first for rational numbers $p_1, p_2, \dots, p_n$ – i. e., when every p_i is a ratio of two integers. Let s be the common denominator of all fractions $p_1, p_2, \dots, p_n$, let $p_i = \dfrac{q_i}{s}$ for $i = 1, 2, \dots, n$. We can write by (2.12) of theorem 2.6:

$$H\Bigg(\underbrace{\frac{1}{s}, \dots, \frac{1}{s},}_{q_1\text{-times}}\ \underbrace{\frac{1}{s}, \dots, \frac{1}{s},}_{q_2\text{-times}}\ \dots\ \underbrace{\frac{1}{s}, \dots, \frac{1}{s},}_{q_n\text{-times}}\Bigg) =$$

$$= H(p_1, p_2, \dots, p_n) + \sum_{i=1}^{n} p_i.H\left(\frac{1}{q_i}, \frac{1}{q_i} \dots, \frac{1}{q_i}\right) =$$

$$= H(p_1, p_2, \dots, p_n) + \sum_{i=1}^{n} p_i.F(q_i) =$$

$$= H(p_1, p_2, \dots, p_n) + c. \sum_{i=1}^{n} p_i. \log_2(q_i). \tag{2.20}$$

The left hand side of (2.20) equals $F(s) = c. \log_2(s)$, therefore we can write:

$$H(p_1, p_2, \dots, p_n) = c \log_2(s) - c. \sum_{i=1}^{n} p_i \log_2(q_i) =$$

$$= c \log_2(s) \sum_{i=1}^{n} p_i - c \sum_{i=1}^{n} p_i \log_2(q_i) = c \sum_{i=1}^{n} p_i \log_2(s) - c \sum_{i=1}^{n} p_i \log_2(q_i) =$$

$$= -c \sum_{i=1}^{n} p_i [\log_2(q_i) - \log_2(s)] =$$

$$= -c \sum_{i=1}^{n} p_i \log_2\left(\frac{q_i}{s}\right) = -c \sum_{i=1}^{n} p_i \log_2(p_i). \tag{2.21}$$

The function H is continuous and (2.21) holds for all rational numbers $p_1 \geq 0$, $p_2 \geq 0, \dots,\ p_n \geq 0$ such that $\sum_{i=1}^{n} p_i = 1$, therefore (2.21) has to hold for all rational arguments p_i fulfilling the same conditions. ■

It remains to determine the constant c. In order to comply with the requirement that the entropy of an experiment with two events with equal probabilities equals 1, it has to hold $H(1/2,1/2)=1$ what implies:

$$1=H\left(\frac{1}{2},\frac{1}{2}\right)=-c.\left[\frac{1}{2}.\log_2\left(\frac{1}{2}\right)+\frac{1}{2}.\log_2\left(\frac{1}{2}\right)\right]=-c.\left(-\frac{1}{2}-\frac{1}{2}\right)=c$$

We can see that axiomatic definition of entropy leads to the same Shannon entropic formula that we have obtained as the mean value of discrete random variable of information.

2.4 Another properties of entropy

Theorem 2.10. *Let $p_i>0$, $q_i>0$, $\sum_{i=1}^n p_i=1$, $\sum_{i=1}^n q_i=1$ for $i=1,2,\dots,n$. Then*

$$-\sum_{i=1}^{n} p_i\log_2(p_i)\le-\sum_{i=1}^{n} p_i\log_2(q_i), \tag{2.22}$$

with equality if and only if $p_i=q_i$ for all $i=1,2,\dots,n$.

Proof. First we prove the following inequality:

$$\ln(1+y)\le y \quad \text{for} \quad y>-1$$

Set $g(y)=\ln(1+y)-y$ and search for extremes of g. It holds $g'(y)=\dfrac{1}{1+y}-1$, $g''(y)=-\dfrac{1}{(1+y)^2}\le 0$. The equation $g'(y)=0$ has unique solution $y=0$ and $g''(0)=-1<0$. Function $g(y)$ takes its global maximum in the point $y=0$. That is why $g(y)\le 0$, i. e., $\ln(1+y)-y\le 0$ and hence $\ln(1+y)\le y$ with equality if and only if $y=0$. Substituting y by $x-1$ in (2.22) we get

$$\ln(x)\le x-1 \quad \text{for} \quad x>0, \tag{2.23}$$

with equality if and only if $x=1$.
Now we use substitution $x=\dfrac{q_i}{p_i}$ in (2.23). We get step by step:

$$\begin{aligned}\ln(q_i)-\ln(p_i)&\le\frac{q_i}{p_i}-1\\ p_i\ln(q_i)-p_i\ln(p_i)&\le q_i-p_i\end{aligned}$$

$$\begin{aligned}
-p_i \ln(p_i) &\leq -p_i \ln(q_i) + q_i - p_i \\
-\sum_{i=1}^{n} p_i \ln(p_i) &\leq -\sum_{i=1}^{n} p_i \ln(q_i) + \underbrace{\sum_{i=1}^{n} q_i}_{=1} - \underbrace{\sum_{i=1}^{n} p_i}_{=1} \\
-\sum_{i=1}^{n} p_i \frac{\ln(p_i)}{\ln(2)} &\leq -\sum_{i=1}^{n} p_i \frac{\ln(q_i)}{\ln(2)} \\
-\sum_{i=1}^{n} p_i \log_2(p_i) &\leq -\sum_{i=1}^{n} p_i \log_2(q_i),
\end{aligned}$$

with equalities in the first three rows if and only if $p_i = q_i$ and with equalities in the last three rows if and only if $p_i = q_i$ for all $i = 1, 2, \dots, n$. ■

Theorem 2.11. *Let $n > 1$ be a fixed integer. The function*

$$H(p_1, p_2, \dots, p_n) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

takes its maximum for $p_1 = p_2 = \cdots = p_n = 1/n$.

Proof. Let $p_1, p_2, \dots, p_n$ be real numbers $p_i \geq 0$ for $i = 1, 2, \dots, n$, $\sum_{i=1}^{n} p_i = 1$ and set $q_1 = q_2 = \cdots = q_n = \dfrac{1}{n}$ into (2.22). Then

$$\begin{aligned}
H(p_1, p_2, \dots, p_n) = -\sum_{i=1}^{n} p_i \log_2(p_i) \leq -\sum_{i=1}^{n} p_i \log_2\left(\frac{1}{n}\right) = \\
= -\log_2\left(\frac{1}{n}\right) \cdot \sum_{i=1}^{n} p_i = -\log_2(\frac{1}{n}) = \log_2 n = H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)
\end{aligned}$$

■

2.5 Application of entropy in selected problem solving

Let $(\Omega, \mathcal{A}, P)$ be a probability space. Suppose that an elementary event $\omega \in \Omega$ occurred. We have no possibility (and no need for it, too) to determine the exact elementary event ω, it is enough to determine the event B_i of the experiment $\mathbf{B} = \{B_1, B_2, \dots, B_n\}$ for which $\omega \in B_i$.[2] The experiment $\mathbf{B} = \{B_1, B_2, \dots, B_n\}$ on the probability space $(\Omega, \mathcal{A}, P)$ answering the required question is called **basic experiment**

There are often problems of the type: "*Determine, using as little questions as possible, which of the events of the given basic experiment* **B** *occurred.*" Unless not specified, we expect that all events of the basic experiment have equal probabilities. Then the entropy of such experiment with n events equals to $\log_2(n)$ – i. e., execution of such experiment gives us $\log_2(n)$ bits of information.

Very often we are not able to organize the basic experiment **B** because the number of available answers to our question is limited (e. g., given by available measuring equipment). An example of limited number of possible answers is the situation when we can get only two answers "*yes*" or "*no*". If we want to get maximum information with one answer, we have to formulate the corresponding question in such a way that the probability of both answers is as close as possible to number 1/2.

Example 2.1. There are 32 pupils in a class, one of them won a literature contest. How to determine the winner using as little as possible questions with only possible answers "*yes*" or "*no*"? In the case of non-limited number of answers this problem could be solved by the basic experiment $\mathbf{B} = \{B_1, B_2, \dots, B_{32}\}$ with 32 possible outcomes and the gained information would be $\log_2(32) = 5$ bits.

Since only answers "*yes*" or "*no*" are allowed we have to replace the experiment **B** by series of experiments of the type $\mathbf{A} = \{A_1, A_2\}$ with only two events. Such experiment can give at most 1 bit of information so that at least 5 such experiments are needed to specify the winner.

If we deal with an average Slovak co-educated class we can ask a question: "*Is the winner a boy?*" This is a good question since in Slovak class the number of boys is approximately equal to the number of girls. The answer to this question gives approximately 1 bit of information.

[2] For the sake of proper ski waxing it suffices to know in which of temperature intervals $(-\infty, -12)$, $(-12, -8)$, $(-8, -4)$, $(-4, 0)$ and $(0, \infty)$ the real temperature is since we have ski waxes designed for mentioned temperature intervals.

The question "*Is John Black the winner?*" gives in average $H(1/32, 31/32) = -(1/32).\log_2(1/32) - (31/32).\log_2(31/32) = 0.20062$ bits of information. It can happen that the answer is "*yes*" and in this case we would get 5 bits of information. However, this happens only in 1 case of 32, in other cases, we get the answer "*no*" and we get only 0.0458 bits of information.

That is why it is convenient that every question divides till now not excluded pupils into two equal subsets.
Here is the procedure how to determine the winner after 5 questions: Assign the pupils integer numbers from 1 to 32.

1. Question: "*Is the winner assigned a number from* 1 *to* 16*?*" If the answer is "*yes*", we know that the winner is in the group with numbers from 1 to 16, if the answer is "*no*" the winner is in the group with numbers from 17 to 32.

2. Question "*Is the number of winner among* 8 *lowest in the group with* 16*-pupils containing the winner?*" Thus the group with 8 elements containing the winner is determined.

3. Similar question about the group with 8 elements determines the group with 4 members.

4. Similar question about the group with 4 elements determines the group with 2 members.

5. Question if the winner is one of two determines the winner.

The last example is slightly artificial one. A person which is willing to answer five questions of the type *yes*" or "*no*" will probably agree to give the direct answer to the question " *Who won the literature contest?*".

Example 2.2. Suppose we have 22 electric bulbs connected into one series circuit. If one of the bulbs blew out, the other bulbs would not be able to shine because electric current would have been interrupted. We have an ohmmeter at our disposal and we can measure the resistance between two arbitrary points of the circuit. What is the minimum number of measurements for determining the blown bulb?

The basic experiment has the entropy $\log_2(22) = 4.46$ bits. A single measurement by ohmmeter says us whether there is or not a disruption between measured points of the circuit so such measuring gives us 1 bit of information. Therefore we need at least 5 measurements for determining the blown bulb.

Assign numbers 1 to 22 to bulbs in the order in which they are connected in the circuit.

First connect the ohmmeter before the first bulb and behind the eleventh one. If the measured resistance is infinite, the blown bulb is among bulb 1 to 11 otherwise the blow bulb is among bulbs 12 to 22.

Now partition the disrupted segment into two subsegments with (if possible) equal number of bulbs and determine by measuring the bad segment etc. After the first measurement there are 11 suspicious bulbs, after the second measurement the set with blown bulb contains 4 or 5 bulbs, the third measurement determines 2 or 3 bulbs, the forth measurement determines the single blown bulb or 2 bad bulbs and finally the fifth measurement (if needed) determines the blown bulb.

Example 2.3. Suppose you have 27 coins. One of the coins is forged. You only know that the forged coin is slightly lighter than the other 26 ones. We have a balance scale as a measuring device. Your task is to determine the forged coin using as little weighing as possible. The basic experiment has 27 outcomes and its entropy is $\log_2(27) = 4.755$ bits.

If we place different number of coins on both sides of the balance surely, the side with greater number of coins will be heavier and such experiment gives us no information.

Place any number of coins on the left side of the balance and the same number of coins on the right side of the balance. Denote by L, R, A the sets of coins on the left side of the balance, on the right side of the balance, and aside the balance. There are three outcomes of such weighing.

- The left side of the balance is lighter. The forged coin is in the set L.
- The right side of the balance is lighter. The forged coin is in the set R.
- Both sides of the balance are equal. The forged coin is in the set A.

The execution of experiment where all coins are partitioned into three subsets L, R and A (where $|L| = |R|$) gives us the answer to the question which of them contains the forged coin. In order to obtain maximum information from this experiment the sets L, R and A should have equal (or as equal as possible) probabilities. In our case of 27 coins we can easy achieve this since 27 is divisible by 3. In such a case it is possible to get $\log_2(3) = 1.585$ bits of information from one weighing.

Since $\log_2(27)/\log_2(3) = \log_2(3^3)/\log_2(3) = 3\log_2(3)/\log_2(3) = 3$, at least three weighing will be necessary for determining the forged coin. The actual problem solving follows:

1. weighing: Partition 27 coins into subsets L, R, A with $|L| = |R| = |A| = 9$ (all subsets contain 9 coins). Determine (and denote by F) the subset containing the forged coin.

2. weighing: Partition 9 coins of the set F into subsets L_1, R_1, A_1 with $|L_1| = |R_1| = |A_1| = 3$ (all subsets contain 3 coins). Determine (and denote by F_1) the subset containing the forged coin.

3. weighing: Partition 3 coins of the set F_1 into subsets L_2, R_2, A_2 with $|L_1| = |R_1| = |A_1| = 1$ (all subsets contain only 1 coin). Determine the forged coin.

In general case where n is not divisible by 3 then $n = 3m+1 = m+m+(m+1)$ – in this case $|L| = |R| = m$ and $|A| = m+1$, or $n = 3m+2 = (m+1)+(m+1)+m$ – in this case $|L| = |R| = m+1$ and $|A| = m$.

Example 2.4. Suppose we have 27 coins. One of the coins is forged. We only know that the forged coin is slightly lighter, or slightly heavier than the other 26 ones.

We are to determine the forged coin and to find out whether it is heavier or lighter. The basic experiment has now $2 \times 27 = 54$ possible outcomes – every one from 27 coins can be forged whereas it can be lighter or heavier than the genuine one. The basic experiment has $2 \times 27 = 54$ outcomes and its entropy is $\log_2(54) = 5.755$ bits. The entropy of one weighing is less or equal to $\log_2(3) = 1.585$ bits from what it follows that three weighings cannot do for determining the forged coin.

One possible solution of this problem: Partition 27 coins into subsets L, R, A with $|L| = |R| = |A| = 9$ (all subsets contain 9 coins). Denote by $w(X)$ the weight of the subset X.

a) If $w(L) = w(R)$ we know that the forged coin is in the set A. The second weighing says us that $w(L) < w(A)$ – the forged coin is heavier, or $w(L) > w(A)$ – the forged coin is lighter. The third weighing determines which triplet – subset of A contains the forged coin. Finally, by the fourth weighing we determine the single forged coin.

b) If $w(L) < w(R)$ we know that A contains only genuine coins. The second weighing says that $w(L) < w(A)$ – the forged coin is lighter and is contained in the set L, or $w(L) > w(A)$ – the forged coin is heavier and is contained in the set L, or $w(L) = w(A)$ – the forged coin is heavier and is contained in the set R. The third weighing determines the triplet of coins with the forged coin and the forth weighing determines the single forged coin.

c) If $w(L) > w(R)$ the procedure is analogous as in the case b).

Example 2.5. We are given n large bins containing iron balls. All balls in one bin have the same known weight w gram. $n-1$ bins contain identical balls but one bin contains balls 1 gram heavier. Our task is to determine the bin with heavier balls. All balls are apparently the same and hence the heavier ball can be identified only by weighing.

We have at hand a precision electronic commercial scale which can weigh arbitrary number of balls with an accuracy better than 1 gram. How many weighings is necessary for determining the bin with heavier balls? The basic experiment has n possible outcomes – its entropy is $\log_2(n)$ bits.

We try to design our measurement in order to obtain as much information as possible. We put on the scale 1 ball from the first bin, 2 balls from the second bin, etc. n balls from the n-th bin. The total number of all balls on the scale is $1 + 2 + \cdots + n = \frac{1}{2}n(n+1)$ and the total weight of all balls on the scale is $\frac{1}{2}n(n+1)w + k$ where k is the serial number of the bin with heavier balls. Hence it is possible to identify the bin with heavier balls using only one weighing.

Example 2.6. Telephone line from the place P to the place Q is 100 m long. The line was disrupted somewhere between P and Q. We can measure the line in such a way that we attach a measuring device to an arbitrary point X of the segment PQ and the device says us whether the disruption is between points P and X or not. We are to design a procedure which identifies the segment of the telephone line not longer than 1 m containing the disruption.

Denote by Y the distance of the point of disruption X from the point P. Then Y is a continuous random variable, $Y \in \langle 0, 100\rangle$ with an uniform distribution on this interval. We have not defined the entropy of an experiment with infinite number of events, but fortunately our problem is not to determine the exact value of Y, but only the interval of the length 1 m containing X. The basic experiment is

$$\mathbf{B} = \big\{\langle 0,1),\langle 1,2),\ldots,\langle 98,99),\langle 99,100\rangle\big\}$$

with the entropy $H(\mathbf{B}) = \log_2(100) = 6.644$ bits.

Having determined the interval $\langle a, b\rangle$ containing the disruption, our measurement allows to specify for every $c \in \langle a, b\rangle$ whether the disruption occurs in interval $\langle a, c)$ or $\langle c, b\rangle$. Provided that the probability of disruption in a segment $\langle a, b\rangle$ is directly proportional to its length, it is necessary to chose the a c in the middle of segment $\langle a, b\rangle$ in order to obtain maximum information from such measurement – 1 bit. Since the basic experiment **B** has entropy 6.644 bits we need at least 7 measurements. The procedure of determining the segment containing the disruption will be as follows: The first measurement says us whether the defect occurred in the first, or in the second half of telephone line, second measurement specifies the segment $100/2^2$ m long containing disruption, etc., the sixth measurement gives us the erroneous segment $\langle a, b\rangle$ $100/2^6 = 100/64 = 1.5625$ m long. This segment contains exactly one integer point c which will be taken as dividing point for the last measurement.

Till now we were studying such organizations of experiments which allow us to **certainly determine** which event of the basic experiment occurred using the minimum number of available experiments. If possible we execute the basic experiment (see iron balls in bins). However, in most cases the difficulty of such problems rests upon the fact that we are limited only to experiments of a certain type. The lower bound of the number of available experiments is directly proportional to the entropy of the basic experiment and indirectly proportional to the entropy of available experiment.

We have the greatest uncertainty before executing an experiment in the case that all its events have the same probability – in this case the entropy of the experiment is $H(1/n, 1/n, \ldots, 1/n) = \log_2(n)$. This is the worst case of uncertainty and that is why we suppose that all events of basic experiment have the same probability in cases that these probabilities are not given. This assumption leads to such a procedure which does not prefer any event of the basic experiment.

How should be modified our procedure in the case of basic experiment with different event probabilities? If the goal "to certainly determine the occurred event using minimum number of available experiments" remains, then nothing needs to be changed.

But we could formulate another goal: "To find such procedure of determining the occurred event that minimizes the mean number of experiments". We abandoned the requirement "**to determine certainly**". We admit the possibility that in several adverse but not very likely situations the proposed procedure will require many executions of available experiments. But our objective is to minimize the mean number of questions if our procedure is repeated many times.

Example 2.7. On the start line of F1 there were 18 cars. Cars a_1 and a_2 are from technologically advanced team and that is why both have the probability of victory equal to 1/4. Every from remaining 16 cars $a_3, \dots, a_{18}$ wins with probability 1/32. The basic experiment is

$$\mathbf{A} = \{\{a_1\}, \{a_2\}, \{a_2\}, \dots, \{a_{18}\}\},$$

and its entropy is

$$H(\mathbf{A}) = H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{32}, \frac{1}{32}, \dots, \frac{1}{32}\right) = 3.5\ .$$

Therefore the mean number of questions with only two possible answers for determining the winner cannot be less than 3.5. For obtaining maximum information, it is necessary to formulate the question in such a way that both answers "*yes*" and "*no*" have the same probability 1/2.

One of possible ways is to make a decision between the sets $A_1 = \{a_1, a_2\}$ and $A_2 = \{a_3, a_4, \dots, a_{18}\}$ after the first question. In half of cases the winner is in A_1, and only one question suffices for determining the winner. In another half of cases we get the set A_2 with 16 equivalent events and here further 4 questions are necessary for determining the winner. The mean number of questions is $\frac{1}{2}.2 + \frac{1}{2}.5 = 3.5$.

For comparison we present the procedure of the solving an analogical problem when no probabilities are given. This procedure needs at least 4 and in several cases 5 questions.

Assign numbers from 1 to 18 to all cars.

1. Is the number of the winner among numbers $1 - 9$? The answer determines the set B_1 with 9 elements containing the winner.
2. Is the number of the winner among four least numbers of B_1? The result is the set B_2 containing the winner, $|B_2| = 4$ or $|B_2| = 5$.
3. Is the number of the winner among two least numbers of B_2? The result is the set B_3 containing the winner, $|B_3| = 2$ or $|B_2| = 3$.
4. Is the number of the winner the least number in B_3? If yes, STOP, we have the winner. Otherwise we have the setB_4 with only two elements.
5. We determine the winner by direct question.

The notion of entropy is very successfully used by modelling mobility of passengers in a studied region. Suppose that there are n bus stops in the given region and we want to determine for every ordered pair (i, j) of bus stops the number Q_{ij} of passengers travelling from the bus stop i to the bus stop j.

The values Q_{ij} can be determined by complex traffic measuring but such research is very expensive. It is much easier to determine for every bus stop i the number P_i of passengers departing from i and the number R_i of passengers arriving into i.

Obviously $\sum_{i=1}^n P_i = \sum_{j=1}^n R_j = Q$, where Q is the total number of passengers during investigated period. The following equations hold for unknown values Qij:

$$\sum_{i=1}^{n} Q_{ij} = R_j \qquad \text{for } j = 1, 2, \ldots, n \tag{2.24}$$

$$\sum_{j=1}^{n} Q_{ij} = P_i \qquad \text{for } i = 1, 2, \ldots, n \tag{2.25}$$

$$Q_{ij} \geq 0 \qquad \text{for } i,\ j = 1, 2, \ldots, n \tag{2.26}$$

Denote by c_{ij} the transport expenses of transportation of one passenger from the place i to the place j. (These expenses contain fares, but they can include time loss of passengers, travel discomfort, etc.) One of hypotheses says that the total transport expenses

$$C = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} Q_{ij} \tag{2.27}$$

are minimal in steady state of transportation system.

Provided that this hypothesis is correct, the values Q_{ij} can be obtained by solving the following problem: Minimize (2.27) subject to (2.24), (2.25) and (2.26), what is nothing else as the notorious transportation problem. Unfortunately, the results of just described model differ considerably from real observations.

It shows that in the frame of the same societal and economical situation there is equal measure of freedom of destination selection, which can be expressed by the entropy

$$H\left(\frac{Q_{11}}{Q}, \ldots \frac{Q_{1n}}{Q}, \frac{Q_{21}}{Q}, \ldots \frac{Q_{2n}}{Q}, \ldots\ldots, \frac{Q_{n1}}{Q}, \ldots \frac{Q_{nn}}{Q}\right). \tag{2.28}$$

The ratio $\dfrac{Q_{ij}}{Q}$ in (2.28) expresses the probability that a passenger travels from the bus stop i to the bus stop j.

Entropic models are based on maximization of (2.28), or on combination objective functions (2.27) and (2.28), or on extending the constrains by $C \le C_0$ or $H \ge H_0$. Such models correspond better to practical experiences.

2.6 Conditional entropy

Let $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ be an experiment on a probability space $(\Omega, \mathcal{A}, P)$. Suppose that an elementary event $\omega \in \Omega$ occurred. It suffices for our purposes to know which event of the experiment $\mathbf{B}$ occurred, i. e., for which B_j $(j = 1, 2, \dots m)$ it holds $\omega \in B_j$. Because of some limitations we cannot execute the experiment $\mathbf{B}$ (neither we can learn which $\omega \in \Omega$ occurred) but we know the result A_i of the experiment $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$.

Suppose that the event A_i occurred. Then the probabilities of events B_1, B_2, $\dots$, B_m given A_i has occurred ought to be $P(B_1|A_i)$, $P(B_2|A_i)$, $\dots$, $P(B_m|A_i)$. Our uncertainty before performing the experiment $\mathbf{B}$ was

$$H(\mathbf{B}) = H(P(B_1), P(B_2), \dots, P(B_m)) \ .$$

After receiving the report that the event A_i occurred the uncertainty about the result of the experiment $\mathbf{B}$ changes to

$$H\big(P(B_1|A_i), P(B_2|A_i), \dots, P(B_m|A_i)\big),$$

which we will denote by $H(\mathbf{B}|A_i)$.

Definition 2.3. Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ are two experiments. **The conditional entropy of the experiment B given the event A_i occurred** (or shortly only given the event A_i) is

$$H(\mathbf{B}|A_i) = H\big(P(B_1|A_i), P(B_2|A_i), \dots, P(B_m|A_i)\big) = \\ = -\sum_{j=1}^{m} P(B_j|A_i). \log_2(P(B_j|A_i)). \quad (2.29)$$

Example 2.8. Die rolling. Denote $\mathbf{B} = \{B_1, B_2, \dots, B_6\}$ the experiment in which the event B_i means *"i spots appeared on the top face of die"* for $i = 1, 2, \dots 6$. The probability of all events B_i is the same – $P(B_i) = 1/6$. Our uncertainty about the result of the experiment $\mathbf{B}$ is

$$H(\mathbf{B}) = H(1/6, 1/6, \dots, 1/6) = \log_2(6) = 2.585 \text{ bits.}$$

Suppose that we have received the report *"The result is an odd number"* after execution of the experiment $\mathbf{B}$. Denote $A_1 = B_1 \cup B_3 \cup B_5$, $A_2 = B_2 \cup B_4 \cup B_6$. The event A_1 means *"The result is an odd number"* the event A_2 means *"The result is an even number"*. Both events carry the same information $-\log_2(1/2) = 1$ bits.
After receiving the message A_1 our uncertainty about the result of the experiment $\mathbf{B}$ changes from $H(\mathbf{B})$ to

$$\begin{aligned} &H(\mathbf{B}|A_1) = \\ &\quad H\big(P(B_1|A_1), P(B_2|A_1), P(B_3|A_1), P(B_4|A_1), P(B_5|A_1), P(B_6|A_1)\big) = \\ &\qquad H(1/3, 0, 1/3, 0, 1/3, 0) = H(1/3, 1/3, 1/3) = \log_2(3) = 1.585 \text{ bits.} \end{aligned}$$

The message *"The result is an odd number"* – i. e., the event A_1 with 1 bit of information has lowered our uncertainty from $H(\mathbf{B}) = 2.585$ to $H(\mathbf{B}|A_1) = 1.585$ – exactly by the amount of its information.

WARNING! This is not a generally valid fact!

The following example shows that in some cases the report *"The event A_i occured"* can even increase the conditional entropy $H(\mathbf{B}|A_i)$.

Example 2.9. Michael Schumacher was a phenomenal pilot of Formula One. He holds seven world championship titles in years 1994, 1995 a 2000–2004. In 2004 he won 13 races out of 18. Hence his chance to win the race was almost 3/4. The following example was inspired by just mentioned facts.

On the start line there are 17 pilots – Schumacher with probability of victory 3/4 and the rest 16 equal pilots every one of them with chance 1/64.

Denote $\mathbf{B} = \{B_1, B_2, \dots, B_{17}\}$ the experiment in which the event B_1 is the event *"Schumacher has won"* and B_i for $i = 2, 3, \dots, 17$ means *"Pilot i has won"*. Let $P(B_1) = 3/4$, $P(B_2) = P(B_3) = \dots = P(B_{17}) = 1/64$. Entropy of the experiment $\mathbf{B}$ is

$$H(\mathbf{B}) = H\,(3/4, 1/64, 1/64, \dots, 1/64) = 1.811.$$

The message B_1 ”*Schumacher has won*” contains $-\log_2 P(B_1) = -\log_2(0.75)$ 0.415 bits of information while the message ”*Pilot* 17 *has won*” carries $-\log_2(P(B_{17})) = -\log_2(1/64) = 6$ bits of information.

Let $\mathbf{A} = \{A_1, A_2\}$ be the experiment where A_1 is the event ”*Schumacher has won*” (i. e., $A_1 = B_1$) and A_2 is the event ”*Schumacher has not won*” (i. e., $A_2 = B_2 \cup B_3 \cup \dots \cup B_{17}$. It holds $P(A_1) = 3/4$, $P(A_2) = 1/4$. Suppose that we get the message that Schumacher has not won – the event A_2 occurred. This message carries with it $-\log_2(P(A_2)) = -\log_2(1/4) = 2$ bits of information. Our uncertainty changes after this message from $H(\mathbf{B}) = 1.811$ to $H(\mathbf{B}|A_2)$. Calculate

$$\begin{aligned} H(\mathbf{B}|A_2) = H\big(P(B_1|A_2), P(B_2|A_2), \dots, P(B_{17}|A_2)\big) = \\ = H(0, 1/16, 1/16, \dots, 1/16) = H(1/16, 1/16, \dots, 1/16) = 4. \end{aligned}$$

The message ”*The event A_2 occurred*” (i. e., ”*Schumacher has not won*”) brought 2 bits of information and in spite of this our uncertainty about the result of the race has risen from $H(\mathbf{B}) = 1.811$ to $H(\mathbf{B}|A_2) = 4$.

If the event A_1 is the result of the experiment $\mathbf{A}$, then $P(B_1|A_1) = P(B_1|B_1) = 1$ and $P(B_j|A_1) = 0$ for $j = 2, 3, \dots, 17$ and therefore $H(\mathbf{B}|A_1) = H(1, 0, \dots, 0) = 0$. The probability of the result A_1 is 3/4. The result A_2 of $\mathbf{A}$ occurred with the probability 1/4. The mean value of conditional entropy of $\mathbf{B}$ after executing the experiment $\mathbf{A}$ is

$$P(A_1).H(\mathbf{B}|A_1) + P(A_2).H(\mathbf{B}|A_2) = (3/4).0 + (1/4).4 = 1 \text{ bit}.$$

Let us make a short summary of this section. We are interested in the result of the experiment $\mathbf{B}$ with the entropy $H(\mathbf{B})$. Suppose that an elementary event $\omega \in \Omega$ occurred. We have received the report that $\omega \in A_i$ and this report has chained the entropy of the experiment $\mathbf{B}$ from $H(\mathbf{B})$ to $H(\mathbf{B}|A_i)$. For every $\omega \in \Omega$ there exists exactly one set $A_i \in \mathbf{A}$ such that $\omega \in A_i$. Hence we can uniquely assign the number $H(\mathbf{B}|A_i)$ to every $\omega \in \Omega$. This assignment is a discrete random variable[3] on the probability space$(\Omega, \mathcal{A}, P)$ with mean value $\sum_{i=1}^{n} P(A_i).H(\mathbf{B}|A_i)$.

[3] The exact definition of this random variable is:

$$h(\mathbf{B}|\mathbf{A})(\omega) = \sum_{i=1}^{n} H(\mathbf{B}|A_i).\chi_{A_i}(\omega),$$

where $\chi_{A_i}(\omega)$ is the indicator of the set A_i, i. e., $\chi_{A_i}(\omega) = 1$ if and only if $\omega \in A_i$, otherwise $\chi_{A_i}(\omega) = 0$.

Definition 2.4. Let $A = \{A_1, A_2, \dots, A_n\}$, $B = \{B_1, B_2, \dots, B_m\}$ are two experiments. **The conditional entropy of experiment B given experiment A** is

$$H(\mathbf{B}|\mathbf{A}) = \sum_{i=1}^{n} P(A_i).H(\mathbf{B}|A_i). \tag{2.30}$$

It holds:

$$\begin{aligned}
\sum_{i=1}^{n} P(A_i).H(\mathbf{B}|A_i) &= \sum_{i=1}^{n} P(A_i).H\big(P(B_1|A_i), P(B_2|A_i), \dots, P(B_m|A_i)\big) = \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i).P(B_j|A_i).\log_2(P(B_j|A_i)) = \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i).\frac{P(A_i \cap B_j)}{P(A_i)}.\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right) = \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right).
\end{aligned}$$

Hence we can write:

$$H(\mathbf{B}|\mathbf{A}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right) \tag{2.31}$$

Definition 2.5. Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ are the experiments on a probability space $(\Omega, \mathcal{A}, P)$. Then the **joint experiment of experiments A, B** is the experiment

$$\mathbf{A} \wedge \mathbf{B} = \big\{A_i \cap B_j \mid A_i \in \mathbf{A},\ B_j \in \mathbf{B}\big\}. \tag{2.32}$$

After executing the experiment $\mathbf{A}$ and afterwards the experiment $\mathbf{B}$, we obtain the same total amount of information as by the executing the joint experiment $\mathbf{A} \wedge \mathbf{B}$.
Execute the experiment $\mathbf{A}$ – the mean value of obtained information from this experiment is $H(\mathbf{A})$. The remaining entropy of experiment $\mathbf{B}$ after executing experiment $\mathbf{A}$ is $H(\mathbf{B}|\mathbf{A})$ so it should hold that $H(\mathbf{A} \wedge \mathbf{B}) = H(\mathbf{A}) + H(\mathbf{B}|\mathbf{A})$. The following reasoning gives the exact proof of the last statement.

By theorem 2.6 (page 28) the equation (2.12) holds. Let $\mathbf{A} \wedge \mathbf{B}$ be the joint experiment of experiments $\mathbf{A}$, $\mathbf{B}$. Denote $q_{ij} = P(A_i \cap B_j)$, $p_i = P(A_i)$. Then

$$p_i = P(A_i) = \sum_{j=1}^{m} P(A_i \cap B_j) = \sum_{j=1}^{m} q_{ij}.$$

Assumptions of theorem 2.6 are fulfilled and that is why

$$\begin{aligned}
H(\mathbf{A} \wedge \mathbf{B}) &= H\left(\underbrace{q_{11}, q_{12}, \dots q_{1m}}_{p_1}, \underbrace{q_{21}, q_{22}, \dots q_{2m}}_{p_2}, \dots, \underbrace{q_{n1}, q_{n2}, \dots q_{nm}}_{p_n}\right) = \\
&= H(p_1, p_2, \dots, p_n) + \sum_{j=1}^{m} p_i . H\left(\frac{q_{i1}}{p_i}, \frac{q_{i2}}{p_i}, \dots, \frac{q_{im}}{p_i}\right) = \\
&= H\big(P(A_1), P(A_2), \dots, P(A_n)\big) + \\
&+ \sum_{i=1}^{m} P(A_i) H\left(\frac{P(A_i \cap B_1)}{P(A_i)}, \frac{P(A_i \cap B_2)}{P(A_i)}, \dots, \frac{P(A_i \cap B_m)}{P(A_i)}\right) = \\
&= H(\mathbf{A}) + H(\mathbf{B}|\mathbf{A})
\end{aligned}$$

Hence the following theorem hods:

Theorem 2.12. *Let* $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ *are two experiments on a probability space* $(\Omega, \mathcal{A}, P)$. *Then*

$$H(\mathbf{A} \wedge \mathbf{B}) = H(\mathbf{A}) + H(\mathbf{B}|\mathbf{A}) \tag{2.33}$$

Equation (2.33) says that $H(\mathbf{B}|\mathbf{A})$ is the remaining entropy of joint experiment $\mathbf{A} \wedge \mathbf{B}$ after executing the experiment $\mathbf{A}$.

Definition 2.6. Let $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ are experiments on a probability space $(\Omega, \mathcal{A}, P)$. We say that the experiments $\mathbf{A}$, $\mathbf{B}$ are **statistically independent** (or only **independent**) if for every $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$ the events A_i, B_j are independent.

2.7 Mutual information of two experiments

Return again to the situation where we are interested in the result of the experiment **B** with the entropy $H(\mathbf{B})$. We are not able to execute this experiment from some reasons but we can execute another experiment **A**. After executing the experiment **A**, entropy of experiment **B** changes from $H(\mathbf{B})$ to $H(\mathbf{B}|\mathbf{A})$ – this is the mean value of additional information obtainable from the experiment **B** after executing the experiment **A**. The difference $H(\mathbf{B})-H(\mathbf{B}|\mathbf{A})$ can be considered to be the mean value of information about the experiment **B** contained in the experiment **A**.

Definition 2.7. The mean value of information $I(\mathbf{A},\mathbf{B})$ about the experiment B in the experiment A is

$$I(\mathbf{A},\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}). \tag{2.34}$$

Theorem 2.13.

$$I(\mathbf{A},\mathbf{B}) = H(\mathbf{A}) + H(\mathbf{B}) - H(\mathbf{A}\wedge\mathbf{B}) \tag{2.35}$$

Proof. From (2.33) it follows: $H(\mathbf{B}|\mathbf{A}) = H(\mathbf{A}\wedge\mathbf{B}) - H(\mathbf{A})$. Substitute $H(\mathbf{A}\wedge\mathbf{B}) - H(\mathbf{A})$ for $H(\mathbf{B}|\mathbf{A})$ in (2.34) and obtain the required formula (2.35). ■

We can see from formula (2.35) that $I(\mathbf{A},\mathbf{B}) = I(\mathbf{B},\mathbf{A}) - I(\mathbf{A},\mathbf{B})$ is a symmetrical function. Hence the mean value of information about the experiment **B** in the experiment **A** equals to the mean value of information about the experiment **A** in the experiment **B**. That is why the value $I(\mathbf{A},\mathbf{B})$ is called **mutual information of experiments A, B**.

Theorem 2.14. *Let* $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ *are two experiments on a probability space* $(\Omega, \mathcal{A}, P)$. *Then*

$$I(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j).\log_2\left(\frac{P(A_i\cap B_j)}{P(A_i).P(B_j)}\right) \ . \tag{2.36}$$

Proof. $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ is a partition of the space Ω, therefore

$$B_j = B_j\cap\Omega = B_j\cap\bigcup_{i=1}^{n} A_i = \bigcup_{i=1}^{n} A_i\cap B_j \ .$$

Since union on the left hand side of the last expression is union of disjoint sets it holds:

$$P(B_j) = \sum_{i=1}^{n} P(A_i \cap B_j) \ .$$

Substituting for $H(\mathbf{B}|\mathbf{A})$ from equation (2.31) into (2.34) we get:

$$\begin{aligned}
&I(\mathbf{A},\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}) = \\
&= -\sum_{j=1}^{m} P(B_j).\log_2 P(B_j) + \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right) = \\
&= -\sum_{j=1}^{m}\sum_{i=1}^{n} P(A_i \cap B_j).\log_2 P(B_j) + \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right) = \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\left[\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right) - \log_2 P(B_j)\right] = \\
&\qquad\qquad = \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i).P(B_j)}\right)
\end{aligned}$$

■

Theorem 2.15. *Let* $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ *are two experiments on a probability space* $(\Omega, \mathcal{A}, P)$. *Then*

$$0 \le I(\mathbf{A},\mathbf{B}), \tag{2.37}$$

with equality if and only if $\mathbf{A}$, $\mathbf{B}$ *are statistically independent.*

Proof. We will make use of formula (2.36) from the theorem 2.14 and inequality $\ln x \le x - 1$ which is valid for all real $x > 0$ with equality if and only if $x = 1$.

$$\begin{aligned}
&P(A_i \cap B_j).\log_2\left(\frac{P(A_i).P(B_j)}{P(A_i \cap B_j)}\right) = P(A_i \cap B_j).\ln(2).\ln\left(\frac{P(A_i).P(B_j)}{P(A_i \cap B_j)}\right) \le \\
&\le P(A_i \cap B_j).\ln(2).\left[\left(\frac{P(A_i).P(B_j)}{P(A_i \cap B_j)}\right) - 1\right] = \ln(2).\left[P(A_i).P(B_j) - P(A_i \cap B_j)\right],
\end{aligned}$$

with equality if and only if $\dfrac{P(A_i).P(B_j)}{P(A_i \cap B_j)} = 1$, i. e., if and only if A_i, B_j are independent events.

From the last formula we have:

$$\begin{aligned}
-I(\mathbf{A},\mathbf{B}) &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j).\log_2\left(\frac{P(A_i).P(B_j)}{P(A_i\cap B_j)}\right) \le \\
&\le \ln(2).\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\big(P(A_i).P(B_j) - P(A_i\cap B_j)\big)\right] = \\
&= \ln(2).\left[\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i).P(B_j) - \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j)}_{=1}\right] = \\
&= \ln(2).\left[\sum_{i=1}^{n} P(A_i)\underbrace{\sum_{j=1}^{m} P(B_j)}_{=1} -1\right] = \ln(2).\left[\underbrace{\sum_{i=1}^{n} P(A_i)}_{=1} -1\right] = 0,
\end{aligned}$$

with equality if and only if all pairs of events A_i, B_j for $i = 1, 2, \dots, n$ and for $j = 1, 2, \dots, m$ are independent . ■

Theorem 2.16.

$$H(\mathbf{B}|\mathbf{A}) \le H(\mathbf{B}), \tag{2.38}$$

with equality if and only if **A**, **B** *are statistically independent.*

Proof. The statement of the theorem follows immediately from the inequality $0 \le I(\mathbf{A},\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A})$. ■

Theorem 2.17.

$$H(\mathbf{A}\wedge\mathbf{B}) \le H(\mathbf{A}) + H(\mathbf{B}), \tag{2.39}$$

with equality if and only if **A**, **B** *are statistically independent.*

Proof. It follows from theorem 2.13, formula (2.35), and from 2.15:

$$0 \le I(\mathbf{A},\mathbf{B}) = H(\mathbf{A}) + H(\mathbf{B}) - H(\mathbf{A}\wedge\mathbf{B}),$$

with equality if and only if **A**, **B** are statistically independent. ■

2.7.1 Summary

The conditional entropy of the experiment $\mathbf{B}$ given the event A_i is

$$H(\mathbf{B}|A_i) = H\big(P(B_1|A_i), \dots, P(B_m|A_i)\big) = -\sum_{j=1}^{m} P(B_j|A_i).\log_2(P(B_j|A_i)).$$

The conditional entropy of experiment $\mathbf{B}$ given experiment $\mathbf{A}$ is

$$H(\mathbf{B}|\mathbf{A}) = \sum_{i=1}^{n} P(A_i).H(\mathbf{B}|A_i).$$

It holds

$$H(\mathbf{B}|\mathbf{A}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i)}\right).$$

The joint experiment of experiments $\mathbf{A}$, $\mathbf{B}$ is the experiment

$$\mathbf{A} \wedge \mathbf{B} = \big\{A_i \cap B_j \mid A_i \in \mathbf{A},\ B_j \in \mathbf{B}\big\}.$$

It holds:

$$H(\mathbf{A} \wedge \mathbf{B}) = H(\mathbf{A}) + H(\mathbf{B}|\mathbf{A}).$$

The mutual information of experiments $\mathbf{A}$, $\mathbf{B}$ is

$$I(\mathbf{A},\mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B}|\mathbf{A}).$$

It holds:

$$\begin{aligned} I(\mathbf{A},\mathbf{B}) &= H(\mathbf{A}) + H(\mathbf{B}) - H(\mathbf{A} \wedge \mathbf{B}) \\ I(\mathbf{A},\mathbf{B}) &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i \cap B_j).\log_2\left(\frac{P(A_i \cap B_j)}{P(A_i).P(B_j)}\right). \end{aligned}$$

The following relations hold:

$$0 \le I(\mathbf{A},\mathbf{B}), \qquad H(\mathbf{B}|\mathbf{A}) \le H(\mathbf{B}), \qquad H(\mathbf{A} \wedge \mathbf{B}) \le H(\mathbf{A}) + H(\mathbf{A})$$

with equalities if and only if $\mathbf{A}$ and $\mathbf{B}$ are statistically independent.

Chapter 3

Sources of information

3.1 Real sources of information

Any object (person, device, equipment) that generates successive messages on its output can be considered a **source of information** Thus a man using a lamp to flash out characters of Morse code, a keyboard transmitting 8-bit words, a telephone set generating analog signal with frequency from 300 to 3400 Hz, a primary signal from audio CD-reader outputting 44100 16-bit audio samples per second, a television camera producing 25 frames per second, etc.

We can see that the television signal is much more complicated than the telephone one. But everyone will agree that 10 minutes of watching TV test pattern (transmitted by the complicated signal) gives less information than 10 minutes of telephone call.

Sources of information can produce the signal in discrete time intervals, or continuously in time. The sources that produces messages in discrete time intervals from an enumerable set of possibilities are called discrete. The sources which are not discrete are called continuous (e. g., speech and music sources). Every continuous signal can be measured in sufficiently small time intervals and replaced by the corresponding sequence of measured values with an arbitrary good accuracy. Such a procedure is called **sampling**. Thus every continuous source can be approximated by a discrete source. It shows that digital signals can be transmitted and stored with extraordinary quality, more effectively, and reliably than analog ones. Moreover, digital processing of sound and picture offers incredible tools. That is why there are plans to replace all analog

TV broadcasting by a digital system. Therefore we will study only discrete information sources with finite alphabet.

We will assume that in discrete time moments $t = t_1, t_2, t_3, \dots$ the source produces messages $X_{t_1}, X_{t_2}, X_{t_3}, \dots$ which are discrete random variables taking only finite number of values. The finite set of possible messages produced by the source is called **source alphabet**, the elements of source alphabet are called **characters** or **source characters**.

Time intervals between time moments $t = t_1, t_2, t_3, \dots$ may be regular or irregular. For example the source transmitting Morse code uses symbols "." "—" and "/" (pause). The time intervals between two successive symbols are not equal since "." is shorter than "—".

However, it is advantageous to suppose that all time intervals between successive characters are the same and equal to 1 time unit. Then we will work with the sequence of discrete random variables $X_1, X_2, X_3 \dots$.

Definition 3.1. The **discrete random process** is a sequence of random variables $\mathcal{X} = X_1, X_2, X_3 \dots$. If X_i takes the value a_i for $i = 1, 2, \dots$, the sequence $a_1, a_2, \dots$ is called **realization of random process** $\mathcal{X}$.

In this chapter we will study the information productivity of various sources of information. Discrete sources of information differ one from another by transmitting frequency, by cardinalities of source alphabets, and by probability distributions of random variables X_i. The dependency of information productivity on the source frequency is simple (is directly proportional to the frequency). Therefore we will characterise the information sources by the amount of information per one transmitted character. We will see that information productivity of an information source depends not only on cardinality of source alphabet, but also on probability distribution of random variables X_i.

3.2 Mathematical model of information source

Definition 3.2. Let X be a finite nonempty set, let X^* be the set of all finite sequences of elements from X including an empty sequence denoted by e. The set X is called **alphabet**, the elements of X are **characters of** X, the elements of the set X^* are called **words**, e **empty word**. Denote by X^n the set of all ordered n-tuples of characters from X (finite sequences of n characters from X). Every element $\mathbf{x}$ of X^n is called **word of the length** n, the number n is called **length** of the word $\mathbf{x} \in X^n$.

Let $P : X^* \to \mathbb{R}$ is a real nonnegative function defined on X^* with the following properties:

1. $P(e) = 1 \qquad (3.1)$
2. $$\sum_{(x_1,\dots,x_n)\in X^n} P(x_1,\dots,x_n) = 1 \qquad (3.2)$$
3. $$\sum_{(y_{n+1},\dots,y_{n+m})\in X^m} P(x_1,\dots,x_n,y_{n+1},\dots,y_{n+m}) = P(x_1,\dots,x_n) \qquad (3.3)$$

Then the ordered couple $\mathcal{Z} = (X^*, P)$ is called **source of information** or shortly **source**. The number $P(x_1, x_2, \dots, x_n)$ is called **probability of the word** $x_1, \dots, x_n$.

The number $P(x_1, x_2, \dots, x_n)$ expresses the probability of the event that the source from its start up generates the character x_1 in time moment 1, the character x_2 in time moment 2 etc., and the character x_n in time moment n. In other words, $P(x_1, x_2, \dots, x_n)$ is the probability of transmitting the word $x_1, x_2, \dots, x_n$ in n time moments starting with the moment of source start up.

The condition (3.1) says that the source generates the empty word in 0 time moments with probability 1. The condition (3.2) says that in n time moments the source surely generates some word of the length n. The third condition (3.3), called also the condition of consistency, expresses the requirement that the probability of all words of the length $n + m$ with prefix $x_1, x_2, \dots, x_n$ is equal to the probability $P(x_1, x_2, \dots, x_n)$ of the word $x_1, x_2, \dots, x_n$ since

$$\{y_1, y_2, \dots, y_{n+m} \mid y_1 = x_1, y_2 = x_2, \dots, y_n = x_n\} =$$
$$= \bigcup_{z_1,z_2\dots,z_m\in X^m} \{x_1, x_2, \dots, x_n, z_1, z_2, \dots, z_m\} \ .$$

It is necessary to note, in this place, two differences between the linguistic and our notion of the term *word*. The word in linguistics is understood to be such a sequence of characters which is an element of the set of words – vocabulary of the given language. In informatics the word is an arbitrary finite sequence of characters. The word "*weekend*" is an English word since it can be found in the English vocabulary but the word "*kweeedn*" is not, while both mentioned character sequences are words by definition 3.2.

The second difference is that in natural language the words are separated by space character "␣" unlike to our definition 3.2 by which the sequence $x_1, x_2, \dots, x_n$ can be understood as one long word, or as n one-character words, or several successive words obtained by dividing the sequence $x_1, x_2, \dots, x_n$ in arbitrary places.

We are interested in probability $P_n(y_1, y_2 \dots, y_m)$ of transmitting the word $y_1, y_2 \dots, y_m$ from time moment n, more exactly in time moments $n, n + 1, \dots, n + m - 1$. This probability can be calculated as follows:

$$P_n(y_1, y_2, \dots, y_m) = \sum_{(x_1,\dots,x_{n-1})\in X^{n-1}} P(x_1, x_2, \dots, x_{n-1}, y_1, y_2, \dots, y_m)\ . \tag{3.4}$$

Definition 3.3. The source $\mathcal{Z} = (X^*, P)$ is called **stationary** if the probabilities $P_i(x_1, x_2, \dots, x_n)$ for $i = 1, 2, \dots$ do not depend on i,
i. e., if for every i and every $x_1, x_2 \dots, x_n \in X^n$

$$P_i(x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n)\ .$$

Denote by X_i the discrete random variable describing the transmission one character from the source in time instant i. Then the event "*The source transmitted the character x in time instant i*" can be written down as $[X_i = x]$ and hence $P([X_i = x]) = P_i(x)$. Generating the word $x_1, x_2, \dots, x_n$ in time i is the event $[X_i = x_1] \cap [X_{i+1} = x_2] \cap \dots \cap [X_{i+n-1} = x_n]$, shortly $[X_i = x_1, X_{i+1} = x_2, \dots, X_{i+n-1} = x_n]$. Therefore we can write

$$P([X_i = x_1, X_{i+1} = x_2, \dots, X_{i+n-1} = x_n]) = P_i(x_1, x_2, \dots, x_n).$$

Definition 3.4. The source $\mathcal{Z} = (X^*, P)$ is called **independent**, or **memoryless** if for arbitrary $i,\ j,\ n,\ m$ such that $i + n \le j$ it holds:

$$P\Big([X_i = x_1, X_{i+1} = x_2, \dots, X_{i+n-1} = x_n] \cap \\ \cap [X_j = y_1, X_{j+1} = y_2, \dots, X_{j+m-1} = y_m]\Big) = \\ = P\big([X_i = x_1, X_{i+1} = x_2, \dots, X_{i+n-1} = x_n]\big). \\ .P\big([X_j = y_1, X_{j+1} = y_2, \dots, X_{j+m-1} = y_m]\big).$$

The source is independent, or memoryless if generating of an arbitrary word in time j does not depend on anything transmitted before time j

The source transmitting in Slovak language is not memoryless. Černý in [5] shows that there are many Slovak words containing "ZA" but there are no Slovak words containing "ZAZA". It is $P(ZA) > 0$ and by assumption of memorylessness it should be $P(ZAZA) = P(ZA).P(ZA) > 0$ but $P(ZAZA) = 0$.

From a short term period Slovak (or any other) language could be considered stationary, but languages change during centuries – some ancient words disappear and new ones appear (radio, television, internet, computer, etc.) The stationarity of source is one of basic assumptions under which it is possible to obtain usable results in the information theory. From the short term point of view this assumption is fulfilled. Hence we will suppose that the sources we will work with are all stationary.

3.3 Entropy of source

Let $\mathcal{Z} = (Z^*, P)$ be a stationary source with source alphabet $Z = \{a_1, a_2, \dots, a_m\}$. We want to know the mean value of information obtainable from the information about the character generated in time 1. Transmission of a character in an arbitrary time can be regarded as the execution of the experiment

$$\mathbf{B} = \big\{\{a_1\}, \{a_2\}, \dots, \{a_m\}\big\}$$

with probabilities $p_1 = P(a_1)$, $p_2 = P(a_2)$, $\dots$, $p_m = P(a_m)$. The entropy of this experiment is $H(\mathbf{B}) = H(p_1, p_2, \dots, p_m)$ – the mean value of information obtained by this experiment.

Now let us calculate the amount of information of two first successive characters generated by a stationary source $\mathcal{Z} = (Z^*, P)$. The corresponding experiment will be now:

$$\mathbf{C}_2 = \big\{(a_{i_1}, a_{i_2}) \mid a_{i_1} \in Z,\ a_{i_2} \in Z\big\}\ .$$

The former experiment $\mathbf{B}$ can be represented as:

$$\mathbf{B} = \big\{\{a_1\} \times Z,\ \{a_2\} \times Z,\ \dots,\ \{a_m\} \times Z\big\}\ .$$

Define $\mathbf{D} = \big\{Z \times \{a_1\},\ Z \times \{a_2\},\ \dots,\ Z \times \{a_m\}\big\}$, then $\mathbf{C}_2 = \mathbf{B} \wedge \mathbf{D}$.
From stationarity of the source $\mathcal{Z} = (Z^*, P)$ it follows:

$$H(\mathbf{D}) = H(\mathbf{B}) = H(p_1, p_2, \dots, p_m).$$

By theorem 2.17 (page 49) it holds:

$$H(\mathbf{C}_2) = H(\mathbf{B} \wedge \mathbf{D}) \le H(\mathbf{B}) + H(\mathbf{D}) = 2.H(\mathbf{B})\ .$$

We prove this property for words of the length n by mathematical induction on n.
Suppose that

$$\mathbf{C}_n = \big\{(a_{i_1}, a_{i_2}, \dots, a_{i_n}) \mid a_{i_k} \in Z,\ \text{for } k = 1, 2, \dots, n\big\}$$

and that $H(\mathbf{C}_n) \le n.H(\mathbf{B})$. The entropy of experiment $\mathbf{C}_n$ is the same as that of

$$\mathbf{C}'_n = \big\{(a_{i_1}, a_{i_2}, \dots, a_{i_n}) \times Z \mid a_{i_k} \in Z,\ \text{for } k = 1, 2, \dots, n\big\}\ .$$

Denote

$$\mathbf{C}_{n+1} = \big\{(a_{i_1}, a_{i_2}, \dots, a_{i_{n+1}}) \mid a_{i_k} \in Z,\ \text{for } k = 1, 2, \dots, n+1\big\},$$
$$\mathbf{D} = \big\{Z^n \times \{a_1\},\ Z^n \times \{a_2\},\ \dots,\ Z^n \times \{a_m\}\big\},$$

then

$$\begin{aligned} H(\mathbf{C}_{n+1}) = H(\mathbf{C}'_n \wedge \mathbf{D}) \le H(\mathbf{C}'_n) + H(\mathbf{D}) \le \\ \le n.H(\mathbf{B}) + H(\mathbf{B}) = (n+1).H(\mathbf{B})\ . \end{aligned}$$

We have proved that for all integer $n > 0$ it holds

$$H(\mathbf{C}_n) \le n.H(\mathbf{B}),\ \text{i. e.,}\ \frac{1}{n}H(\mathbf{C}_n) \le H(\mathbf{B}).$$

We can see that in the case of stationary source the mean value of entropy per one character $\dfrac{1}{n}H(\mathbf{C}_n)$ is not greater than the entropy $H(\mathbf{B})$ of the first character. This leads to the idea to define the entropy of the source as the average entropy per character for very long words.

Definition 3.5. Let $\mathcal{Z} = (Z^*, P)$ be a source of information. Let exists the limit

$$H(\mathcal{Z}) = -\lim_{n\to\infty} \frac{1}{n} \ . \sum_{(x_1,\dots,x_n)\in Z} P(x_1, x_2, \dots, x_n). \log_2 P(x_1, x_2, \dots, x_n). \qquad (3.5)$$

Then the number $H(\mathcal{Z})$ is called **entropy of the source** $\mathcal{Z}$.

The following theorem says how to calculate the entropy of a stationary independent source $\mathcal{Z} = (Z^*, P)$

Theorem 3.1. *Let (Z^*, P) be a stationary independent source. Then*

$$H(\mathcal{Z}) = -\sum_{x\in Z} P(x). \log_2 P(x). \qquad (3.6)$$

Proof. It holds:

$$\sum_{(x_1,\dots,x_n)\in Z} P(x_1, x_2, \dots, x_n). \log_2(P(x_1, x_2, \dots, x_n)) =$$

$$= \sum_{(x_1,\dots,x_n)\in Z} P(x_1).P(x_2), \dots, P(x_n).\big[\log_2 P(x_1) + \log_2 P(x_2) + \dots + \log_2 P(x_n)\big] =$$

$$= \sum_{(x_1,\dots,x_n)\in Z} P(x_1).P(x_2), \dots, P(x_n). \log_2 P(x_1) +$$

$$+ \sum_{(x_1,\dots,x_n)\in Z} P(x_1).P(x_2), \dots, P(x_n). \log_2 P(x_2) +$$

$$+ \dots\dots\dots\dots\dots\dots \dots +$$

$$+ \sum_{(x_1,\dots,x_n)\in Z} P(x_1).P(x_2), \dots, P(x_n). \log_2 P(x_n) =$$

$$= \sum_{x_1\in Z} P(x_1). \log_2 P(x_1) \ . \underbrace{\sum_{(x_2,\dots,x_n)\in Z} P(x_2).P(x_3), \dots, P(x_n)}_{=1} + \dots =$$

$$= \sum_{x_1\in Z} P(x_1).\log_2 P(x_1) + \sum_{x_2\in Z} P(x_2).\log_2 P(x_2) + \cdots + \sum_{x_3\in Z} P(x_3).\log_2 P(x_3) =$$
$$= n.\sum_{x\in Z} P(x).\log_2 P(x).$$

The desired assertion of the theorem follows from the last expression. ■

Remark. The assumption of source stationarity without independence is not enough to guarantee the existence of the limit (3.5).

Theorem 3.2. Shannon – Mac Millan. *Let $\mathcal{Z} = (Z^*, P)$ be a stationary independent source with entropy $H(\mathcal{Z})$. Then for every $\varepsilon > 0$ there exists an integer $n(\varepsilon)$ such that for all $n \geq n(\varepsilon)$ it holds:*

$$P\left\{x_1, \dots x_n \in Z^n \mid \left|\frac{1}{n}.\log_2 P(x_1, \dots x_n) + H(\mathcal{Z})\right| \geq \varepsilon\right\} < \varepsilon\ . \qquad (3.7)$$

We introduce this theorem in its simplest form and without the proof. It holds also for much more general sources including natural languages. However, the mentioned more general sources can hardly be defined and studied without an application of the measure theory.

The interested reader can find some more general formulations of Shannon – Mac Millan theorem in the book [9]. The cited book uses as simple mathematical tools as possible.

Denote

$$E(n, \varepsilon) = \left\{x_1, \dots x_n \in Z^n \mid \left|\frac{1}{n}.\log_2 P(x_1, \dots x_n) + H(\mathcal{Z})\right| < \varepsilon\right\} \qquad (3.8)$$

Shannon – Mac Millan theorem says that for every $\varepsilon > 0$ there exists a set $E(n, \varepsilon)$ for which it holds $P(E(n, \varepsilon)) > 1 - \varepsilon$.
It holds:

$$(x_1, \dots, x_n) \in E(n, \varepsilon) \iff -\varepsilon < \frac{1}{n}\log_2 P(x_1, \dots, x_n) + H(\mathcal{Z}) < \varepsilon \iff$$
$$\iff -n(H(\mathcal{Z}) + \varepsilon) < \log_2 P(x_1, \dots, x_n) < -n(H(\mathcal{Z}) - \varepsilon) \iff$$
$$\iff 2^{-n(H(\mathcal{Z})+\varepsilon)} < P(x_1, \dots, x_n) < 2^{-n(H(\mathcal{Z})-\varepsilon)}$$

Let $|E(n,\varepsilon)|$ be the number of elements of the set $E(n,\varepsilon)$. Since the probability of every element of $E(n,\varepsilon)$ is greater than $2^{-n(H(\mathcal{Z})+\varepsilon)}$, we have

$$1 \geq P(E(n,\varepsilon)) > |E(n,\varepsilon)|.2^{-n(H(\mathcal{Z})+\varepsilon)}.$$

At the same time the probability of every element of $E(n,\varepsilon)$ is less than $2^{-n(H(\mathcal{Z})-\varepsilon)}$ from which it follows:

$$1-\varepsilon < P(E(n,\varepsilon)) < |E(n,\varepsilon)|.2^{-n(H(\mathcal{Z})-\varepsilon)}.$$

From the last two inequalities we have:

$$(1-\varepsilon).2^{n(H(\mathcal{Z})-\varepsilon)} < |E(n,\varepsilon)| < 2^{n(H(\mathcal{Z})+\varepsilon)} \tag{3.9}$$

The set of all words of the length n is decomposed into a significant set (in the sense of probability) $E(n,\varepsilon)$ with approximately $2^{n.H(\mathcal{Z})}$ words, the probability of which is approximately equal to $2^{H(\mathcal{Z})}$, and to the rest of words with negligible total probability.

Slovak language uses 26 letters of alphabet without diacritic marks and 15 letters with the diacritic marks á, č, ď, é, í, ľ, ĺ, ň, ó, ô, ť, ú, ý, ž.

Suppose that Slovak language uses alphabet Z with 40 letters. Surely the entropy of Slovak language is less than 2. The number of all 8-letter words of Z is 40^8, the number of significant words is $|E(8,\varepsilon)| \approx 2^{n.H(\mathcal{Z})} = 2^{8.2} = 2^{16}$.

$$\text{It holds:} \qquad \frac{|E(8,\varepsilon)|}{|Z|} \approx \frac{2^{16}}{40^8} = 6.10^{-8}.$$

The set $E(8,\varepsilon)$ of all significant 8-letter words contains only 6 millionths of one percent of all 8-letter words.

3.4 Product of information sources

Definition 3.6. Let $\mathcal{Z}_1 = (A^*, P_1)$, $\mathcal{Z}_2 = (B^*, P_2)$ be two sources. The **product of sources** $\mathcal{Z}_1$**,** $\mathcal{Z}_2$ is the source $\mathcal{Z}_1 \times \mathcal{Z}_2 = ((A \times B)^*, P)$, where $(A \times B)$ is the Cartesian product of sets A and B (i. e., the set of all ordered couples (a, b) with $a \in A$ and $b \in B$), and where $P(e) = 1$ (the probability of transmitting of the empty word in 0 time moments) and where

$$P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\big) = P(a_1, a_2, \dots, a_n).P(b_1, b_2, \dots, b_n) \qquad (3.10)$$

for an arbitrary $a_i \in A$, $b_j \in B$, $i, j \in \{1, 2, \dots, n\}$.

Theorem 3.3. *The product* $\mathcal{Z}_1 \times \mathcal{Z}_2$ *of sources* $\mathcal{Z}_1$, $\mathcal{Z}_2$ *is correctly defined, i. e., the probability function* P *fulfills* (3.1), (3.2), (3.3) *from definition 3.2.*

Proof. Let $a_i \in A$, $b_i \in B$ for $i = 1, 2, \dots, n$, let $p_j \in A$, $q_j \in B$ for $j = 1, 2, \dots, m$. We are to prove (3.1), (3.2), (3.3) from definition 3.2 (page 52). These equations are now in the following form:

1. $$P(e) = 1 \qquad (3.11)$$
2. $$\sum_{(a_1,b_1),\dots,(a_n,b_n)\in(A\times B)^n} P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\big) = 1 \qquad (3.12)$$
3. $$\sum_{(p_1,q_1),\dots,(p_m,q_m)\in(A\times B)^m} P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n), (p_1, q_1), \dots, (p_m, q_m)\big) =$$ $$= P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\big) \qquad (3.13)$$

First equation (3.11) follows from definition 3.2 of the product of sources. Now we will prove the third equation.

$$\sum_{(p_1,q_1),\dots,(p_m,q_m)\in(A\times B)^m} P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n), (p_1, q_1), \dots, (p_m, q_m)\big) =$$
$$= \sum_{p_1p_2\dots p_m\in A^m} \sum_{q_1q_2\dots q_m\in B^m} P(a_1, \dots, a_n, p_1, \dots, p_m).P(b_1, \dots, b_n, q_1, \dots, p_m) =$$
$$= \sum_{p_1p_2\dots p_m\in A^m} P(a_1, \dots, a_n, p_1, \dots, p_m) \sum_{q_1q_2\dots q_m\in B^m} P(b_1, \dots, b_n, q_1, \dots, p_m) =$$
$$= P(a_1, a_2, \dots, a_n)\ .\ P(b_1, b_2, \dots, b_n) = P\big((a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\big)\ .$$

The second equation can be proved by a similar way. ■

Theorem 3.4. *Let $\mathcal{Z}_1$, $\mathcal{Z}_2$ be two sources with entropies $H(\mathcal{Z}_1)$, $H(\mathcal{Z}_2)$. Then*

$$H(\mathcal{Z}_1 \times \mathcal{Z}_2) = H(\mathcal{Z}_1) + H(\mathcal{Z}_2)\ . \tag{3.14}$$

Proof.

$$H(\mathcal{Z}_1 \times \mathcal{Z}_2) =$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{(a_1,b_1),\dots,(a_n,b_n)\in(A\times B)^n} P\big((a_1,b_1),\dots,(a_n,b_n)\big).\log_2 P\big((a_1,b_1),\dots,(a_n,b_n)\big) =$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{(a_1,b_1),\dots,(a_n,b_n)\in(A\times B)^n} \Bigg\{ P(a_1,\dots,a_n).P(b_1,\dots,b_n).$$

$$.\,[\log_2 P(a_1,\dots,a_n) + \log_2 P(b_1,\dots,b_n)] \Bigg\} =$$

$$= \lim_{n\to\infty} \frac{1}{n} \Bigg[\sum_{(a_1,b_1),\dots,(a_n,b_n)\in(A\times B)^n} P(a_1,\dots,a_n).P(b_1,\dots,b_n).\log_2 P(a_1,\dots,a_n) +$$

$$+ \sum_{(a_1,b_1),\dots,(a_n,b_n)\in(A\times B)^n} P(a_1,\dots,a_n).P(b_1,\dots,b_n).\log_2 P(b_1,\dots,b_n) \Bigg] =$$

$$= \lim_{n\to\infty} \frac{1}{n} \Bigg[\sum_{a_1,\dots,a_n\in A^n} P(a_1,\dots,a_n).\log_2 P(a_1,\dots,a_n)\ .\ \underbrace{\sum_{b_1,\dots,b_n\in B^n} P(b_1,\dots,b_n)}_{=1} +$$

$$+ \sum_{b_1,\dots,b_n\in B^n} P(b_1,\dots,b_n)\log_2 P(b_1,\dots,b_n)\ .\ \underbrace{\sum_{a_1,\dots,a_n\in A^n} P(a_1,\dots,a_n)}_{=1} \Bigg] =$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{a_1,\dots,a_n\in A^n} P(a_1,\dots,a_n).\log_2 P(a_1,\dots,a_n) +$$

$$+ \lim_{n\to\infty} \frac{1}{n} \sum_{b_1,\dots,b_n\in B^n} P(b_1,\dots,b_n)\log_2 P(b_1,\dots,b_n) = H(\mathcal{Z}_1) + H(\mathcal{Z}_2).$$

■

Definition 3.7. Let $\mathcal{Z} = (A^*, P)$ be a source. Define $\mathcal{Z}^2 = \mathcal{Z} \times \mathcal{Z}$ and further by induction $\mathcal{Z}^n = \mathcal{Z}^{n-1} \times \mathcal{Z}$.

The source $\mathcal{Z}^n = \underbrace{\mathcal{Z} \times \mathcal{Z} \times \cdots \times \mathcal{Z}}_{n\text{-times}}$ is the source with alphabet A^n. Applying the theorem 3.4 and using mathematical induction we can get the following theorem:

Theorem 3.5. *Let $\mathcal{Z}$ be a source withe entropy $H(\mathcal{Z})$. Then it holds for the entropy $H(\mathcal{Z}^n)$ of $\mathcal{Z}^n$:*

$$H(\mathcal{Z}^n) = n.H(\mathcal{Z}) \tag{3.15}$$

Definition 3.8. Let $\mathcal{Z} = (A^*, P)$ be a source. Denote $\mathcal{Z}_{(k)} = \big((A^k)^*, P_{(k)}\big)$ the source with the alphabet A^k, where $P_{(k)}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$ for $\mathbf{a}_i \in A^k$, $\mathbf{a}_i = a_{i1}a_{i2}\dots a_{ik}$ is defined as follows:

$$P_{(k)}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = P(a_{11}, a_{12}, \dots, a_{1k}, a_{21}, a_{22}, \dots, a_{2k}, \dots, a_{n1}, a_{n2}, \dots, a_{nk})$$

Remark. The source $\mathcal{Z}_{(k)}$ is obtained from the source $\mathcal{Z}$ in such a way that we will take from the source $\mathcal{Z}$ every k-th moment the whole k-letter output word and we will consider this word of length k as a single letter of alphabet A^k.

Attention! There is an essential difference between $\mathcal{Z}_{(k)}$ and $\mathcal{Z}^k$. While the output words of the source $\mathcal{Z}_{(k)}$ are k-tuples of successive letters of original source $\mathcal{Z}$ and their letters can be dependent, the words of the source $\mathcal{Z}^k$ originated as k-tuples of outcomes of k mutually independent identical sources and the letters are independent in separate output words.
However, in the case of independent stationary source $\mathcal{Z}$, the sources $\mathcal{Z}_{(k)}$ and $\mathcal{Z}^k$ are equivalent.

Theorem 3.6. *Let $\mathcal{Z}$ is a source with entropy $H(\mathcal{Z})$. Let $H(\mathcal{Z}_{(k)})$ be the entropy of the source $\mathcal{Z}_{(k)}$. Then*

$$H(\mathcal{Z}_{(k)}) = k.H(\mathcal{Z}) \tag{3.16}$$

Proof. It holds:

$$H(\mathcal{Z}_{(k)}) = \lim_{n\to\infty} \frac{1}{n} \sum_{\mathbf{a}_1,\dots,\mathbf{a}_n \in A^n} P_{(k)}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) =$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{a_{ij}\in A \text{ for } 1\le i\le n,\ 1\le j\le k} P(a_{11}, \dots, a_{1k}, a_{21}, \dots, a_{2k}, \dots, a_{n1}, \dots, a_{nk}) =$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{x_1,x_2,\dots,x_{n.k}\in A} P(x_1,x_2,\dots,x_{n.k}) =$$

$$= k.\left[\lim_{n\to\infty} \frac{1}{k.n} \sum_{x_1,x_2,\dots,x_{n.k}\in A} P(x_1,x_2,\dots,x_{n.k})\right] = k.H(\mathcal{Z}) \tag{3.17}$$

■

The last theorem says that the mean value of information per one k-letter word of the source $\mathcal{Z}$ (i. e., one letter of the source $\mathcal{Z}_{(k)}$) is the k-multiple of the mean value of information per one letter. This is not a surprising fact. One would expect that the mean information per letter will be the same regardless we take from the source single letters, or k-letter words.

3.5 Source of information as a measure product space*

In spite of the fact that the model of the last section allows to define and to prove many useful properties of information sources, it has several disadvantages. The principal one of them is that the function $P(x_1, x_2, \dots, x_n)$ is not a probability measure on the set Z^* of all words of alphabet Z

There exists a model which has not the disadvantages mentioned above but it requires the utilization of measure theory. This part is based on the theory of extension of measures and the theory of product measure spaces (3-rd and 7-th chapter of the book [6]) and on results of the ergodic theory [3].
Let $Z = \{a_1, a_2, \dots, a_r\}$. Denote

$$\Omega = \prod_{i=-\infty}^{\infty} Z \tag{3.18}$$

the set of all infinite sequences of elements from Z of the form

$$\boldsymbol{\omega} = (\dots, \omega_{-2}, \omega_{-1}, \omega_0, \omega_1, \omega_2, \dots)\ .$$

Let X_i for every integer i be a function defined by the formula:

$$X_i(\boldsymbol{\omega}) = \omega_i\ .$$

Let $E_1, E_2, \dots E_k$ are subsets of Z. The **cylinder** is the set

$$C_n(E_1, E_2, \dots, E_k) =$$
$$= \{\boldsymbol{\omega} \mid X_n(\boldsymbol{\omega}) \in E_1,\ X_{n+1}(\boldsymbol{\omega}) \in E_2, \dots,\ X_{n+k-1}(\boldsymbol{\omega}) \in E_k\}.$$

Let $x_1, x_2, \dots, x_k$ is an arbitrary finite sequence of elements from Z. The **elementary cylinder** is the set

$$EC_n(x_1, x_2, \dots, x_k) =$$
$$= \{\boldsymbol{\omega} \mid X_n(\boldsymbol{\omega}) = x_1,\ X_{n+1}(\boldsymbol{\omega}) = x_2, \dots,\ X_{n+k-1}(\boldsymbol{\omega}) = x_k\}.$$

Remember that we can write:

$$C_n(E_1, E_2, \dots, E_k) = \dots \times Z \times Z \times E_1 \times E_2 \times \dots \times E_k \times Z \times Z \times \dots,$$

resp.

$$EC_n(x_1, x_2, \dots, x_k) = \dots \times Z \times Z \times \{x_1\} \times \{x_2\} \times \dots \times \{x_k\} \times Z \times Z \times \dots$$

Elementary cylinder $EC_n(x_1, x_2, \dots, x_k)$ represents the situation when the source transmits the word $(x_1, x_2, \dots, x_k)$ in time moments n, $n+1, \dots$, $n+k-1$.

Denote by $\mathcal{F}_0$ the set of all cylinders. The set $\mathcal{F}_0$ contains the empty set (e. g. cylinder $C_1(\emptyset)$ is empty), it contains Ω (since $C_1(Z) = \Omega$), it is closed under the formation of complements, finite intersections and finite unions. Therefore, there exists the unique smallest σ-algebra $\mathcal{F}$ of subsets of Ω containing $\mathcal{F}_0$. See [6], (chapter 7 – Product Spaces).

Definition 3.9. The **source of information with alphabet** Z is the probability space $\mathcal{Z} = (\Omega, \mathcal{F}, P)$ where $\Omega = \prod_{i=-\infty}^{\infty} Z$, $\mathcal{F}$ is the smallest σ-algebra of subsets of Ω containing all cylinders and where P is a probability measure on σ-algebra $\mathcal{F}$.

Remark. Since every cylinder can be written as a finite union of elementary cylinders it would be enough to define $\mathcal{F}$ as the smallest σ-algebra containing all elementary cylinders.

Remark. The probability space $(\Omega, \mathcal{F}, P)$ from definition 3.9 is called **infinite product space** in the measure theory resources (e. g. [6]).

Definition 3.9 fulfills what we required. We have defined the source as a probability space in which a transmission of arbitrary word in arbitrary time is modelled as an event – an elementary cylinder – and in which various general properties of sources can be studied.

Definition 3.10. Let $(\Omega, \mathcal{F}, P)$ be a probability space, let $T : \Omega \to \Omega$ is a bijection on Ω. Denote for $A \subseteq \Omega$:

$$T^{-1}A = \{\omega \mid T(\omega) \in A\} \qquad T(A) = \{T(\omega) | \ \omega \in A\}. \tag{3.19}$$

$T^{-n}A$ can be defined by induction as follows: $T^{-1}A$ is defined in (3.19). If $T^{-n}A$ is defined then define: $T^{-(n+1)}A = T^{-1}(T^{-n}A)$.
The mapping T is called **measurable** if for every $A \in \mathcal{F}$ it holds $T^{-1}A \in \mathcal{F}$.
The mapping T is called **measure preserving** if T is a bijection, both T and T^{-1} are measurable and for every $A \in \mathcal{F}$ it holds $P(T^{-1}A) = P(A)$.
We say that the mapping T is **mixing** if T is a measure preserving and for arbitrary sets $A, B \in \mathcal{F}$ it holds:

$$\lim_{n\to\infty} P(A \cap \ T^{-n}B) = P(A).P(B). \tag{3.20}$$

We say that the set $B \in \mathcal{F}$ is T**-invariant** if

$$T^{-1}B = B.$$

We say that the mapping T is **ergodic**, if T is measure preserving and the only T-invariant sets are the sets with measure 0 or 1.

Theorem 3.7. *Let T be a mixing mapping. Then T is ergodic.*

Proof. T is measure preserving. It remains to prove that the only T-invariant sets have measure 0 or 1.

Let $B \in \mathcal{F}$ is T-invariant, let $A \in \mathcal{F}$ is an arbitrary measurable set. Then $T^{-n}B = B$ and hence:

$$\begin{aligned}
\lim_{n\to\infty} P(A \cap \ T^{-n}B) &= P(A).P(B) \\
P(A \cap B) &= P(A).P(B) \quad \text{for every } A \in \mathcal{F} \\
P(B \cap B) &= P(B).P(B) \\
P(B) &= (P(B))^2 \\
(P(B))^2 - P(B) &= 0 \\
P(B)[1 - P(B)] &= 0
\end{aligned}$$

From the last equation it follows that $P(B) = 0$ or $P(B) = 1$. ■

Theorem 3.8. Ergodic theorem. *Let T be an ergodic mapping on a probability space $(\Omega, \mathcal{F}, P)$. Then it holds for every measurable set $A \in \mathcal{F}$ and for almost all*[1] $\omega \in \Omega$*:*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \chi_A\left(T^i(\omega)\right) = P(A), \tag{3.21}$$

where $\chi_A(\omega)$ is the indicator of the set A, i. e., $\chi_A(\omega) = 1$ if $\omega \in A$, otherwise $\chi_A(\omega) = 0$.

Proof. The proof of the ergodic theorem is complicated, the interested reader can find it in [3]. ■

Definition 3.10 and theorems 3.7, 3.8 hold for arbitrary general probability spaces.

Let us return now to our source of information $\mathcal{Z} = (\Omega, \mathcal{F}, P)$ where Ω is a set of infinite (from both sides) sequences of letters from a finite alphabet Z. Define the bijection T on the set Ω:

$$X_n(T(\boldsymbol{\omega})) = X_{n+1}(\boldsymbol{\omega}) \tag{3.22}$$

$$\begin{aligned} \boldsymbol{\omega} &= \dots, \omega_{-2}, \omega_{-1}, \omega_0, \omega_1, \omega_2, \dots \\ T(\boldsymbol{\omega}) &= \dots, \omega_{-1}, \omega_0\ \ , \omega_1, \omega_2, \omega_3, \dots \end{aligned}$$

The mapping T "shifts" the sequence $\boldsymbol{\omega}$ of letters one position to the left – that is why it is sometimes called **left shift**.

Let $T^n(\boldsymbol{\omega})$ be n-times applied left shift T:

$$T^n(\boldsymbol{\omega}) = \underbrace{T(T(\dots T(\boldsymbol{\omega})\dots))}_{n\text{-times}}.$$

Here is the exact definition by induction: $T^1(\boldsymbol{\omega}) = T(\boldsymbol{\omega}),\ T^{n+1}(\boldsymbol{\omega}) = T(T^n(\boldsymbol{\omega}))$.

$X_0(\boldsymbol{\omega})$ is the letter of the sequence $\boldsymbol{\omega}$ transmitted by the source in time 0, $X_0(T(\boldsymbol{\omega}))$ is the letter of the sequence $\boldsymbol{\omega}$ transmitted by the source in time 1, $X_0(T^2(\boldsymbol{\omega}))$ is the letter of the sequence $\boldsymbol{\omega}$ transmitted by the source in time 2, etc.

[1] The term "for almost all $\omega \in \Omega$" means: for all $\omega \in \Omega - \phi$ where $\phi \subset \Omega$ has zero probability measure – $P(\phi) = 0$.

Let us have a cylinder $C_n(E_1, E_2, \dots, E_k)$, then

$$T^{-1}C_n(E_1, E_2, \dots, E_k) = C_{n+1}(E_1, E_2, \dots, E_k),$$

$$T^{-m}C_n(E_1, E_2, \dots, E_k) = C_{n+m}(E_1, E_2, \dots, E_k).$$

The properties of left shift T with probability measure P fully characterise all properties of the source. That is why the quadruple $\mathcal{Z} = (\Omega, \mathcal{F}, P, T)$ can be considered as the source of information.

Definition 3.11. We say that the source $\mathcal{Z} = (\Omega, \mathcal{F}, P, T)$ is **stationary** if the left shift T is a measure preserving mapping.

Theorem 3.9. *Let $\mathcal{F}_0$ be an algebra generating the σ-algebra $\mathcal{F}$. Let $T^{-1}A \in \mathcal{F}_0$ and $P(T^{-1}A) = P(A)$ for every $A \in \mathcal{F}_0$. Then T is measure preserving mapping.*

Proof. The proof of this theorem requires knowledge of measure theory procedures. That is why we omit it. The reader can find it in [3]. ■

This theorem is typical for the approach to modelling and studying properties of sources by means of measure theory and for the modelling sources as product spaces. In many cases it suffices to show some property only for elements of generating algebra $\mathcal{F}_0$ and the procedures of measure theory extend this property to all events of generated σ-algebra $\mathcal{F}$. The consequence of this theorem is the fact that for the proof of stationarity of a source $\mathcal{Z}$ it suffices to prove that the shift T preserves the measure of cylinders.

Example 3.1. Let $\mathcal{Z} = (\Omega, \mathcal{F}, P, T)$ be a source with a finite alphabet

$$Z = \{a_1, a_2, \dots, a_r\}.$$

Let $p_1 = P(a_1),\ p_2 = P(a_2), \dots,\ p_r = P(a_r)$ be probabilities, $\sum_{i=1}^{r} p_i = 1$. For $E \subseteq Z$ it holds $P(E) = \sum_{a \in E} p(a)$.
The measure P is defined by the set of its values on the set of elementary cylinders by the following equation

$$P\big(EC_n(a_{i_1}, a_{i_2}, \dots, a_{i_k})\big) = p_{i_1}.p_{i_2}, \dots, p_{i_k}. \tag{3.23}$$

This measure can be extended to algebra $\mathcal{F}_0$ of all cylinders as follows:

$$P\big(C_n(E_1, E_2, \dots, E_k)\big) = P(E_1).P(E_2).\dots.P(E_k). \tag{3.24}$$

Theorem 3.10. *Let $\mathcal{F}_0$ be an algebra generating the σ-algebra $\mathcal{F}$, let $T : \Omega \to \Omega$ be a bijection. Suppose $T^{-1}A \in \mathcal{F}$ and $P(T^{-1}A) = P(A)$ for all $A \in \mathcal{F}_0$. Then T is a measure preserving mapping.*

Proof. For the proof see [3]. ■

Measure theory guarantees the existence of the unique measure P on $\mathcal{F}$ fulfilling (3.23). Let $\mathcal{Z} = (\Omega, \mathcal{F}, P, T)$ be a source with probability P fulfilling (3.23) resp. (3.24). Then the shift T is called **Bernoulli shift**. The source $\mathcal{Z}$ is stationary and independent. The question is whether it is ergodic.

Let $A = C_s(E_1, E_2, \dots, E_k)$, $B = C_t(F_1, F_2, \dots, F_l)$ be two cylinders. If n is large enough the set $A \cap T^{-n}A$ is in the form

$$\begin{aligned} &A \cap T^{-n}B = \\ &= \dots \times Z \times Z \times E_1 \times E_2 \times \dots \times E_k \times Z \times \dots \times Z \times F_1 \times F_2 \times \dots \times F_l \times Z \times Z \dots \end{aligned}$$

which is cylinder $C_s(E_1, E_2, \dots, E_k, Z, \dots, Z, F_1, F_2, \dots, F_l)$ whose probability is by (3.24) $\prod_{i=1}^{k} P(E_i).\prod_{j=1}^{l} P(F_j) = P(A).P(B)$. For A, B cylinders we have:

$$\lim_{n\to\infty} P(A \cap T^{-n}B) = P(A).P(B) \tag{3.25}$$

Once again we can make use of another theorem of measure theory:

Theorem 3.11. *Let $\mathcal{F}_0$ is an algebra generating σ-algebra $\mathcal{F}$, T is a measure preserving mapping on Ω. If* (3.25) *holds for all $A, B \in \mathcal{F}_0$ then T is mixing.*

Therefore Bernoulli shift is mixing and hence ergodic.

Let Ω, $\mathcal{F}$, T be as in the previous example. Let P be a general probability measure on $\mathcal{F}$. Define $P(e) = 0$ for the empty word e and for arbitrary integer $n > 0$ and $(x_1, x_2, \ldots, x_n) \in Z^n$

$$\begin{aligned} &P(x_1, x_2, \ldots, x_n) = \\ &\qquad = P\{\boldsymbol{\omega} \mid X_1(\boldsymbol{\omega}) = x_1, X_2(\boldsymbol{\omega}) = x_2, \ldots, X_n(\boldsymbol{\omega}) = x_n\}. \end{aligned} \tag{3.26}$$

Then $P : Z^* \to \langle 0, 1\rangle$. It is easy to show that the function P fulfills (3.1), (3.2) and (3.3) from definition 3.2 (page 52) and hence (Z^*, P) is an information source in the sense of definition 3.2.

Let P be a probability measure on $\mathcal{F}$ such that the left shift T is measure preserving. The statement "T is measure preserving" is equivalent with the assertion that

$$\begin{aligned} &P_i(x_1, x_2, \ldots, x_n) = \\ &\qquad = P\{\boldsymbol{\omega} \mid X_i(\boldsymbol{\omega}) = x_1, X_{i+1}(\boldsymbol{\omega}) = x_2, \ldots, X_{i+n-1}(\boldsymbol{\omega}) = x_n\} \end{aligned} \tag{3.27}$$

does not depend on i which is equivalent with definition 3.3 (page 54) of stationarity of the source (Z^*, P). We showed that the source $(\Omega, \mathcal{F}, P, T)$ can be thought of as the source (Z^*, P).

On the other hand, given a source (Z^*, P) with function $P : Z^* \to \langle 0, 1\rangle$ fulfilling (3.1), (3.2) and (3.3) from definition 3.2, we can define the product space $(\Omega, \mathcal{F}, P, T)$ where Ω is the product space defined by (3.18), $\mathcal{F}$ is the smallest unique σ-algebra containing all elementary cylinders, T is the left shift on Ω and P is the unique probability measure such that for arbitrary elementary cylinder it holds (3.27). Measure theory guarantees the existence and uniqueness of such measure P. Thus, the source (Z^*, P) can be studied as the source $(\Omega, \mathcal{F}, P, T)$. The reader can find corresponding definitions and theorems in [6], chapter 3 and 7, and in [3].

We have two similar models for source of information – an elementary model (Z^*, P) and a product space model $(\Omega, \mathcal{F}, P, T)$. We could easy formulate several properties of sources in both models. Unfortunately the ergodicity of the source which was in product space model formulated as "*the only T-invariant events have probability* 0 *or* 1" cannot be formulated in a similar simple way.

Source ergodicity is a very strong property. The entropy always exists for ergodic sources. Shannon – Mac Millan theorem (till now formulated only for stationary independent sources) holds for all ergodic sources.

As we have shown, natural language (e. g., Slovak, English, etc.) can be considered stationary but it is not independent. Let A, B be two words of natural language (i. e., elementary cylinders) then $T^{-n}B$ with large n is the event that the word B will be transmitted in far future. The larger time interval between transmitting both words A and B will be, the less the event $T^{-n}B$ will depend on the event A. Therefore, we can suppose that $\lim_{n\to\infty} P(A \cap T^{-n}B) = P(A).P(B)$, and hence that the shift T is mixing and by theorem 3.7 ergodic.

Natural language can be considered ergodic. Therefore Shann–Mac Millan theorem and many other important theorems hold for such languages. Most important ones of them are two Shannon's theorems on channel capacity (theorems 5.1 and 5.2, page 152).

Chapter 4

Coding theory

4.1 Transmission chain

General scheme of transmission chain is shown here:

$$\boxed{\text{Source of signal}} \to \boxed{\text{Encoder}} \to \boxed{\text{Channel}} \to \boxed{\text{Decoder}} \to \boxed{\text{Receiver}}$$

It can happen that a source of signal, a communication channel and a receiver use different alphabets. A radio studio has a song which is stored on CD in binary code. This code has to be converted to radio high frequency signal (ca 100 MHz) what is the signal of communication channel. A radio receiver turns this signal into sound waves (from 16 Hz to 20 kHz).

If one needs to transmit a message using only flash light capable to produce only symbols ”.”, ”—” and ”/” he has to encode the letters of his message into a sequence of mentioned symbols (e. g. using Morse alphabet)

The main purpose of encoding messages is to express the message in characters of alphabet of the communication channel. However, we can have also additional goals. We can require that the encoded message is as short as possible (data compression). On the other hand, we can request for such encoding which allows to detect whether a single error, (or some given limited number of errors), occurred during transmission. There are even ways of encoding capable to correct a given limited number of errors. Moreover, we want that the encoding and the decoding have low computational complexity.

Just mentioned requirements are conflicting and it is not easy to ensure every single one of them and even harder in combinations. The purpose of encoding is not to ensure the secrecy or security of messages, that is why it is necessary to make a difference between encoding and enciphering – data security is the objective of cryptography and not that of coding theory.

Coding theory deals with problems of encoding, decoding, data compression, error detecting and error correcting codes. This chapter contains fundamentals of coding theory.

4.2 Alphabet, encoding and code

Let $A = \{a_1, a_2, \dots, a_r\}$ be a finite set with r elements. The elements of A are called **characters**, the set A is called **alphabet**. The set

$$A^* = \bigcup_{i=1}^{\infty} A^i \cup \{e\}$$

where e is an empty word is called **set of all words of alphabet** A. The **length of the word** $\mathbf{a} \in A^*$ is the number of characters of the word $\mathbf{a}$. Define a binary operation $|$ on A^* called **concatenation of words** as follows: If $\mathbf{b} = b_1 b_2 \dots b_p$, $\mathbf{c} = c_1 c_2 \dots c_q$ are two words from A^*, then

$$\mathbf{b}|\mathbf{c} = b_1 b_2 \dots b_p c_1 c_2 \dots c_q.$$

The result of concatenation of two words is written without space, or any other separating character. Every word can be regarded as the concatenation of its arbitrary parts according as is convenient. So $01010001 = 0101|0001 = 010|100|01 = 0|1|0|1|0|0|0|1$.

Let $A = \{a_1, a_2, \dots, a_r\}$, $B = \{b_1, b_2, \dots, b_s\}$ are two alphabets. The **encoding** is a mapping

$$K : A \to B^*,$$

i. e., a recipe assigning to every character of alphabet A a word of alphabet B. Alphabet A is the **source alphabet**, the characters of A are **source characters**, alphabet B is the **code alphabet** and its characters are **code characters**. The set $\mathcal{K}$ of all words of the code alphabet is defined as

$$\mathcal{K} = \{\mathbf{b} \mid \mathbf{b} = K(a),\ a \in A\} = \{K(a_1), K(a_2), \dots, K(a_r)\}$$

is called the **code**, every word of the set $\mathcal{K}$ is the **code word**, other words of alphabet B are the **noncode words**.

Only injective encodings K are of practical importance – such that if $a_i,\ a_j \in A$ and $a_i \neq a_j$ then $K(a_i) \neq K(a_j)$. Therefore we will assume that K is injective. Every encoding K can be extended to the encoding K^* of source words by the formula:

$$K^*(a_{i_1}a_{i_2}\dots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\dots|K(a_{i_n}) \tag{4.1}$$

The encoding K^* is actually a sequential encoding of characters of the source word.

An encoding can assign code words of various lengths to various source characters. Very often we work with encodings where all code words have the same length. The **block encoding of the length** n is an encoding where all code words have the same length n. The corresponding code is the **block code** of the length n.

Example 4.1. Let $A = \{a, b, c, d\}$, $B = \{0, 1\}$, let $K(a) = 00,\ K(b) = 01,\ K(c) = 10,\ K(d) = 11$. Then the message *aabd* (i. e., the word in alphabet A) is encoded as $K^*(aabd) = 00000111$. After receiving the word 00000111 (and provided we know the mapping K), we know that every character of source alphabet was encoded into two characters of code alphabet and hence the only possible splitting of received message into code words is 00|00|01|11 what leads to unique decoding of received message. The encoding K is a block encoding of the length 2.

Example 4.2. The results of exams are 1, 2, 3, 4. We know that most frequent results are 2 and then 1. Other outcomes are rare. The code words of code alphabet $B = \{0, 1\}$ of length two would suffice to encode four results. But we want to assign a short code word to the frequent outcome 2. Therefore, we will use the following encoding: $K(1) = 01,\ K(2) = 0,\ K(3) = 011,\ K(4) = 111$. The message 1234 will be encoded as 01|0|011|111. When decoding the message 010011111, we have to decode it from behind. We cannot decode from start of the received message. If we receive a partial message 01111... we do not know whether it was transmitted as 0|111|1..., or 01|111..., or 011|11..., we cannot decode character by character, or more precisely, codeword by codeword.

Definition 4.1. We say that the encoding $K : A \to B^*$ is **uniquely decodable**, if every source word $a_1a_1 \dots a_n$ can be uniquely retrieved from the encoded message $K^*(a_1a_1 \dots a_n)$, i. e., if the mapping $K^* : A^* \to B^*$ is an injection.

Example 4.3. Extend the source alphabet from example 4.2 to $A = \{1, 2, 3, 4, 5\}$ and define encoding

$$K(1) = 01,\ K(2) = 0,\ K(3) = 011,\ K(4) = 111,\ K(5) = 101.$$

Note that K is an injection. Let us have the message 0101101. We have following possible ways of splitting this message into code words: 0|101|101, 01|01|101, 01|011|01, whereas these ways correspond to source words 255, 115, 131. We can see that in spite of fact that the encoding $K : A \to B^*$ is an injection, the corresponding mapping $K^* : A^* \to B^*$ is not. K is not an uniquely decodable encoding.

4.3 Prefix encoding and Kraft's inequality

Definition 4.2. The **prefix of the word b** $= b_1b_2 \dots b_k$ is every word

$$b_1, \quad b_1b_2, \quad \dots, \quad b_1b_2 \dots b_{k-1}, \quad b_1b_2 \dots b_k.$$

An encoding resp., a code is called **prefix encoding**, resp., **prefix code**, if no code word is a prefix of another code word.

Remark. Note that every block encoding is a prefix encoding.

Example 4.4. The set of telephone numbers of telephone subscribers is an example of a prefix code which is not a block code. Ambulance service has the number 155 and there is no other telephone number starting with 155. The numbers of regular subscribers are longer. A number of a particular telephone station is never identical to a prefix of a different station, otherwise the station with prefix number would always accept a call during the process of dialing the longer telephone number.

The prefix encoding is the only encoding decodable character by character, i. e., in the process of receiving a message (and we do not need to wait for the end of the message). Decoding of received message is as follows:
Find the least number of characters of the message (from the left) creating a code word $K(a)$ which corresponds to the source character a. Decode this word as a, discard the word $K(a)$ from the message and continue by the same way till the end of the received message.

Theorem 4.1. Kraft's inequality. *Let $A = \{a_1, a_2, \dots, a_r\}$ be a source alphabet with r characters, let $B = \{b_1, b_2, \dots, b_n\}$ be code alphabet with n*

characters. A prefix code with code word lengths $d_1, d_2, \dots, d_r$ *exists if and only if*

$$n^{-d_1} + n^{-d_2} + \dots + n^{-d_r} \le 1. \tag{4.2}$$

Inequality (4.2) *is called* **Kraft's inequality***.*

Proof. Suppose that the Kraft's inequality holds (4.2). Sort the numbers d_i such that $d_1 \le d_2 \le \dots \le d_r$. Set $K(a_1)$ to an arbitrary word of alphabet B of the length d_1. Now we will proceed by mathematical induction.
Suppose that $K(a_1)$, $K(a_2)$, $\dots$, $K(a_i)$ are code words of the lengths d_1, d_2, $\dots$, d_i. When choosing a code word $K(a_{i+1})$ of the length d_{i+1} we have to avoid using $n^{(d_{i+1}-d_1)}$ words of the length d_{i+1} with prefix $K(a_1)$, $n^{(d_{i+1}-d_2)}$ words of the length d_{i+1} with prefix $K(a_2)$ etc. till $n^{(d_{i+1}-d_i)}$ words of the length d_{i+1} with prefix $K(a_i)$, whereas the number of all words of the length d_{i+1} is $n^{d_{i+1}}$. The number of forbidden words is

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \dots + n^{(d_{i+1}-d_i)}. \tag{4.3}$$

Since (4.2) holds, it holds also for the first $i+1$ items of the left side of (4.2):

$$n^{-d_1} + n^{-d_2} + \dots + n^{-d_i} + n^{-d_{i+1}} \le 1. \tag{4.4}$$

After multiplying both sides of (4.4) by $n^{d_{i+1}}$ we get:

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \dots + n^{(d_{i+1}-d_i)} + 1 \le n^{d_{i+1}}. \tag{4.5}$$

By (4.5) the number of forbidden words is less at least by 1 than the number of all words of the length d_{i+1} – there is at least one word of the length d_{i+1} which is not forbidden. Therefore, we can define this word as the code word $K(a_{i+1})$.

Now suppose that we have a prefix code with code word lengths $d_1, d_2, \dots, d_r$, let $d_1 \le d_2 \le \dots \le d_r$. There exist n^{d_r} words of the length d_r, one of them is used as $K(a_r)$. For every $i = 1, 2, \dots, r-1$ the word $K(a_i)$ is a prefix of $n^{(d_r-d_i)}$ words of the length d_r (forbidden words) – these words are different from $K(a_r)$ (otherwise the code is not prefix code). Since $K(a_r)$ is different from all forbidden words, it has to hold:

$$n^{(d_r-d_1)} + n^{(d_r-d_2)} + \dots + n^{(d_r-d_{r-1})} + 1 \le n^{d_r}\ . \tag{4.6}$$

After dividing both sides of (4.6) by n^{d_r} we get the required Kraft's inequality (4.2). ■

Remark. **Algorithm for construction of prefix code with given code word lengths $d_1, d_2, \dots, d_r$.** The first part of the proof of the theorem 4.1 is constructive – it gives directions how to construct a prefix code provided that the code word lengths $d_1 \le d_2 \le \dots \le d_r$ fulfilling Kraft's inequality are given. Choose an arbitrary word of the length d_1 as $K(a_1)$. Having assigned $K(a_1)$, $K(a_2)$, $\dots$, $K(a_i)$, for $K(a_{i+1})$ choose an arbitrary word w of the length d_{i+1} for which no of words $K(a_1)$, $K(a_2)$, $\dots$, $K(a_i)$ is a prefix of w. The existence of such a word w is guaranteed by Kraft's inequality.

Theorem 4.2. Mac Millan.
Kraft's inequality (4.2) *holds for every uniquely decodable encoding with source alphabet $A = \{a_1, a_2, \dots, a_r\}$ and code alphabet $B = \{b_1, b_2, \dots, b_n\}$ with code word lengths $d_1, d_2, \dots, d_r$.*

Proof. Let K be a uniquely decodable encoding with code word lengths $d_1 \le d_2 \le \dots \le d_r$. Denote

$$c = n^{-d_1} + n^{-d_2} + \dots + n^{-d_r} \ . \qquad (4.7)$$

Our plan is to show that $c \le 1$.
Let k be an arbitrary natural number. Let $\mathcal{M}_k$ be the set of all words of code alphabet of the type $\mathbf{b} = K(a_{i_1})|K(a_{i_2})|\dots|K(a_{i_k})$. The length of such word $\mathbf{b}$ is $d_{i_1} + d_{i_2} + \dots + d_{i_k}$ and it is less or equal to $k.d_r$ since maximum of code word lengths is d_r.
Let us study the following expression:

$$c^k = \left[n^{-d_1} + n^{-d_2} + \dots + n^{-d_r}\right]^k = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \dots \sum_{i_k=1}^{n} n^{-(d_{i_1}+d_{i_2}+\dots+d_{i_k})} \ . \qquad (4.8)$$

Since K is uniquely decodable it holds for two different words $a_{i_1}a_{i_2}\dots a_{i_k}$, $a'_{i_1}a'_{i_2}\dots a'_{i_k}$ of source alphabet A

$$K(a_{i_1})|K(a_{i_2})|\dots|K(a_{i_k}) \ne K(a'_{i_1})|K(a'_{i_2})|\dots|K(a'_{i_k}) \ .$$

Therefore we can assign to every word $\mathbf{b} = K(a_{i_1})|K(a_{i_2})|\dots|K(a_{i_k})$ from $\mathcal{M}_k$ exactly one summand $n^{-(d_{i_1}+d_{i_2}+\dots+d_{i_k})}$ on the left side of (4.8) such that its exponent multiplied by -1 $(d_{i_1} + d_{i_2} + \dots + d_{i_k})$ equals to the length of the word $\mathbf{b}$.

As we have shown the maximum of word lengths from the set $\mathcal{M}_k$ is kd_r. Denote $M = kd_r$. The expression on the right side of (4.8) is a polynomial of degree M of variable $\dfrac{1}{n}$. Therefore we can write it in the form:

$$c^k = s_1.n^{-1} + s_2.n^{-2} + \cdots + s_M.n^{-M} = \sum_{i=1}^{M} s_i.n^{-i}.$$

The item n^{-i} occurs in the sum on the right side of the last equation exactly as many times as how many words from the set $\mathcal{M}_k$ have the length i. Since the code alphabet has n characters, at most n^i words from $\mathcal{M}_k$ can have the length i. Therefore we can write:

$$\begin{aligned} c^k &= s_1.n^{-1} + s_2.n^{-2} + \cdots + s_M.n^{-M} \le \\ &\le n^1.n^{-1} + n^2.n^{-2} + \cdots + n^M.n^{-M} \le 1 + 1 + \cdots + 1 = M = k.d_r \end{aligned} \tag{4.9}$$

and hence

$$\frac{c^k}{k} \le d_r \ . \tag{4.10}$$

The inequality (4.10) has to hold for arbitrary k which implies that $c \le 1$. ■

The corollary of Mac Millan theorem is that no uniquely decodable encoding has shorter code word lengths than the prefix encoding. Since the prefix encoding has a lot of advantages, e. g. simple decoding character by character, it suffices to restrict ourselves to the prefix encodings.

4.4 Shortest code - Huffman's construction

Definition 4.3. Let $\mathcal{Z} = (A^*, P)$ be a source transmitting characters of source alphabet $A = \{a_1, a_2, \dots, a_r\}$ with probabilities $p_1, p_2, \dots, p_r$, $\sum_{i=1}^{r} p_i = 1$. Let K be a prefix encoding with code word lengths $d_1, d_2, \dots, d_r$. Then the **mean code word length** of encoding K is

$$d(K) = p_1.d_1 + p_2.d_2 + \cdots + p_r.d_r = \sum_{i=1}^{r} p_i.d_i \ . \tag{4.11}$$

Let us have a message $\mathbf{m}$ from the source $\mathcal{Z}$ containing a large number N of characters. We can expect that the length of encoded message $\mathbf{m}$ will be approximately $N.d(K)$. Very often we require that the encoded message is as short as possible. That is why we are searching for an encoding with minimum mean code word length.

Definition 4.4. Let $A = \{a_1, a_2, \dots, a_r\}$ be a source alphabet with probabilities of characters $p_1, p_2, \dots, p_r$, let $B = \{b_1, b_2, \dots, b_n\}$ be a code alphabet. The **shortest** n**-ary encoding** of alphabet A is such encoding $K : A \to B^*$ which has the least mean code word length $d(K)$. The corresponding code is called the **shortest** n**-nary code**.

The shortest prefix code was constructed by O. Huffman in 1952. We will study namely binary codes – codes with code alphabet $B = \{0, 1\}$ – which are most important in practice.

4.5 Huffman's Algorithms

Let $A = \{a_1, a_2, \dots, a_r\}$ be a source alphabet, let $p_1, p_2, \dots, p_r$ are the probabilities of characters $a_1, a_2, \dots, a_r$, suppose $p_1 \geq p_2 \geq \dots \geq p_r$. Let $B = \{0, 1\}$ be the code alphabet. Our goal is to find the shortest binary coding of alphabet A.

We will create step by step a binary rooted tree whose leaf vertices are $a_1, a_2, \dots, a_r$. Every node v of the tree has two labels: the probability $p(v)$ and the character $ch(v) \in B \cup \{\texttt{UNDEFINED}\}$.

Step 1: Initialization: Create a graph $G = (V, E, p, ch)$, with vertex set $V = A$, edge set $E = \emptyset$, $p : V \to \langle 0, 1 \rangle$, where $p(a_i) = p_i$ is the probability of character a_i and $ch(v) = \texttt{UNDEFINED}$ for all $v \in V$. A vertex $v \in V$ with $ch(v) = \texttt{UNDEFINED}$ is called **unlabeled**.

Step 2: Find two unlabeled u, $w \in V$ with two least probabilities $p(u)$, $p(w)$. Set $ch(u) = 0$, $ch(w) = 1$. Extend the vertex set V by a new vertex x, i. e., set $V := V \cup \{x\}$ for some $x \notin V$, set $p(x) := p(u) + p(w)$, $ch(x) = \texttt{UNDEFINED}$, and $E := E \cup \{(x, u), (x, w)\}$ (make x a parent of both u, w).

Step 3: If G is a connected graph, GO TO Step 4, otherwise continue by Step 2.

Step 4: At this moment G is a rooted tree with leaf vertices corresponding to the characters of the source alphabet A. All vertices of the tree G expect the root are labeled by binary labels 0 or 1. There is a unique path from the root of the tree G to every character $a_i \in A$. The sequence of labels $ch(\ )$ along this path defines the code word assigned to character a_i.

The construction of n-ary Huffman's code is analogical. Suppose the alphabet A has $r = n + k.(n - 1)$ characters (otherwise we can add several dummy characters with zero probabilities – their code words remain unused). Find n characters of source alphabet with the least probabilities and assign them the characters of code alphabet in arbitrary order (these will be the last characters of the corresponding code words). Reduce the alphabet A such that instead of n characters with the least probabilities we add one fictive character with the total probability of replaced characters. The reduced alphabet has $n+(k-1).(n-1)$ characters. If $k-1>0$ we repeat this procedure, etc.

4.6 Source Entropy and the Length of the Shortest Code

The entropy of a source $\mathcal{Z}=(Z^*,P)$ was defined by definition 3.5 (page 57) as

$$H(\mathcal{Z})=-\lim_{n\to\infty}\frac{1}{n}\ .\sum_{(x_1,\dots,x_n)\in Z} P(x_1,x_2,\dots,x_n).\log_2 P(x_1,x_2,\dots,x_n)\ . \tag{4.12}$$

For a stationary independent source $\mathcal{Z}=(A^*,P)$ with alphabet $A=\{a_1,a_2,\dots,a_r\}$ and with character probabilities $p_1,p_2,\dots,p_r$ it was shown (theorem 3.1, page 57) that

$$H(\mathcal{Z})=-\sum_{i=1}^{r}p_1.\log_2(p_i)\ .$$

Let K be an arbitrary prefix encoding of alphabet A with code word length $d_1,d_2,\dots,d_r$ and with mean code word length $d=d(K)$. We want to find out the relation between the entropy $H(\mathcal{Z})$ and $d(K)$. The simplest case is that of stationary independent source.
We can write step by step:

$$\begin{aligned}
H(\mathcal{Z})-d &= \sum_{i=1}^{r}p_i.\log_2\left(\frac{1}{p_i}\right)-\sum_{i=1}^{r}p_i.d_i=\sum_{i=1}^{r}p_i.\left[\log_2\left(\frac{1}{p_i}\right)-d_i\right]=\\
&= \sum_{i=1}^{r}p_i.\left[\log_2\left(\frac{1}{p_i}\right)+\log_2\left(2^{-d_i}\right)\right]=\sum_{i=1}^{r}p_i.\left[\log_2\left(\frac{2^{-d_i}}{p_i}\right)\right]=\\
&= \frac{1}{\ln 2}.\sum_{i=1}^{r}p_i.\left[\ln\left(\frac{2^{-d_i}}{p_i}\right)\right]\leq
\end{aligned}$$

$$
\begin{aligned}
&\le \frac{1}{\ln 2}.\sum_{i=1}^{r} p_i.\left(\frac{2^{-d_i}}{p_i}-1\right) = \frac{1}{\ln 2}.\left[\sum_{i=1}^{r} 2^{-d_i} - \sum_{i=1}^{r} p_i\right] = \\
&= \frac{1}{\ln 2}.\left[\sum_{i=1}^{r} 2^{-d_i} - 1\right] \le 0\ .
\end{aligned}
$$

The first inequality follows from well known inequality $\ln(x) \le x-1$ applied to $x = 2^{-d_i}/p_i$, the second one holds since natural numbers d_i are the lengths of code words of a prefix code, and Kraft's inequality $\sum_{i=1}^{r} 2^{-d_i} \le 1$ holds.
Hence

$$H(\mathcal{Z}) \le d(K) \tag{4.13}$$

holds for an arbitrary prefix encoding (and also for uniquely decodable).

Let d_i for $i = 1, 2, \dots, r$ are natural numbers such that

$$-\log_2(p_i) \le d_i < -\log_2(p_i) + 1$$

for every $i = 1, 2, \dots, r$. Then the first inequality can be rewritten as follows:

$$\log_2\left(\frac{1}{p_i}\right) \le d_i \quad \Rightarrow \quad \frac{1}{p_i} \le 2^{d_i} \quad \Rightarrow \quad 2^{-d_i} \le p_i\ .$$

The last inequality holds for every i, therefore we can write:

$$\sum_{i=1}^{r} 2^{-d_i} \le \sum_{i=1}^{r} p_i \le 1\ .$$

The integers d_i for $i = 1, 2, \dots, r$ fulfill Kraft's inequality and that is why there exists a binary prefix encoding with code word lengths $d_1, d_2, \dots, d_r$. The mean code word length of this encoding is:

$$d = \sum_{i=1}^{r} p_i.d_i < -\sum_{i=1}^{r} p_i.\big[\log_2(p_i)+1\big] = -\sum_{i=1}^{r} p_i.\log_2(p_i) + \sum_{i=1}^{r} p_i = H(\mathcal{Z}) + 1.$$

We have proved that there exists a prefix binary encoding K of alphabet A for which it holds:

$$d(K) < H(\mathcal{Z}) + 1. \tag{4.14}$$

Corollary: Let d_{opt} be the length of the shortest prefix binary encoding of alphabet A. Then

$$d_{\text{opt}} < H(\mathcal{Z}) + 1. \tag{4.15}$$

Just proved facts are summarized in the following theorem:

Theorem 4.3. *Let $\mathcal{Z} = (A^*, P)$ be a stationary independent source with entropy $H(\mathcal{Z})$, let d_{opt} is the mean code word length of the shortest binary prefix encoding of A. Then it holds:*

$$H(\mathcal{Z}) \leq d_{\text{opt}} < H(\mathcal{Z}) + 1. \tag{4.16}$$

Example 4.5. Suppose that $\mathcal{Z} = (A^*, P)$ is a source with the source alphabet $A = \{x, y, z\}$ having three characters with probabilities $p_x = 0.8$, $p_y = 0.1$, $p_z = 0.1$. Encoding $K(x) = 0$, $K(y) = 10$, $K(z) = 11$ is the shortest binary prefix encoding of A with the length $d(K) = 1 \times 0.8 + 2 \times 0.1 + 2 \times 0.1 = 1.2$. The entropy of $\mathcal{Z}$ is $H(\mathcal{Z}) = 0.922$ bits per character. Given a source message with length N, the length of the corresponding binary encoded text is approximately $N \times 1.2$, and its lower bound is equal to $N \times 0.922$ by theorem 4.3. A long encoded text will be 30% longer than the lower bound determined by entropy $H(\mathcal{Z})$.

It is possible to find more visible examples of percentage difference between the lower bound determined by entropy and the length of the shortest binary prefix encoding (try $p_x = 0.98$, $p_y = 0.01$, $p_z = 0.01$). Since no uniquely decodable binary encoding of source alphabet A can have a less mean code word length, this example does not offer too much optimism about usefulness of the lower bound from theorem 4.3.

However, the encoding character by character is not the only possible way how to encode the source text. In section 3.4 (page 62), in definition 3.8, for every source $\mathcal{Z}$ with entropy $H(\mathcal{Z})$ we defined the source $\mathcal{Z}_{(k)}$ with entropy $k.H(\mathcal{Z})$. The source alphabet of $\mathcal{Z}_{(k)}$ is the set of all k-character words of alphabet A. Provided that $\mathcal{Z}$ is a stationary independent source, the source $\mathcal{Z}_{(k)}$ is a stationary independent source, too. For the mean code word length $d_{\text{opt}}^{(k)}$ of the shortest binary prefix encoding of alphabet A^k the inequalities (4.16) from theorem 4.3 are in the form:

$$\begin{aligned}
H(\mathcal{Z}_{(k)}) \leq & d_{\text{opt}}^{(k)} < H(\mathcal{Z}_{(k)}) + 1 \\
k.H(\mathcal{Z}) \leq & d_{\text{opt}}^{(k)} < k.H(\mathcal{Z}) + 1 \\
H(\mathcal{Z}) \leq & \frac{d_{\text{opt}}^{(k)}}{k} < H(\mathcal{Z}) + \frac{1}{k}
\end{aligned} \tag{4.17}$$

These facts are formulated in the following theorem:

Theorem 4.4. Fundamental theorem on source coding. *Let $\mathcal{Z} = (A^*, P)$ be a stationary independent source with entropy $H(\mathcal{Z})$. Then the mean code word length of binary encoded text per one character of source alphabet A is bounded from below by entropy $H(\mathcal{Z})$. Moreover, it is possible to find an integer $k > 0$ and a binary prefix encoding of words from A^k such that the mean code word length per one character of source alphabet is arbitrarily near to the entropy $H(\mathcal{Z})$.*

The fundamental theorem on source coding holds also for more general stationary sources $\mathcal{Z}$ (the proof is more complicated). The importance of this theorem is in the fact that the source entropy is the limit value of the average length per one source character of optimally binary encoded source text.

Here we can see that the notion of source entropy was suitably and purposefully defined and has its deep meaning. Remember that the entropy $H(\mathcal{Z})$ stands in the formula (4.16) without any conversion coefficient (resp. with coefficient 1) which is the consequence of felicitous choosing the number 2 for the basis of logarithm in Shannon's formula of information and Shannon – Hartley formula for entropy.

As we have shown natural language cannot be considered to be an independent source, its entropy is much less than the entropy of the first character $H_1 = -\sum_i p_i \log_2(p_i)$. Here, the principle of fundamental source coding theorem can be applied – in order to obtain a shorter encoded message, we have to encode words of source text instead of single characters. Here described principles are the fundamentals for many compression methods.

4.7 Error detecting codes

In this section, we will study natural n-ary block codes with a code alphabet having n characters. This codes are models for real situation. In the place of the code alphabet in most cases the set of computer keyboard characters, or decimal characters $0 - 9$, or any other finite set of symbols can be used.

Human factor is often present in processing natural texts or numbers, and it is the source of many errors. Our next problem is how to design a code capable to find out that a single error, (or at most a given number of errors) have occurred after transmission.

We have several data from Anglo Saxon literature about percentage of errors arising by typing texts and numbers on computer keyboard.

- Simple typing error $a \to b$ 79%
- Neighbour transposition $ab \to ba$ 10.2%
- Jump transposition $abc \to cba$ 0.8%
- Twin error $aa \to bb$ 0.6%
- Phonetic error $X0 \to 1X$ 0.5%
- Other errors 8.9%

We can see that the two most frequent human errors are the simple error and the neighbour transposition.

The phonetic error is probably an English speciality, and the cause of it is probably the little difference between English numerals (e. g., fourteen – forty, fifteen – fifty etc.,).

The reader can wonder why drop character or add character errors are not mentioned. The answer is that we are studying block codes, and the two just mentioned errors change the word length so that they are immediately visible.

If the code alphabet B has n characters then the number of all words of B of the length k is n^k – this is the largest possible number of code words of n-ary block code of the length k. The only way how to detect an error in a received message is following: To use only a part of all n^k possible words for the code words, the others are claimed as non code words. If the received word is a non code word we know that an error occurred in the received word.

Two problems can arise when designing such a code. The first one is how to choose the set of words in order to ensure that a single error (or at most specified number of errors) makes a non code word from arbitrary code word. The second problem is how to find out quickly whether the received word is a code word or a non code word.

First, we restrict ourselves to typing errors. It proves useful to introduce a function expressing the difference between a pair of arbitrary words on the set $B^n \times B^n$ of all ordered pairs of words.

We would like that this function has properties similar to the properties of the distance between points in a plane or a space.

Definition 4.5. A real function d defined on Cartesian product $V \times V$ is called **metric on** V, if it holds:

1. For every $u, v \in V$ $d(u,v) \geq 0$ with equality if and only if $u = v$. (4.18)
2. For every $u, v \in V$ $d(u,v) = d(v,u)$. (4.19)
3. If $u, v, w \in V$, then $d(u,w) \leq d(u,v) + d(v,w)$. (4.20)

The inequality (4.20) is called **triangle inequality**.

Definition 4.6. The **Hamming distance** $d(\mathbf{v},\mathbf{w})$ of two words $\mathbf{v} = v_1v_2\dots v_n$, $\mathbf{w} = w_1w_1\dots w_n$ is the number of places in which $\mathbf{v}$ and $\mathbf{w}$ differs, i. e.,

$$d(\mathbf{v},\mathbf{w}) = \Big|\{i \mid v_i \neq w_i, \quad i = 1, 2, \dots, n\}\Big|.$$

It is easy to show that the Hamming distance has all properties of metric, that is why it is sometimes called also **Hamming metric**.

Definition 4.7. The **minimum distance** $\Delta(\mathcal{K})$ **of a block code** (K) is the minimum of distances of all pairs of different code words from $\mathcal{K}$.

$$\Delta(\mathcal{K}) = \min\{d(\mathbf{a},\mathbf{b}) \mid \mathbf{a},\mathbf{b} \in \mathcal{K},\ \mathbf{a} \neq \mathbf{b}\}. \tag{4.21}$$

We say that the code $\mathcal{K}$ **detects t-tuple simple errors** if for every code word $\mathbf{u} \in \mathcal{K}$ and every word $\mathbf{w}$ such that $0 < d(\mathbf{u},\mathbf{w}) \leq t$ the word $\mathbf{w}$ is a non code word.

We say that we have detected an error after receiving a non code word. Pleas note that a block code $\mathcal{K}$ with the minimum distance $\Delta(\mathcal{K}) = d$ detects $(d-1)$-tuple simple errors.

Example 4.6 (Two-out-of-five code)**.** The number of ways how to choose two elements out of five ones is $\binom{5}{2} = 10$. This fact can be used for encoding decimal digits. This code was used by US Telecommunication, another system by U.S. Post Office. The IBM 7070, IBM 7072, and IBM 7074 computers used this code to represent each of the ten decimal digits in a machine word.

Several two-out-of-five code systems

Digit	Telecommunication	IBM	POSTNET
	01236	01236	74210
0	01100	01100	11000
1	11000	11000	00011
2	10100	10100	00101
3	10010	10010	00110
4	01010	01010	01001
5	00110	00110	01010
6	10001	10001	01100
7	01001	01001	10001
8	00101	00101	10010
9	00011	00011	10100

The decoding can be made easily by adding weights[1] (in the table the second row from above) corresponding to code word characters 1 except source digit 0.

Two-out-of-five code detects one simple error – when changing arbitrary 0 to 1 the result is the word with three characters 1, changing 1 to 0 leads to the word with only one 1 – both resulting words are non code words. However, the Hamming distance of code words 11000 and 10100 is equal to 2 which implies that the two-out-of-five code cannot detect all 2-tuple simple errors.

Example 4.7. 8-bit even-parity code is an 8-bit code where the first 7 bits create an arbitrary 7-bit code (with $2^7 = 128$ code words) and where the last bit is added such that the number of ones in every code word is even. The even-parity code detects one simple error, its minimal distance is 2. The principle of parity bit was frequently used by transmissions and in some applications is used till now.

Example 4.8. Doubling code. The doubling code is a block code of even length in which every character stands in every code word twice. The binary doubling code of the length 6 has 8 code words:

000000 000011 001100 001111 110000 110011 111100 111111

The minimum distance of doubling code is 2, it detects one simple error.

[1] The weights for IBM are 01236.
The decoding of code word 00011 is $0.0 + 1.0 + 2.0 + 3.1 + 6.1 = 9$.

Example 4.9. Repeating code. The principle of repeating code is several-fold repeating of the same character. Codewords are only the words with all characters equal – e. g. 11111, 22222, . . . , 99999, 00000. The minimum distance of the repeating code $\mathcal{K}$ of the length is $\Delta\mathcal{K} = n$ and that is why it detects $(n-1)$-tuple simple errors. Note that we are able to restore a transmitted word provided that we have a repeating code of the length 5 and that at most 2 errors occurred. After receiving 10191, we know that the word 11111 was transmitted assuming that at most two errors occurred.

Example 4.10. UIC railway car number is a unique 12 digit number for each car containing various information about the car[2] in the form

$$X\ X\ XX\ X\ XXX\ XXX\ X$$

The last digit is the check digit.
Let us have a railway car number

$$a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}$$

The check digit a_{12} is calculated so that the sum of all digits

$$2a_1\ a_2\ 2a_3\ a_4\ 2a_5\ a_6\ 2a_7\ a_8\ 2a_9\ a_{10}\ 2a_{11}\ a_{12}$$

is divisible by 10. By another words: Multiply the digits 1 to 11 alternately by 2 and 1 and add the digits of the results. Subtract the last digit of the resulting number from 10 and take the last digit of what comes out: this is the check digit.

The digits on odd and even positions are processed differently – the designers evidently made efforts to detect at least some of neighbour transposition errors.

Let C, D are the two neighbouring digits, let C be on an odd position. Denote $\delta(Y)$ the sum of digits of the number $2Y$ for $Y = 0, 1, \ldots, 9$. Then

$$\delta(Y) = \begin{cases} 2Y & \text{if } Y \leq 4 \\ 2Y - 9 & \text{if } Y > 4 \end{cases}$$

For what values of digits C, D the check digit remains unchanged after their neighbour transposition?

[2] The specification of the meaning can be found at unofficial source `http://www.railfaneurope.net/misc/uicnum.html`.

The sum $\delta(C) + D$ has to give the same remainder by integer division by 10 as $\delta(D) + C$ in order to retain the check digit unchanged. Therefore, their difference has to be divisible by 10.

$$\delta(C) + D - \delta(D) - C =$$

$$= \begin{cases} 2C + D - 2D - C = C - D & \text{if } C \leq 4 \text{ and } D \leq 4 \\ 2C - 9 + D - 2D - C = C - D - 9 & \text{if } C \geq 5 \text{ and } D \leq 4 \\ 2C + D - 2D + 9 - C = C - D + 9 & \text{if } C \leq 4 \text{ and } D \geq 5 \\ 2C + 9 + D - 2D - 9 - C = C - D & \text{if } C \geq 5 \text{ and } D \geq 5 \end{cases}$$

In the first and in the fourth case, the difference $C - D$ is divisible by 10 if and only if $C = D$, which implies, that the code can detect the neighbour transposition error of every pair of digits provided both are less than 5, or both are greater than 4.
In the second case, if $C \geq 5$ and $D \leq 4$ then $1 \leq (C - D) \leq 9$. The expression $\delta(C) + D - \delta(D) - C$ equals to $(C - D) - 9$ in this case. The last expression is divisible by 10 if and only if $C - D = 9$, what can happen only for $C = 9$ and $D = 0$.
In the third case, if $C \leq 4$ and $D \geq 5$ then $0 - 9 = -9 \leq (C - D) \leq 4 - 5 = -1$. The expression $\delta(C) + D - \delta(D) - C$ equals to $(C - D) + 9$ in this case. The last expression is divisible by 10 only if $C - D = -9$, from what it follows $C = 0$ and $D = 9$. We see can that the equation

$$\delta(C) + D - \delta(D) - C \equiv 0 \mod 10$$

has only two solutions, namely $(C, D) = (0, 9)$ and $(C, D) = (9, 0)$.

The UIC railway car number detects one simple error or one neighbour transposition provided that the transposed pair is different from $(0, 9)$ and $(9, 0)$. The designers did not succeed in constructing a code detecting all neighbour transpositions.

4.8 Elementary error detection methods

This and the the section 4.9 will be devoted to error detecting methods in natural decimal block codes of the length n. The code alphabet of these codes is the set $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$. The principle of these methods is that the first $n - 1$ digits of code word $\mathbf{a} = a_1 a_2 \dots a_{n-1} a_n$ can be arbitrary $(n - 1)$-tuple of digits (they are intended to carry information) and the last digit a_n is so called **check digit** satisfying so called **check equation**:

$$f(a_1, a_2, \dots, a_n) = c \ , \tag{4.22}$$

where f is an appropriate function. We will search for such a function f for which it holds:
If the word $\mathbf{a}' = a'_1 a'_2 \dots a'_{n-1} a'_n$ originated from the word $\mathbf{a} = a_1 a_2 \dots a_{n-1} a_n$ by one simple error or one neighbour transposition then $f(\mathbf{a}) \neq f(\mathbf{a}')$.

4.8.1 Codes with check equation mod 10

The check number a_n for decimal codes is calculated from the equation:

$$a_n \equiv -\sum_{i=1}^{n-1} w_i.a_i \mod 10 \ ,$$

where w_i are fixed preselected numbers, $0 \leq w_i \leq 9$. This approach can be slightly generalized that the code words are words $\mathbf{a} = a_1 a_2 \dots a_{n-1} a_n$ satisfying the following check equation:

$$\sum_{i=1}^{n} w_i.a_i \equiv c \mod 10 \ . \tag{4.23}$$

After replacing the digit a_j by a'_j in the code word $\mathbf{a} = a_1 a_2 \dots a_{n-1} a_n$ the left side of check equation 4.23 will be equal to

$$\sum_{i=1}^{n} w_i.a_i + w_j.a'_j - w_j.a_j \equiv c + w_j.(a'_j - a_j) \mod 10 \ .$$

The right side of equation 4.23 remains unchanged and the corresponding code cannot detect this simple error if

$$w_j.(a'_j - a_j) \equiv 0 \mod 10 \ .$$

The last equation has unique solution $a'_j = a_j$ if and only if w_j and 10 are relatively prime. Coefficients w_i can be equal to one of numbers 1, 3, 7 and 9.

Try to find out whether the code with check equation (4.23) can detect neighbour transpositions. The code cannot detect the neighbour transposition of digits $x,\ y$ on places $i,\ i+1$ if and only if

$$w_i.y + w_{i+1}.x - w_i.x - w_{i+1}.y \equiv 0 \mod 10$$

$$\begin{aligned} w_i.(y-x) - w_{i+1}.(y-x) &\equiv 0 \mod 10 \\ (w_i - w_{i+1})(y-x) &\equiv 0 \mod 10 \end{aligned}$$

It is necessary for detection of neighbour transposition of x and y that the last equation has the only solution $x = y$. This happens if and only if the numbers $(w_i - w_{i+1})$ and 10 are relatively prime. But, as we have shown above, the coefficients w_i and w_{i+1} have to be elements of the set $\{1, 3, 7, 9\}$ and that is why $(w_i - w_{i+1})$ is always even.

Theorem 4.5. *Let $\mathcal{K}$ be a decimal block code of the length n with check equation* (4.23). *The code $\mathcal{K}$ detects all single simple errors if and only if all w_i are relatively prime to* 10, *i. e., if $w_i \in \{1, 3, 7, 9\}$. No decimal block code of the length n with check equation* (4.23) *detects all single simple error and at the same time all single neighbour transpositions.*

Example 4.11. EAN European Article Number is a 13 digit decimal number used worldwide for unique marking of retail goods. EAN-13 code is placed as a bar code on packages of goods. It allows scanning by optical scanners and thus reduces the amount of work with stock recording, billing and further manipulation with goods.

First 12 digits $a_1, \dots, a_{12}$ of EAN code carry information, the digit a_{13} is the check digit fulfilling the equation:

$$a_{13} \equiv -(1.a_1 + 3.a_2 + 1.a_3 + 3.a_4 + \cdots + 1.a_{11} + 3.a_{12}) \mod 10\ .$$

EAN code detects one simple error. The EAN code cannot detect the neighbour transposition for a pair x, y subsequent digits if

$$\begin{aligned} (x + 3y) - (3x + y) &\equiv 0 \mod 10 \\ (2y - 2x) &\equiv 0 \mod 10 \\ 2.(y - x) &\equiv 0 \mod 10 \end{aligned}$$

The last equation has the following solutions (x, y):

$$\begin{gathered} (0,0),\ (0,5),\ (1,1),\ (1,6),\ (2,2),\ (2,7),\ (3,3),\ (3,8),\ (4,4),\ (4,9), \\ (5,5),\ (5,0),\ (6,6),\ (6,1),\ (7,7),\ (7,2),\ (8,8),\ (8,3),\ (9,9),\ (9,4) \end{gathered}$$

The EAN code cannot detect the neighbour transposition for the following ten ordered pairs of digits:

$$(0,5),\ (1,6),\ (2,7),\ (3,8),\ (4,9),$$

$$(5,0),\ (6,1),\ (7,2),\ (8,3),\ (9,4)$$

EAN code with 10 undetectable instances of neighbour transpositions is much worse then UIC railway car number which cannot detect only two neighbour transpositions of pairs $(0,9)$ and $(9,0)$.

4.8.2 Checking mod 11

These codes work with the code alphabet $B \cup \{X\}$ where $B = \{1,2,3,4,5,6,7,8,9,0\}$ and where the digit X expresses the number 10. Every code word $\mathbf{a} = a_1a_2\dots a_{n-1}a_n$ of the length n has the first $n-1$ digits the elements of the alphabet B, and the last digit $a_{n-1} \in B \cup \{X\}$ is calculated from the equation:

$$\sum_{i=1}^{n} w_i.a_i \equiv c \mod 11\ , \quad \text{where } 0 < w_i \le 10 \text{ for } i = 1,2,\dots,n\ . \tag{4.24}$$

Similarly, as in the case of check equation mod 10, we show that the code with checking mod 11 detects one simple error if and only if the equation

$$w_j.(a'_j - a_j) \equiv 0 \mod 11$$

has the only solution $a'_j = a_j$ and this happens if and only if w_j and 11 are relatively prime – it suffices that $w_j \neq 0$.
The code with checking mod 11 detects all neighbour transpositions on word positions i, $i+1$ if and only if the equation

$$(w_i - w_{i+1}).(y - x) \equiv 0 \mod 11$$

is fulfilled only by such pairs of digits (x, y) for which $x = y$.
In conclusion, let us remark that simple error detecting property and transposition error detecting property will not be lost if we allow all characters of code words from alphabet $B \cup \{X\}$.

Example 4.12. ISBN code – The International Standard Book Number is a 10 digit number assigned to every officially issued book. The first four digits $a_1a_2a_3a_4$ of ISBN number define the country and the publishing company, the following five digits $a_5a_6a_7a_8a_9$ specify the number of the book in the frame of its publisher and the last digit a_{10} is the check digit defined by the equation:

$$a_{10} \equiv \sum_{i=1}^{9} i.a_i \mod 11\ .$$

The characters a_1 till a_9 are decimal digits – elements of alphabet $B = \{0,1,\dots,9\}$, the character a_{10} is element of alphabet $A \cup \{X\}$ where X represents the value 10.
The last equation is equivalent with equation

$$\sum_{i=1}^{10} i.a_i \equiv 0 \mod 11 \ ,$$

since $-a_{10} \equiv -a_{10} + 11.a_{10} \equiv 10.a_{10} \mod 11$. If $a_{10} = 10$, the character X is printed on the place of check digit. This is a disadvantage because the alphabet of ISBN code has in fact 11 elements but the character X is used only seldom. ISBN code detects all single simple errors and all single neighbour transpositions.

Definition 4.8. The **geometric code mod 11** is a block code of the length n with characters from alphabet $B \cup \{X\}$ with check equation (4.24) in which

$$w_i = 2^i \mod 11 \quad \text{for } i = 1, 2, \dots, n \ .$$

Example 4.13. Bank account numbers of Slovak banks. The bank account number is a 10 digit decimal number

$$a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9.$$

The meaning of single positions is not specified. A valid account number has to fulfill the check equation:

$$0 = \left(\sum_{i=0}^{9} 2^i.a_i\right) \mod 11 = (1.a_0{+}2.a_1{+}4.a_2{+}8.a_3{+}\cdots{+}512.a_9) \mod 11 =$$

$$= (a_0 + 2a_1 + 4a_2 + 8a_3 + 5a_4 + 10a_5 + 9a_6 + 7a_7 + 3a_8 + 6a_9) \mod 11 \ .$$

We can see that the geometrical code mod 11 is used here. In order to avoid the cases when $a_9 = 10$ simply leave out the numbers $a_0a_1\dots a_8$ leading to check digit $a_9 = 10$.
The bank account number code detects all single simple errors, all single neighbour transpositions, but moreover all single transpositions on arbitrary positions of bank account number.

Example 4.14. Slovak personal identification number. The internet site `www.minv.sk/vediet/rc.html` specifies Slovak personal identification numbers. The personal identification number is a 10 digit decimal number in the form $YYMMDDKKKC$, where $YYMMDD$ specifies the birthday date of a person, KKK is a distinctive suffix for persons having the same birthday date and C is the check digit. The check digit has to satisfy the condition that the decimal number $YYMMDDKKKC$ is divisible by 11.

Let us have a 10 digit decimal number $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$. Let us study which errors can our code detect. The condition of divisibility by 11 leads to the following check equation:

$$\sum_{i=0}^{9} 10^i.a_i \equiv 0 \mod 11 \ .$$

If i is even, i. e., $i = 2k$, then $10^i = 10^{2k} = 100^k = (99+1)^k$.

By the binomial theorem we can write:

$$10^i = (99+1)^k = \binom{k}{k}99^k + \binom{k}{k-1}99^{k-1} + \cdots + \binom{k}{1}99^1 + 1 \ . \tag{4.25}$$

Since 99 is divisible by 11, the last expression implies:

$$10^i \equiv 1 \mod 11 \quad \text{for } i \text{ even.}$$

If i is odd, i. e., $i = 2k+1$, then $10^i = 10^{2k+1} = 10.100^k = 10.(99+1)^k$. Utilizing (4.25) we can write

$$\begin{aligned} 10^i = 10.(99+1)^k = 10. &\left[\binom{k}{k}99^k + \binom{k}{k-1}99^{k-1} + \cdots + \binom{k}{1}99^1 + 1\right] = \\ &= 10.\binom{k}{k}99^k + 10.\binom{k}{k-1}99^{k-1} + \cdots + 10.\binom{k}{1}99^1 + 10 \ . \end{aligned}$$

From the last expression we have:

$$10^i \equiv 10 \mod 11 \quad \text{for } i \text{ odd.}$$

The check equation of 10 digit personal identification number is equivalent with:

$$a_0 + 10a_1 + a_2 + 10a_3 + a_4 + 10a_5 + a_6 + 10a_7 + a_8 + 10a_9 \equiv 0 \mod 11 \tag{4.26}$$

from where we can see that the code of personal identification numbers detects all single simple errors and all single neighbour transpositions[3].

The reader may ask what to do when the check digit C is equal to 10 for some $YYMMDDXXX$. In such cases the distinctive suffix XXX is skipped and the next one is used.

4.9 Codes with check digit over a group*

In this section we will be making efforts to find a decimal code with one check digit capable to detect one error of the two types: a simple error or a neighbour transposition.

Codes with code alphabet $B = \{0, 1, \dots, 9\}$ and with check equation mod 10 detected single simple error if and only if the mapping $\delta : B \to B$ defined by formula $\delta(a_i) = (w_i.a_i \mod 10)$ was an one to one mapping – a permutation of the set B. The assignment $\delta(x)$ defined as the sum of digits of $2.x$ used in UIC railway car encoding is a permutation of the set B of decimal digits. UIC railway car code is, till now, the most successful decimal code from the point of view of detecting one simple error and one neighbour transposition at the same time. We have also seen that the decimal code with check equation mod 10 is not able to detect one single error and at the same time one neighbour transposition – see theorem 4.5, 89.

This suggests an idea to replace summands $w_i a_i$ by permutations $\delta(a_i)$ in check equation (4.23). The new check equation is in the form

$$\sum_{i=1}^{n} \delta(a_i) \equiv c \mod 10 \tag{4.27}$$

Example 4.15. UIC railway car number is in fact a code with permutations

$$\delta_1 = \delta_3 = \dots = \delta_{11} := \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 2 & 4 & 6 & 8 & 1 & 3 & 5 & 7 & 9 \end{pmatrix}$$
$$\delta_2 = \delta_4 = \dots = \delta_{12} := \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix}$$

[3] The reader can easily verify that the check equation (4.26) is equivalent also with the equation:

$$a_0 - a_1 + a_2 - a_3 + a_4 - a_5 + a_6 - a_7 + a_8 - a_9 \equiv 0 \mod 11.$$

and with the check equation

$$\sum_{i=1}^{12} \delta_i(a_i) \equiv 0 \mod 10 \ .$$

Example 4.16. German postal money-order number is a 10 digit decimal code $a_1a_2 \dots a_{10}$ with check digit a_{10} and with the check equation

$$\sum_{i=1}^{10} \delta_i(a_i) \equiv 0 \mod 10 \ ,$$

where

$$\delta_1 = \delta_4 = \delta_7 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \end{pmatrix} \qquad \delta_2 = \delta_5 = \delta_8 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2 & 4 & 6 & 8 & 0 & 1 & 3 & 5 & 7 & 9 \end{pmatrix}$$

$$\delta_3 = \delta_6 = \delta_9 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 6 & 9 & 1 & 4 & 7 & 0 & 2 & 5 & 8 \end{pmatrix} \qquad \delta_{10} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

None of mentioned codes detects both simple error and neighbour transposition. Therefore – as the further generalization – we replace the group of residue classes[4] mod m by another group $\mathbb{G} = (A, *)$. The check equation will be formulated as

$$\prod_{i=1}^{n} \delta_i(a_i) = c \ . \tag{4.28}$$

The multiplicative form of group operation $*$ indicates that the group $\mathbb{G}$ need not be commutative.

Definition 4.9. Let A be an alphabet, let $\mathbb{G} = (A, *)$ be a group. Let $\delta_1, \delta_2, \dots, \delta_n$, are permutations of A. Then the code defined by check equation (4.28) is called **code with check digit over the group** $\mathbb{G}$.

Permutations are one to one mappings. Therefore, for every permutation δ of A there exits unique inverse permutation δ^{-1} of A for which it holds

$$\delta(a) = x \quad \text{if and only if} \quad \delta^{-1}(x) = a \ .$$

[4] The check equation (4.23) can be equivalently formulated as

$$\delta_1(a_1) \oplus \delta_2(a_2) \oplus \dots \oplus \delta_n(a_n) = c,$$

where operation $x \oplus y = (x + y) \mod (10)$ is a group operation on the set $B = \{0, 1, \dots, 9\}$ – the structure $(B, \oplus)$ is a group called group of residue classes mod 10.

Having two permutations δ_i, δ_j of A, we can define a new permutation by the formula $\forall a \in A\ a \mapsto \delta_i\bigl(\delta_j(a)\bigr)$. The new permutation will be denoted by $\delta_i \circ \delta_j$ and thus:

$$\delta_i \circ \delta_j(a) = \delta_i\bigl(\delta_j(a)\bigr) \quad \forall a \in A\ .$$

Theorem 4.6. *A code $\mathcal{K}$ with check digit of the group $\mathbb{G} = (A, *)$ detects neighbour transposition on positions i and $i+1$ if and only if:*

$$x * \delta_{i+1} \circ \delta_i^{-1}(y) \neq y * \delta_{i+1} \circ \delta_i^{-1}(x) \tag{4.29}$$

for arbitrary $x \in A$, $y \in A$, $x \neq y$.

For an Abel (i. e., commutative) group $\mathbb{G} = (A, +)$ the equation (4.29) can be rewritten in the form $x + \delta_{i+1} \circ \delta_i^{-1}(y) \neq y + \delta_{i+1} \circ \delta_i^{-1}(x)$, from where we have the following corollary:

Corollary. *A code $\mathcal{K}$ with check digit over an Abel group $\mathbb{G} = (A, +)$ detects neighbour transposition of arbitrary digits on positions i, $i+1$ if and only if it holds for arbitrary $x,\ y \in A$, $x \neq y$:*

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) \neq y - \delta_{i+1} \circ \delta_i^{-1}(y). \tag{4.30}$$

Proof. Let the code $\mathcal{K}$ detects neighbour transposition of arbitrary digits on positions i, $i+1$. Then for arbitrary a_i, a_{i+1} such that $a_i \neq a_{i+1}$ it holds:

$$\delta_i(a_i) * \delta_{i+1}(a_{i+1}) \neq \delta_i(a_{i+1}) * \delta_{i+1}(a_i) \tag{4.31}$$

For arbitrary $x \in A$ there exists $a_i \in A$ such that $a_i = \delta_i^{-1}(x)$. Similarly for arbitrary $y \in A$ there exists $a_{i+1} \in A$ such that $a_{i+1} = \delta_i^{-1}(y)$. Substitute x for $\delta_i(a_i)$ and y for $\delta_i(a_{i+1})$, then $\delta_i^{-1}(x)$ for a_i and $\delta_i^{-1}(y)$ for a_{i+1} in (4.31) . We get:

$$\begin{gathered}
x * \delta_{i+1}(a_{i+1}) \neq y * \delta_{i+1}(a_i) \\
x * \delta_{i+1}\bigl(\delta_i^{-1}(y)\bigr) \neq y * \delta_{i+1}\bigl(\delta_i^{-1}(x)\bigr) \\
x * \delta_{i+1} \circ \delta_i^{-1}(y) \neq y * \delta_{i+1} \circ \delta_i^{-1}(x)
\end{gathered}$$

and hence (4.29) holds.

Let (4.29) holds for all $x,\ y \in A$, $x \neq y$. Then (4.29) holds also for $x = \delta_i(a_i)$, $y = \delta_i(a_{i+1})$, where $a_i,\ a_{i+1} \in A$, $a_i \neq a_{i+1}$.

$$\delta_i(a_i) * \delta_{i+1} \circ \delta_i^{-1}(\delta_i(a_{i+1})) \neq \delta_i(a_{i+1}) * \delta_{i+1} \circ \delta_i^{-1}(\delta_i(a_i))$$

$$\delta_i(a_i) * \delta_{i+1}\Big(\underbrace{\delta_i^{-1}\big(\delta_i(a_{i+1})\big)}_{a_{i+1}}\Big) \neq \delta_i(a_{i+1}) * \delta_{i+1}\Big(\underbrace{\delta_i^{-1}\big(\delta_i(a_i)\big)}_{a_i}\Big)$$

$$\delta_i(a_i) * \delta_{i+1}(a_{i+1}) \neq \delta_i(a_{i+1}) * \delta_{i+1}(a_i)\ ,$$

what implies that the code $\mathcal{K}$ detects neighbour transposition of arbitrary digits on positions i, $i+1$. ■

Note the formula (4.30). It says that the assignment $x \mapsto \big(x - \delta_{i+1} \circ \delta_i^{-1}(x)\big)$ is one to one mapping – permutation.

Definition 4.10. A permutation δ of a (multiplicative) group $\mathbb{G} = (A, *)$ is called **complete mapping**, if the mapping defined by the formula

$$\forall x \in A \quad x \mapsto \eta(x) = x * \delta(x)$$

is also a permutation.
A permutation δ of a (additive) group $\mathbb{G} = (A, +)$ is called **complete mapping**, if the mapping defined by the formula

$$\forall x \in A \quad x \mapsto \eta(x) = x + \delta(x)$$

is also a permutation.

Theorem 4.7. *A code $\mathcal{K}$ with check digit over an Abel group $\mathbb{G} = (A, +)$ detects one simple error and one neighbour transposition if and only if there exists a complete mapping of group $\mathbb{G}$.*

Proof. Define the mapping $\mu : A \to A$ by the formula $\mu(x) = -x$. The mapping μ is a bijection – it is a permutation. The mapping $x \mapsto -\delta(x) = \mu \circ \delta(x)$ is again a permutation of set A for arbitrary permutation δ of A .

Let the code $\mathcal{K}$ detects neighbour transpositions. Then the mapping $x \mapsto \big(x - \delta_{i+1} \circ \delta_i^{-1}(x)\big)$ is a permutation by corollary of the theorem 4.6. But

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) = x + \underbrace{\mu \circ \delta_{i+1} \circ \delta_i^{-1}(x)}_{\delta(x)} = x + \delta(x)$$

The permutation δ defined by the formula $\delta = \mu \circ \delta_{i+1} \circ \delta_i^{-1}$ is the required complete mapping of $\mathbb{G}$.

Let δ be a complete mapping of group $\mathbb{G}$. Define:

$$\delta_i = (\mu \circ \delta)^i. \tag{4.32}$$

Then

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) = x - (\mu \circ \delta)^{i+1} \circ (\mu \circ \delta)^{-i}(x) = x - (\mu \circ \delta)(x) = x + \delta(x),$$

what implies that $x - \delta_{i+1} \circ \delta_i^{-1}(x)$ is a permutation. By the corollary of the theorem 4.6, the code with check digit over the group $\mathbb{G}$ with permutations δ_i defined by (4.32) detects neighbour transpositions. ■

Theorem 4.8. *Let $\mathbb{G} = (A, +)$ be an Abel finite group. Then the following assertions hold (see [11], 8.11 page. 63):*

a) If $\mathbb{G}$ group of an odd order then identity is complete mapping of $\mathbb{G}$.

b) A group $\mathbb{G}$ of order $r = 2.m$ where m is an odd number has no compete mapping.

c) Let $\mathbb{G} = (A, +)$ be an Abel group of the even order. A complete mapping of $\mathbb{G}$ exists if an only if $\mathbb{G}$ contains at least two different involutions, i. e., such elements $g \in\ A$ that $g \neq 0$, and $g + g = 0$

Proof. The proof of this theorem exceeds the frame of this publication. The reader can find it in [11]. ■

Let us have the alphabet $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Let $(A, \oplus)$ be an arbitrary Abel group (let $\oplus$ be an arbitrary binary operation on A such that $(A, \oplus)$ is a commutative group). Since the order of group $(A, \oplus)$ is $10 = 2 \times 5$ there is no complete mapping of this group.

Corollary. There is no decimal code with check digit over an Abel group $\mathbb{G} = (A, \oplus)$ where $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ detecting simple errors and neighbour transpositions.

The only chance for designing a decimal code capable to detect simple errors and neighbour transpositions is to try a code with check digit over a non–commutative group.

Definition 4.11. Dieder group $\mathbb{D}_n$ is a finite group of order $2.n$ of the form

$$\{1, a, a^2, \dots, a^{n-1}, b, ab, a^2.b, \dots, a^{n-1}b\},$$

where it holds

$$a^n = 1 \quad (a^i \neq 1 \text{ for } i = 1, 2, \dots, n-1)$$

$$\begin{aligned} b^2 &= 1 \quad (b \neq 1) \\ b.a &= a^{n-1}.b \end{aligned}$$

Dieder group $\mathbb{D}_n$ will be denoted

$$\mathbb{D}_n = \left\langle a, b \;\middle|\; a^n = 1 = b^2,\ ba = a^{n-1}b \right\rangle$$

Dieder group $\mathbb{D}_n$ can be interpreted as a group of symmetries of the regular n-sided polygon – the element a expresses rotation around the center by angle $2\pi/n$, the element b expresses axial symmetry. Let us have $\mathbb{D}_3$, let (ABC) is a regular triangle. Then $1 = (ABC)$, $a = (CAB)$, $a^2 = (BCA)$, $b = (ACB)$, $ab = (BAC)$, $a^2b = (CBA)$.

Example 4.17. Dieder group $\mathbb{D}_3 = \left\langle a, b \;\middle|\; a^3 = 1 = b^2,\ ba = a^2b \right\rangle$. The elements of $\mathbb{D}_3$ can be assigned to integers from 1 to 6:

1	a	a^2	b	ab	a^2b
1	2	3	4	5	6

Denote by $\oplus$ the corresponding group operation on the set $\{1, 2, \ldots, 6\}$. Then we calculate

$$2 \otimes 3 = a.a^2 = a^3 = 1$$

$$3 \otimes 6 = a^2.a^2b = a^4b = a^3.ab = 1.ab = ab = 5$$

$$6 \otimes 3 = a^2b.a^2 = ba.a^2 = ba^3 = b.1 = b = 4$$

$$4 \otimes 5 = b.ab = ba.b = a^2b.b = a^2.b^2 = a^2.1 = a^2 = 3$$

$$5 \otimes 4 = ab.b = a.b^2 = a.1 = a = 2$$

Theorem 4.9. *Let $\mathbb{D}_n = \left\langle a, b \;\middle|\; a^n = 1 = b^2,\ ba = a^{n-1}b \right\rangle$ be a Dieder group of an odd degree n, $n \geq 3$. Define a permutation $\delta : \mathbb{D}_n \to \mathbb{D}_n$ by the formula:*

$$\delta(a^i) = a^{n-1-i} \quad a \quad \delta(a^ib) = a^ib \quad \forall i = 0, 1, 2, \ldots, n-1\ . \tag{4.33}$$

Then it holds for the permutation δ:

$$x.\delta(y) \neq y.\delta(x) \quad \forall x, y \in \mathbb{D}_n \text{ such that } x \neq y\ . \tag{4.34}$$

Proof. Let us realize one fact before proving the theorem. It holds by definition of Dieder group that $b.a = a^{n-1}b$. Since $a^{n-1}.a = 1$, it holds $a^{n-1} = a^{-1}$, and that is why it holds $ba = a^{-1}b$. Let k be an arbitrary natural number. Then $b.a^k = a^{-1}ba^{k-1} = a^{-2}ba^{k-2} = \cdots = a^{-k}b$.
For arbitrary integer number it holds:

$$b.a^k = a^{-k}b \ . \tag{4.35}$$

Now return to the proof of theorem. Let δ be defined by (4.33). It is easy to seen that δ is a permutation. We want to prove (4.34). We will distinguish three cases:
1. case:
Let $x = a^i$, $y = a^j$, where $i \neq j$, let $0 \leq i, j \leq n-1$.
Suppose that $x.\delta(y) = y.\delta(x)$ then $a^i.a^{n-1-j} = a^j.a^{n-1-i}$ which implies $a^{2i-2j} = a^{2(i-j)} = 1$. The number $2(i-j)$ has to be divisible by an odd number n, otherwise $2(i-j) = kn + r$, where $1 \leq r \leq n-1$, and then $a^{2(i-j)} = a^{kn+r} = a^{kn}a^r = 1.a^r \neq 1$. If an odd n divides $2(i-j)$, the number $(i-j)$ has to be divisible by n, from which it follows that $(i-j) = 0$ because $0 \leq i, j \leq n-1$.
2. case:
Let $x = a^i$, $y = a^jb$, $0 \leq i, j \leq n-1$.
Suppose $x.\delta(y) = y.\delta(x)$, i. e., $a^ia^jb = a^jba^{n-1-i}$. Using (4.35) we have $a^{i+j}b = a^j.a^{i+1}b$, from where we get step by step $a^{i+j} = a^{i+j+1}$, $1 = a$. However, $a \neq 1$ for ≥ 3 in corresponding Dieder group $\mathbb{D}_n$ for $n \geq 3$.
3. case:
Let $x = a^ib$, $y = a^jb$, $0 \leq i, j \leq n-1$.
Let $x.\delta(y) = y.\delta(x)$ which means in this case $a^ib.a^jb = a^jba^ib$. Using (4.35) we have $a^ibb.a^{-j} = a^jbba^{-i}$. Since $b.b = b^2 = 1$, the last equation can be rewritten as $a^{i-j} = a^{j-i}$, and thus $a^{2(i-j)} = 1$. In the same way as in the 1. case we can show that this implies $i = j$. ■

Theorem 4.10. *Let* $\mathbb{D}_n = \langle a, b \mid a^n = 1 = b^2,\ ba = a^{n-1}b \rangle$ *be a Dieder group of an odd order* n*,* $n \geq 3$*. Let* $\delta : \mathbb{D}_n \to \mathbb{D}_n$ *be the permutation defined by the formula* (4.33)*. Define*

$$\delta_i = \delta^i \quad \textit{for } i = 1, 2, \ldots, m.$$

Then the block code of the length m *with check digit over the group* $\mathbb{D}_n$ *detects simple errors and neighbour transpositions.*

Proof. By contradiction. It suffices to prove (by the theorem 4.6) that it holds for code characters $x,\ y$ such that $x \neq y$

$$x * \delta_{i+1} \circ \delta_i^{-1}(y) \neq y * \delta_{i+1} \circ \delta_i^{-1}(x)$$

Let for some $x \neq y$ equality in the last formula holds. Using substitutions $\delta_i = \delta^i,\ \delta_{i+1} = \delta^{i+1}$ we get:

$$x * \delta^{i+1} \circ \delta^{-i}(y) = y * \delta^{i+1} \circ \delta^{i}(x)$$
$$x * \delta(y) = y * \delta(x),$$

what contradicts with the property (4.34) of permutation δ. ■

Remark. Definition 4.33 can be generalized by the following way: Define $\delta : \mathbb{D}_n \to \mathbb{D}_n$ by the formula

$$\delta(a^i) = a^{c-i+d} \quad \text{and} \quad \delta(a^i b) = a^{i-c+d} b \quad \forall i = 1, 2, \ldots, n-1 \tag{4.36}$$

The permutation δ defined in the definition (4.33) is a special case of that defined in (4.36), namely for $c = d = \dfrac{n-1}{2}$.

Example 4.18. Dieder group $\mathbb{D}_5 = \langle a, b \mid a^5 = 1 = b^2,\ ba = a^4 b \rangle$. The elements of $\mathbb{D}_5$ can be assigned to decimal characters as follows:

1	a	a^2	a^3	a^4	b	ab	a^2b	a^3b	a^4b
0	1	2	3	4	5	6	7	8	9

The following scheme can be used for group operation $i * j$:

$i * j$	$0 \le j \le 4$	$5 \le j \le 9$
$0 \le i \le 4$	$(i + j) \mod 5$	$5 + [(i + j) \mod 5]$
$5 \le i \le 9$	$5 + [(i - j) \mod 5]$	$(i - j) \mod 5$

The corresponding table of operation $*$ is:

	j									
*	0	1	2	3	4	5	6	7	8	9
i 0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

4.10 General theory of error correcting codes

Let us have an alphabet $A = \{a_1, a_2, \dots, a_r\}$ with r characters. In this section we will explore block codes $\mathcal{K}$ of the length n (i. e., subsets of the type $\mathcal{K} \subset A^n$) from the point of view of general possibilities of detecting and correcting t simple errors.

Till the end of this chapter we will use the notation $d(\mathbf{v}, \mathbf{w})$ for the Hamming distance of two words $\mathbf{v}$ and $\mathbf{w}$, that was defined in definition 4.6 (page 84) as the number of places in which $\mathbf{v}$ and $\mathbf{w}$ differ. By the definition 4.7 (page 84), the minimum distance $\Delta\mathcal{K}$ of a block code $\mathcal{K}$ is minimum of Hamming distances of all pairs of different words of the code $\mathcal{K}$.

Remark. Maximum of distances of two words from A^n is n – namely in the case when corresponding words differ in all positions.

Theorem 4.11. *The Hamming distance is a metric on A^n, i. e., it holds:*

$$\begin{aligned} &d(\mathbf{a}, \mathbf{b}) \geq 0 \ ; \quad d(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b} \\ &d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a}) \\ &d(\mathbf{a}, \mathbf{b}) \leq d(\mathbf{a}, \mathbf{c}) + d(\mathbf{c}, \mathbf{b}) \end{aligned}$$

Hence (A^n, d) is a metric space.

Proof. The simple straightforward proof is left to the reader. ■

Definition 4.12. We will say that a code $\mathcal{K}$ **detects t-tuple simple errors** if the result of replacing arbitrary at least 1 and at most t characters of any code word $\mathbf{c}$ by different characters is a non code word. We say that **we have detected an error** after receiving a non code word.

Definition 4.13. A **ball** $B_t(\mathbf{c})$ **with center** $\mathbf{c} \in A^n$ **and radius** t is the set

$$B_t(\mathbf{c}) = \{\mathbf{x} \mid \mathbf{x} \in A^n,\ d(\mathbf{x}, \mathbf{c}) \leq t\}.$$

The ball $B_t(\mathbf{c})$ is the set of all such words which originated from the word $\mathbf{c}$ by at most t simple errors.

Calculate how many words the ball $B_t(\mathbf{c})$ contains provided $|A| = r$. Let $\mathbf{c} = c_1c_2 \dots c_n$. The number of words $\mathbf{v} \in A^n$ with $d(\mathbf{c}, \mathbf{v}) =$ is $n.(r-1) = \binom{n}{1}.(r-1)$, since we can obtain $r-1$ words that differ from $\mathbf{c}$ at every position i, $i = 1, 2, \dots, n$.

To count the number of words which differ from $\mathbf{c}$ at k positions first choose a subset $\{i_1, i_2, \dots i_k\}$ of k indices – there are $\binom{n}{k}$ such subsets. Every character at every position $i_1, i_2, \dots i_k$ can be replaced by $r-1$ different characters what leads to $(r-1)^k$ different words. Thus the total number of words $\mathbf{v} \in A^n$ with $d(\mathbf{c}, \mathbf{v}) = k$ is $\binom{n}{k}(r-1)^k$. The word $\mathbf{c}$ itself is also an element of the ball $B_t(\mathbf{c})$ and contributes to its cardinality by the number $1 = \binom{n}{0}.(r-1)^0$. Therefore the number of words in $B_t(\mathbf{c})$ is

$$|B_t(\mathbf{c})| = \sum_{i=0}^{t} \binom{n}{i}.(r-1)^i\ . \tag{4.37}$$

The cardinality of the ball $B_t(\mathbf{c})$ does not depend on the center word $\mathbf{c}$ – all balls with the same radius t have the same cardinality (4.37).

Definition 4.14. We say that the code $\mathcal{K}$ **corrects** t **simple errors** if for every word $\mathbf{y}$ which originated from a code word by at most t simple errors, there exists an unique code word $\mathbf{x}$ such that $d(\mathbf{x}, \mathbf{y}) \leq t$.

Note that if $\mathbf{b} \in B_t(\mathbf{c}_1) \cap B_t(\mathbf{c}_2)$ then the word $\mathbf{b}$ could originated by at most t simple errors from both words $\mathbf{c}_1$, $\mathbf{c}_2$. Hence if the code $\mathcal{K}$ corrects t simple errors then the following formula

$$B_t(\mathbf{c}_1) \cap B_t(\mathbf{c}_2) = \emptyset \tag{4.38}$$

has to hold for an arbitrary pair $\mathbf{c}_1$, $\mathbf{c}_2$ of two distinct code words.

The reverse assertion is also true. If (4.37) holds for an arbitrary pair of code words of the code $\mathcal{K}$ then the code $\mathcal{K}$ corrects t simple errors.

Let a code $\mathcal{K} \subseteq A^n$ corrects t simple errors. Since $|A^n| = r^n$, it follows from formulas (4.37) and (4.38) that the number of code words $|\mathcal{K}|$ fulfils

$$\sum_{i=0}^{t} \binom{n}{i} .(r-1)^i \ . \ |\mathcal{K}| \leq r^n \ . \tag{4.39}$$

When designing a code which corrects t errors we try to utilize the whole set (A^n, d). The ideal case would be if the system of balls covered the whole set A^n, i. e., if (4.39) was equality. Such codes are called perfect.

Definition 4.15. We say that the code $\mathcal{K} \subseteq A^n$ is t-**perfect code**, if

$$\forall \mathbf{a},\ \mathbf{b} \in A^n, \quad \mathbf{a} \neq \mathbf{b} \quad B_t(\mathbf{a}) \cap B_t(\mathbf{b}) = \emptyset \ ,$$
$$\bigcup_{\mathbf{a} \in \mathcal{K}} B_t(\mathbf{a}) = A^n \ .$$

While perfect codes are very efficient, they are very rare – most of codes are not perfect.

Theorem 4.12. *A code $\mathcal{K}$ corrects t simple errors if and only if*

$$\Delta(\mathcal{K}) \geq 2t + 1 \ , \tag{4.40}$$

where $\Delta(\mathcal{K})$ is the minimum distance of the code $\mathcal{K}$.

Proof. By contradiction. Let (4.40) holds.
Suppose that there are two words $\mathbf{a} \in \mathcal{K}$, $\mathbf{b} \in \mathcal{K}$ such that $B_t(\mathbf{a}) \cap B_t(\mathbf{b}) \neq \emptyset$, let $\mathbf{c} \in B_t(\mathbf{a}) \cap B_t(\mathbf{b})$. By triangle inequality we have

$$d(\mathbf{a}, \mathbf{b}) \leq \underbrace{d(\mathbf{a}, \mathbf{c})}_{\leq t} + \underbrace{d(\mathbf{c}, \mathbf{b})}_{\leq t} \leq 2t,$$

what contradicts with assumption that $\Delta\mathcal{K} \geq 2t + 1$.

Let the code $\mathcal{K} \subseteq A^n$ corrects t simple errors. Then for arbitrary $\mathbf{a}$, $\mathbf{b} \in \mathcal{K}$ such that $\mathbf{a} \neq \mathbf{b}$ it holds $B_t(\mathbf{a}) \cap B_t(\mathbf{b}) = \emptyset$. Let $d(\mathbf{a}, \mathbf{b}) = s \leq 2t$. Create the following sequence of words

$$\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s \tag{4.41}$$

where $\mathbf{a}_0 = \mathbf{a}$, and having defined $\mathbf{a}_i$ we define $\mathbf{a}_{i+1}$ as follows: Compare step by step the characters at the first, the second,..., n-th position of both words $\mathbf{a}_i$ and $\mathbf{b}$ until different characters are found on the position denoted by k. Create the word $\mathbf{a}_{i+1}$ as the word $\mathbf{a}_i$ in which the k-th character is substituted by k-th character of the word $\mathbf{b}$.

The sequence (4.41) represents one of several possible procedures of transforming the word $\mathbf{a}$ into the word $\mathbf{b}$ by stepwise impact of simple errors.

Clearly $\mathbf{a}_s = \mathbf{b}$, $d(\mathbf{a}, \mathbf{a}_i) = i$ and $d(\mathbf{a}_i, \mathbf{b}) = s - i$ for $i = 1, 2, \dots, s$. Therefore, $d(\mathbf{a}, \mathbf{a}_t) = t$, $\mathbf{a}_t \in B_t(\mathbf{a})$ and also $d(\mathbf{a}_t, \mathbf{b}) = s - t \leq 2t - t = t$, and hence $\mathbf{a}_t \in B_t(\mathbf{b})$, what contradicts with the assumption that $B_t(\mathbf{a}) \cap B_t(\mathbf{b}) = \emptyset$. ■

Example 4.19. Suppose we have the alphabet $A = \{a_1, a_2, \dots, a_r\}$. The repeating code of the length k is the block code whose every code word consists of k same characters, i. e., $\mathcal{K} = \{a_1a_1\dots a_1,\ a_2a_2\dots a_2,\ \dots\ ,\ a_ra_r\dots a_r\}$. The minimum distance of the code $\mathcal{K}$ is $\Delta\mathcal{K} = k$ and such a code corrects t simple errors for $t < k/2$. Specially for $r = 2$ (i. e., for the binary alphabet A)and k odd, i. e., $k = 2t + 1$ the repeating code is t-perfect.

Example 4.20. The minimum distance of the 8-bit-even-parity-code is 2 (see example 4.7, page 85), that is why it does not correct even one simple error.

Example 4.21. Two dimensional parity check code. This is a binary code. Information bits are written into a matrix of the type (p, q). Then the even parity check bit is added to every row and to every column. Finally, the even parity "check character of check characters" is added. This code corrects one simple error. Such error will change the parity of exactly one row i and exactly one column j. Then the incorrect bit is in the position (i, j). The example of one code word of the length 32 for $p = 3$, $q = 7$ follows:

$$\begin{array}{r|l}
101 & 0 \leftarrow \text{row check digit} \\
000 & 0 \\
001 & 1 \\
010 & 1 \\
111 & 1 \\
111 & 1 \\
000 & 0 \\
\hline
\text{column check digits} \rightarrow 110 & 0 \leftarrow \text{check digit of check digits}
\end{array}$$

Suppose, we have a code $\mathcal{K}$ which corrects t errors and we have received a word $\mathbf{a}$. We need an instruction how to determine the transmitted word from the received word $\mathbf{a}$ provided at most t simple errors occurred during transmission.

Definition 4.16. The **decoding of code $\mathcal{K}$** (or **code decoding of $\mathcal{K}$**) is an arbitrary mapping δ with codomain $\mathcal{K}$, whose domain $\mathcal{D}(\delta)$ is a subset of the set A^n, which contains as a subset the code $\mathcal{K}$ and for which it holds: for arbitrary $\mathbf{a} \in \mathcal{K}$ it holds $\delta(\mathbf{a}) = \mathbf{a}$.

$$\delta : \mathcal{D}(\delta) \to \mathcal{K}, \qquad \mathcal{K} \subset \mathcal{D}(\delta) \subseteq A^n, \qquad \delta : \mathcal{D}(\delta) \to \mathcal{K} \;, \qquad \forall \mathbf{a} \in \mathcal{K} \;\; \delta(\mathbf{a}) = \mathbf{a} \;.$$

If $\mathcal{D}(\delta) = A^n$, we say that the decoding of the code $\mathcal{K}$ is **complete decoding of the code $\mathcal{K}$**, otherwise we say that δ is **partial decoding of a code $\mathcal{K}$**.

Remark. Please distinguish between the terms "decoding function" (or simple "decoding") which is used for inverse function of encoding K, while the term "decoding of the code $\mathcal{K}$" is a function which for some words $\mathbf{a}$ from A^n says which word was probably transmitted if we received the word $\mathbf{a}$.

Some codes allow to differentiate the characters of code words into characters carrying an information and check characters. Check characters are fully determined by information characters. Even-parity codes (example 4.7, page 85), UIC railway car number (example 4.10), EAN code (example 4.11) ISBN code (example 4.12), Slovak personal identification number (example 4.14) are examples of such codes with the last digit in the role of check digit.

If we know how the meanings of single positions of code words were defined, we have no problem with distinguishing between information and check characters. The problem is how to make differentiation when we know only the set $\mathcal{K}$ of code words. The following definition gives us the answer:

Definition 4.17. Let $\mathcal{K} \subseteq A^n$ be a block code of the length n. We say that the **code $\mathcal{K}$ has k information and $n-k$ check characters**, if there exists an one–to–one mapping $\phi : A^k \leftrightarrow \mathcal{K}$. The mapping ϕ is called **encoding of information characters**.

Example 4.22. The repeating block code of the length 5 with the alphabet $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ has one information character and 4 check characters since the mapping ϕ defined by:

$$\begin{array}{lllll} \phi(0) = 00000 & \phi(1) = 11111 & \phi(2) = 22222 & \phi(3) = 33333 & \phi(4) = 44444 \\ \phi(5) = 55555 & \phi(6) = 66666 & \phi(7) = 77777 & \phi(8) = 88888 & \phi(9) = 99999 \end{array}$$

is an one–to–one mapping $\phi : A^1 \leftrightarrow \mathcal{K}$.

Example 4.23. The doubling code of the length $2n$ has n information characters and n check characters. The encoding of information characters $\phi : A^n \leftrightarrow \mathcal{K}$ is defined by the formula:

$$\phi(a_1a_2 \dots a_n) = a_1a_1a_2a_2 \dots a_na_n.$$

Example 4.24. The two-out-of-five-code (see example 4.6 page 84) has not distinguished the information characters from the check ones. The number of code words of this code is $|\mathcal{K}| = 10$ and this number is not an integer power of 2, therefore there cannot exist an one to one mapping $\phi : \{0, 1\}^k \to \mathcal{K}$.

In many examples we have seen that the check digit was the last digit of the code word. Similarly we would like to have codes with k information and $n - k$ check characters in such a form that the first k characters of code words are the information characters and $n - k$ remaining are the check characters. Such codes are called systematic.

Definition 4.18. A block code $\mathcal{K}$ is called **systematic code** with k information characters and $n - k$ check characters if for every word $a_1a_2 \dots a_k \in A^k$ there exists exactly one code word $\mathbf{a} \in \mathcal{K}$ such that

$$\mathbf{a} = a_1a_2 \dots a_k, a_{k+1} \dots a_n \ .$$

Example 4.25. The repeating code of the length n is a systematic code with $k = 1$. The even parity code of the length 8 is a systematic code with $k = 7$. UIC railway car number is a systematic code with $k = 11$.
Doubling code of the length $2n$, greater than 2, is not systematic.

Theorem 4.13. *Let $\mathcal{K}$ be a systematic code with k information characters and $n - k$ check characters, let $\Delta\mathcal{K}$ be the the minimum distance of $\mathcal{K}$. Then it holds:*

$$\Delta\mathcal{K} \leq n - k + 1 \ . \tag{4.42}$$

Proof. Choose two words $\mathbf{a} = a_1a_2 \dots a_{k-1}a_k \in A^k$, $\overline{\mathbf{a}} = a_1a_2 \dots a_{k-1}\overline{a}_k \in A^k$ which differ only in the last k-th position. Since the code $\mathcal{K}$ is systematic, for every word $\mathbf{a}$, resp., $\overline{\mathbf{a}}$ there exists exactly one code word $\mathbf{b}$, resp., $\overline{\mathbf{b}}$ such that $\mathbf{a}$ is the prefix of $\mathbf{b}$, resp. $\overline{\mathbf{a}}$ is the prefix of $\overline{\mathbf{b}}$:

$$\mathbf{b} = a_1a_2 \dots a_{k-1}a_ka_{k+1} \dots a_n \ ,$$
$$\overline{\mathbf{b}} = a_1a_2 \dots a_{k-1}\overline{a}_k\overline{a}_{k+1} \dots \overline{a}_n \ .$$

Since the words $\mathbf{b}$, $\overline{\mathbf{b}}$ have the same characters in $k-1$ positions, they can have at most $n-(k-1) = n-k+1$ different characters. Therefore $d(\mathbf{b}, \overline{\mathbf{b}}) \leq n-k+1$ and hence $\Delta\mathcal{K} \leq n-k+1$. ■

Corollary A code $\mathcal{K}$ with k information and $n-k$ check characters can correct at most $\left[\dfrac{n-k}{2}\right]$ errors (where $[x]$ is the integral part of x).

Example 4.26. For the doubling code of the length $n = 2t$ is $k = t$, $n-k = t$, but the minimum distance of this code is 2 – this number is much lower for large t then the upper estimation (4.42) which gives for this case $\Delta\mathcal{K} \leq 2t-t+1 = t+1$.

Definition 4.19. Let $\mathcal{K}$ be a code with k information and $n-k$ check characters. The fraction

$$R = \frac{k}{n} \tag{4.43}$$

is called **information ratio**.

Designers of error correcting codes want to protect the code against as large number of errors as possible – this leads to increasing the number of check digits – but the other natural requirement is to achieve as large information ratio as possible. The mentioned aims are in contradiction. Moreover we can see that adding check characters need not result in larger minimum distance of code (see example 4.26).

4.11 Recapitulation of some algebraic structures

Group $(G, .)$ is a set G with a binary operation "." assigning to every two elements $a \in G$, $b \in G$ an element $a.b$ (shortly only ab) such that it holds:

(i) $\forall a, b \in G \; ab \in G$

(ii) $\forall a, b, c \in G \; (ab)c = a(bc)$ – associative law

(iii) $\exists\, 1 \in G \; \forall a \in G \quad 1a = a1 = a$ – existence of a neutral element

(iv) $\forall a \in G \; \exists a^{-1} \in G \quad aa^{-1} = a^{-1}a = 1$ – existence of an inverse element

The group G is **commutative** if it holds $\forall a, b \in G\ ab = ba$. Commutative groups are also called **Abel groups**. In this case additive notation of group binary operation is used, i. e., $a + b$ instead of $a.b$ and the neutral element is denoted by 0. The inverse element to element a in the commutative group is denoted by $-a$.

Field $(T, +, .)$ is a set T containing at least two elements 0 and 1 together with two binary operations "+" and "." such that it holds:

(i) The set T with binary operation "+" is a commutative group with neutral element 0.

(ii) The set $T - \{0\}$ with binary operation "." is a commutative group with neutral element 1.

(iii) $\forall a, b, c \in G \quad a(b + c) = ab + ac$ – distributive law

Maybe the properties of fields are better visible if we rewrite (i), (ii), (iii), of the definition of the field into single conditions:

Field is a set T containing at least two elements 0 and 1 together with two binary operations "+" and "." such that it holds:

(T1) $\forall a, b \in T\ a + b \in T,\ ab \in T.$

(T2) $\forall a, b, c \in T\ a + (b + c) = (a + b) + c, \quad a(bc) = (ab)c$ – associative laws

(T3) $\forall a, b \in T\ a + b = b + a, \quad ab = ba$ – commutative laws

(T4) $\forall a, b, c \in T\ a(b + c) = ab + ac$ – distributive law

(T5) $\forall a \in T\ a + 0 = a, \quad a.1 = a$

(T6) $\forall a \in T\ \exists(-a) \in T\ a + (-a) = 0$

(T7) $\forall a \in T,\ a \neq 0\ \exists a^{-1} \in T\ a.a^{-1} = 1$

Commutative ring with 1 is a set R containing at least two elements $0 \in R$ and $1 \in R$ together with two operations + and ., in which (T1) till (T6) hold.

Example 4.27. The set $\mathbb{Z}$ of all integers with operations ”+” and ”.” is commutative ring with 1. However, the structure $(\mathbb{Z}, +, .)$ is not a field since (T7) does not hold.

Factor ring modulo p. Let us have the set $\mathbb{Z}_p = \{0, 1, 2, \dots, p-1\}$. Define two binary operations $\oplus$, $\otimes$ on the set $\mathbb{Z}_p$:

$$a \oplus b = (a+b) \mod p \qquad a \otimes b = (ab) \mod p,$$

where $n \mod p$ is the remainder after integer division of the number n by p. It can be easily shown that for an arbitrary natural number $p > 1$ the structure $(\mathbb{Z}_p, \oplus, \otimes)$ is a commutative ring with 1, i. e., it fulfills conditions (T1) till (T6).

We will often write + and . instead of $\oplus$ and $\otimes$ – namely in situations where no misunderstanding may happen.

Example 4.28. The ring $\mathbb{Z}_6$ has the following tables for operation $\oplus$ and $\otimes$:

$\oplus$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	5

$\otimes$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

By the above tables it holds $5 \otimes 5 = 1$, i. e., the inverse element to 5 is the element 5. The elements 2, 3, 4 have no inverse element at all. The condition (T7) does not hold in $\mathbb{Z}_6$, therefore $\mathbb{Z}_6$ is not a field.

For coding purposes such factor rings $\mathbb{Z}_p$ are important, which are fields. When is the ring $\mathbb{Z}_p$ also a field? The following theorem gives the answer.

Theorem 4.14. *Factor ring $\mathbb{Z}_p$ is a field if and only if p is a prime number.*

Proof. The reader can find an elementary proof of this theorem in the book [1].
■

Linear space over the field F. Let $(F, +, .)$ be a field. The **linear space over the field** F is a set $\mathcal{L}$ with two binary operations: vector addition: $\mathcal{L} \times \mathcal{L} \to \mathcal{L}$ denoted $\mathbf{v} + \mathbf{w}$, where $\mathbf{v}, \mathbf{w} \in \mathcal{L}$, and scalar multiplication: $F \times \mathcal{L} \to \mathcal{L}$ denoted $t.\mathbf{v}$, where $t \in F$ and $\mathbf{v} \in \mathcal{L}$, satisfying axioms below:

(L1) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L}$ a $\forall t \in T \quad \mathbf{u} + \mathbf{v} \in \mathcal{L}$, $t.\mathbf{u} \in \mathcal{L}$.

(L2) $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{L} \qquad \mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.

(L3) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L} \qquad \mathbf{u} + \mathbf{b} = \mathbf{b} + \mathbf{u}$.

(L4) $\exists \mathbf{o} \in \mathcal{L}$ such that $\forall \mathbf{u} \in \mathcal{L} \quad \mathbf{u} + \mathbf{o} = \mathbf{u}$

(L5) $\forall \mathbf{u} \in \mathcal{L} \quad \exists (-\mathbf{u}) \in \mathcal{L}$ such that $\mathbf{u} + (-\mathbf{u}) = \mathbf{o}$

(L6) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L}$ a $\forall t \in T \quad t.(\mathbf{u} + \mathbf{v}) = t.\mathbf{u} + t.\mathbf{v}$

(L7) $\forall \mathbf{u} \in \mathcal{L}$ a $\forall s, t \in T \quad (s.t)\mathbf{u} = s.(t.\mathbf{u})$

(L8) $\forall \mathbf{u} \in \mathcal{L}$ a $\forall s, t \in T \quad (s + t)\mathbf{u} = s.\mathbf{u} + t.\mathbf{u}$

(L9) $\forall \mathbf{u} \in \mathcal{L} \quad 1.u = u$.

The requirements (L1) till (L5) are equivalent to the condition that $(\mathcal{L}, +)$ is a commutative group with neutral element $\mathbf{o}$. The synonym **vector space** is often used instead of linear space. Elements of a linear space are called **vectors**.

Vectors (or the set of vectors) $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ are called **linearly independent** if the only solution of the vector equation

$$t_1\mathbf{u}_1 + t_2\mathbf{u}_2 + \cdots + t_n\mathbf{u}_n = \mathbf{o}$$

is the n-tuple $(t_1, t_2, \dots, t_n)$ where $t_i = 0$ for $i = 1, 2, \dots, n$. Otherwise, we say that vectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ are **linearly dependent**.
Any linearly independent set is contained in some **maximal linearly independent set**, i.e. in a set which ceases to be linearly independent after any element in $\mathcal{L}$ has been added to it.

We say that the linear space $\mathcal{L}$ is **finite dimensional**, if there exists a natural number k such that every set of vectors with $k+1$ elements is linearly dependent. In a finite dimensional linear space $\mathcal{L}$ all maximal independent sets have the same cardinality n – this cardinality is called **dimension of linear space** $\mathcal{L}$ and $\mathcal{L}$ is called n**-dimensional linear space**.

Basis of finite dimensional linear space $\mathcal{L}$ is an arbitrary maximal linearly independent set if its vectors.

Let $(F,+,.)$ be a field. Linear space $(F^n,+,.)$ is the space of all ordered n-tuples of the type $\mathbf{u} = u_1u_2\dots u_n$ where $u_i \in T$ with vector addition and scalar multiplication defined as follows:
Let $\mathbf{u} = u_1u_2\dots u_n$, $\mathbf{v} = v_1v_2\dots v_n$, $t \in T$. Then

$$\mathbf{u}+\mathbf{v} = (u_1+v_1),(u_2+v_2),\dots(u_n+v_n) \qquad t.\mathbf{u} = (tu_1),(tu_2),\dots,(tu_n).$$

The linear space $(F^n,+,.)$ is called **arithmetic linear space over the field** F.
Scalar product of vectors $\mathbf{u} \in F^n$**,** $\mathbf{v} \in F^n$ is defined by the following formula:

$$\mathbf{u} * \mathbf{v} = u_1v_1 + u_2v_2 + \dots + u_nv_n$$

The vectors $\mathbf{u}$, $\mathbf{v}$ are called **orthogonal** if $\mathbf{u} * \mathbf{v} = 0$.

Importance of the arithmetic linear space $(F^n,+,.)$ over the field F follows from the next theorem:

Theorem 4.15. *Every n-dimensional linear space over the field F is isomorphic to the arithmetic linear space $(F^n,+,.)$ over the field F.*

The theory of linear codes makes use of the fact that the code alphabet A is a field with operations "+" and ".". Then the set of all words of the length n is n-dimensional arithmetic linear space over the field A. We have seen that a factor ring $\mathbb{Z}_p$ is a finite field if and only if p is prime. There are also other finite fields called Galois fields denoted by $GF(p^n)$ with p^n elements where p is prime. There are no other finite fields except fields of the type $\mathbb{Z}_p$ and $GF(p^n)$ with p prime.

In the theory of linear codes the cardinality of the code alphabet is limited to the numbers of the type p^n where p is prime and $n = 1,2,\dots$, i. e., 2,3,4,5,7,8,9,11,13,16,17..., but the code alphabet cannot contain 6,10,12,14,15, etc., elements because these numbers are not powers of primes. These limitations are not crucial since the most important code alphabet is the binary alphabet $\{0,1\}$ and alphabets with greater non feasible number of elements can be replaced by fields with the nearest greater cardinality (several characters of which will be unused).

4.12 Linear codes

In this chapter we will suppose that the code alphabet $A = \{a_1, a_2, \dots, a_p\}$ has p characters where p is a prime number or a power of prime. We further suppose that operations $+$ and $.$ are defined on A such that the structure $(A, +, .)$ is a finite field. Then we can create the arithmetic n-dimensional linear space $(A^n, +, .)$ (shortly only A^n) over the field A. Thus the set A^n of all words of the length n of the alphabet a can be considered to be a n-dimensional linear space.

Definition 4.20. A code $\mathcal{K}$ is called **linear** (n, k)**-code**, if it is k-dimensional subspace of the linear space A^n, i. e., if $\dim(\mathcal{K}) = k$, and for arbitrary $\mathbf{a}, \mathbf{b} \in \mathcal{K}$ and arbitrary $c \in A$ it holds:

$$\mathbf{a} + \mathbf{b} \in \mathcal{K}, \qquad c.\mathbf{a} \in \mathcal{K}.$$

Since a linear (n, k)-code is k-dimensional linear space, it has to have a basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$ with k elements. Then every code word $\mathbf{a} \in \mathcal{K}$ has unique representation in the form:

$$\mathbf{a} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_k\mathbf{b}_k, \tag{4.44}$$

where $a_1, a_2, \dots, a_n$ are coordinates of the vector $\mathbf{a}$ in the basis $\mathbf{B}$. Since $|A| = p$ then in the place of every a_i p different numbers can stand what implies that there exists p^k different code words. Hence, a linear (n, k)-code has p^k code words.
Let $\phi : A^k \to A^n$ be a mapping defined by formula:

$$\forall (a_1a_2 \dots a_k) \in A^k \quad \phi(a_1a_2 \dots a_k) = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_k\mathbf{b}_k.$$

Then ϕ is one to one mapping $A^k \leftrightarrow \mathcal{K}$ and thus by definition 4.17 (page 105) ϕ is the encoding of information characters and the linear (n, k)-code $\mathcal{K}$ has k information characters and $n - k$ check characters.

We will often use an advantageous matrix notation in which vectors stand as matrices having one column or one row. Now we make an agreement that the words – i. e., vectors $\mathbf{a} \in A^n$ – will be always considered as **one-column matrices**, i. e., if the word $\mathbf{a} = a_1a_2 \dots a_k$ stands in the matrix notation we will suppose that

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_k \end{bmatrix} .$$

If vector **a** in the form of one-row matrix is needed, it will be written as the transposed matrix $\mathbf{a}^T$, i. e.,

$$\mathbf{a}^T = \begin{bmatrix} a_1 & a_2 & \dots & a_k \end{bmatrix} .$$

The scalar product of two vectors $\mathbf{u}$, $\mathbf{v} \in A^n$ can be considered to be a product of two matrices and can be written as $\mathbf{u}^T.\mathbf{v}$.

Definition 4.21. Let $\mathcal{K}$ be a linear (n, k)-code, let $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$ be an arbitrary basis of the code $\mathcal{K}$. Let $\mathbf{b}_i = (b_{i1}\, b_{i2} \dots b_{in})^T$ for $i = 1, 2, \dots, k$. Then the matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \dots \\ \mathbf{b}_k^T \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{kn} \end{bmatrix} \tag{4.45}$$

of the type $(k \times n)$ is called **generating matrix of the code $\mathcal{K}$.**

Remark. By definition 4.21 every matrix **G** for which

a) every row is a code word,

b) rows are linearly independent vectors, i. e., the rank of **G** equals to k,

c) every code word is a linear combination of rows of **G**,

is a generating matrix of the code $\mathcal{K}$.

If the matrix $\mathbf{G}'$ originated from a generating matrix **G** of a linear code $\mathcal{K}$ by several equivalent row operations (row switching, row multiplication by a non zero constant and row addition) then the matrix $\mathbf{G}'$ is also a generating matrix of $\mathcal{K}$.

Remark. Let (4.45) be the generating matrix of a linear (n, k)-code for the basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$. If $u_1, u_2, \dots, u_k$ are the coordinates of the word $\mathbf{a} = a_1 a_2 \dots a_n$ in the basis **B** then

$$\mathbf{a}^T = u_1\mathbf{b}_1^T + u_2\mathbf{b}_2^T + \dots + u_k\mathbf{b}_k^T = \begin{bmatrix} u_1 & u_2 & \dots & u_k \end{bmatrix} . \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \dots \\ \mathbf{b}_k^T \end{bmatrix} ,$$

or more detailed:

$$\begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \dots & u_k \end{bmatrix} . \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{kn} \end{bmatrix} ,$$

or shortly:

$$\mathbf{a}^T = \mathbf{u}^T.\mathbf{G} \ .$$

Example 4.29. Several linear codes.

a) Binary code of the length 4 with parity check – linear $(4,3)$-code:
$\mathcal{K} \subset A^4$, $A = \{0,1\}$: 0000, 0011, 0101, 0110
1001, 1010, 1100, 1111
Basis: $B = \{0011, 0101, 1001\}$.

$$\text{Generating matrix} \quad \mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

b) Ternary repeating code of the length 5 – linear $(5,1)$-code:
$\mathcal{K} \subset A^5$, $A = \{0,1,2\}$: 00000, 11111, 22222
Basis: $\{11111\}$.

$$\text{Generating matrix} \quad \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

c) Binary doubling code of the length 6 – linear $(6,3)$-code:
$\mathcal{K} \subset A^6$, $A = \{0,1\}$: 000000, 000011, 001100, 001111
110000, 110011, 111100, 111111
Basis: $\{000011, 001100, 110000\}$.

$$\text{Generating matrix} \quad \mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

d) Decimal code of the length n with check digit modulo 10 is not a linear code, since a finite field with 10 elements does not exist.

Definition 4.22. We say that two block codes $\mathcal{K}$, $\mathcal{K}'$ of the length n are **equivalent** if there exists a permutation π of the set $\{1, 2, \dots, n\}$ such that it holds

$$\forall a_1 a_2 \dots a_n \in A^n \quad a_1 a_2 \dots a_n \in \mathcal{K} \quad \text{if and only if} \quad a_{\pi[1]} a_{\pi[2]} \dots a_{\pi[n]} \in \mathcal{K}' \ .$$

By definition 4.18 (page 106) a block code $\mathcal{K}$ with k information characters and $n-k$ check characters is systematic if for every $a_1 a_2 \dots a_k \in A^k$ there exists exactly one code word $\mathbf{a} \in \mathcal{K}$ with the prefix $a_1 a_2 \dots a_k \in A^k$. We have shown that a linear (n, k)-code is a code with k information characters and with $n-k$ check characters, but it do not need to be systematic. Doubling code is a linear $(n = 2k, k)$-code which is not systematic if $k > 1$. It suffices to change the order of characters in the code word $a_1 a_2 \dots a_n$ – first the characters on odd positions and then the characters on even positions, and the new code is systematic. Similar procedure can be made with any linear (n, k)-code.

Theorem 4.16. *A linear (n, k)-code $\mathcal{K}$ is systematic if and only if there exists a generating matrix $\mathbf{G}$ of $\mathcal{K}$ of the type:*

$$\mathbf{G} = \left[\ \mathbf{E}\ \middle|\ \mathbf{B}\ \right] = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & h_{1n-k} \\ 0 & 1 & 0 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n-k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & b_{k1} & h_{k2} & \dots & h_{kn-k} \end{bmatrix} . \tag{4.46}$$

Proof. Let (4.46) be the generating matrix of $\mathcal{K}$. Let $\mathbf{u} = u_1, u_2, \dots u_k$ are the coordinates of the word $\mathbf{a} = a_1 a_2 \dots a_n \in \mathcal{K}$ in the basis containing the rows of the generating matrix $\mathbf{G}$. Then by remark following the definition 4.21 $\mathbf{a}^T = \mathbf{b}^T.\mathbf{G}$. Specially for $\mathbf{u} = a_1 a_2 \dots a_k$ it holds:

$$\mathbf{u}^T.\mathbf{G} = \begin{bmatrix} a_1 & a_2 & \dots & a_k \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1n-k} \\ 0 & 1 & 0 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n-k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kn-k} \end{bmatrix} =$$

$$= \begin{bmatrix} a_1 & a_2 & \dots & a_k & v_{k+1} & \dots & v_n \end{bmatrix},$$

where v_{k+i} is uniquely defined by the equation:

$$v_{k+i} = \begin{bmatrix} a_1 & a_2 & \dots & a_k \end{bmatrix} . \begin{bmatrix} b_{1i} \\ b_{2i} \\ \dots \\ b_{ki} \end{bmatrix} .$$

For every $a_1a_2\dots a_k \in A^k$ there exists exactly one code word of the code $\mathcal{K}$ with the prefix $a_1a_2\dots a_k$. Hence the code $\mathcal{K}$ is systematic.

Let the code $\mathcal{K}$ is systematic. If the first k rows of the generating matrix $\mathbf{G}$ of $\mathcal{K}$ are linearly independent we can obtain from $\mathbf{G}$ by means of equivalent row operations an equivalent matrix $\mathbf{G}'$ in the form $\mathbf{G}' = \left[\begin{array}{c|c} \mathbf{E} & \mathbf{B} \end{array}\right]$ which is also a generating matrix of the code $\mathcal{K}$.
If the first k rows of generating matrix $\mathbf{G}$ of $\mathcal{K}$ are not linearly independent, then $\mathbf{G}$ can be converted by means of equivalent row operations to the form:

$$\mathbf{G}' = \left[\begin{array}{cccc|cccc} d_{11} & d_{12} & \dots & d_{1k} & d_{1(k+1)} & d_{1(k+2)} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2k} & d_{2(k+1)} & d_{2(k+2)} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ d_{(k-1)1} & d_{(k-1)2} & \dots & d_{(k-1)k} & d_{(k-1)(k+1)} & d_{(k-1)(k+2)} & \dots & d_{(k-1)n} \\ 0 & 0 & \dots & 0 & d_{k(k+1)} & d_{k(k+2)} & \dots & d_{kn} \end{array}\right].$$

The rank of the matrix $\mathbf{G}'$ equals to k since it is equivalent to the matrix $\mathbf{G}$ which has k linearly independent rows. For $\mathbf{u}$, $\mathbf{v} \in A^k$ such that $\mathbf{u} \neq \mathbf{v}$ it holds $\mathbf{u}^T.\mathbf{G}' \neq \mathbf{v}^T.\mathbf{G}'$. Both $\mathbf{u}^T.\mathbf{G}'$ and $\mathbf{v}^T.\mathbf{G}'$ are code words. Notice that the first k coordinates of the vector $\mathbf{u}^T.\mathbf{G}$ do not depend on the k-th coordinate of the vector $\mathbf{u}$ what implies that there are several code words of the code $\mathcal{K}$ with the same prefix – the code $\mathcal{K}$ is not systematic. The assumption that the first k columns of generating matrix are not independent leads to the contradiction. ■

Corollary. A linear (n, k)-code $\mathcal{K}$ is systematic if and only if the first k rows of its generating matrix $\mathbf{G}$ are linearly independent.

Theorem 4.17. *Every linear (n, k)-code $\mathcal{K}$ is equivalent to some systematic linear code.*

Proof. Let $\mathbf{G}$ be a generating matrix of a linear (n, k)-code $\mathcal{K}$. The matrix $\mathbf{G}$ has k linearly independent rows and hence it has to have at least one k-tuple of linearly independent columns. If the first k columns are independent, the code $\mathcal{K}$ is systematic by the corollary of the theorem 4.16.
If the first k columns are not linearly independent, we can make such permutation π of columns in $\mathbf{G}$ so that in the permutated matrix, the first k columns are linearly independent

Then the corresponding code $\mathcal{K}'$ obtained by the same permutation π of characters in code words of $\mathcal{K}$ is systematic. ■

There exists another way of characterization of a linear (n, k)-code. This method specifies the properties of code words by an equation which the code words have to satisfy. So the binary block code of the length n with even parity check character can be defined by the equation:

$$x_1 + x_2 + \cdots + x_n = 0$$

The doubling code of the length $n = 2k$ is characterized by the system of equations:

$$\begin{aligned} x_1 - x_2 &= 0 \\ x_3 - x_4 &= 0 \\ &\dots \\ x_{2i-1} - x_{2i} &= 0 \\ &\dots \\ x_{n-1} - x_n &= 0 \end{aligned}$$

And here is the system of equation for a repeating code of the length n:

$$\begin{aligned} x_1 - x_2 &= 0 \\ x_1 - x_3 &= 0 \\ &\dots \\ x_1 - x_n &= 0 \end{aligned}$$

Definition 4.23. Check matrix of the linear code $\mathcal{K}$ is such matrix $\mathbf{H}$ of elements of code alphabet A for which it holds: The word $\mathbf{v} = v_1 v_2 \dots v_n$ is the code word if and only if:

$$\mathbf{H}.\mathbf{v} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{bmatrix} . \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{o}\ . \tag{4.47}$$

Shortly: $\mathbf{v} \in \mathcal{K}$ if and only if $\mathbf{H}.\mathbf{v} = \mathbf{o}$.

Suppose we are given a linear (n, k)-code $\mathcal{K}$ with generating matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \dots \\ \mathbf{b}_k^T \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{kn} \end{bmatrix} \tag{4.48}$$

of the type $(k \times n)$. What is the check matrix of the code $\mathcal{K}$, i. e., the matrix $\mathbf{H}$ such that $\mathbf{H}.\mathbf{u} = \mathbf{o}$ if and only if $\mathbf{u} \in \mathcal{K}$?
The first visible property of the matrix $\mathbf{H}$ is that it should have n columns in order $\mathbf{H}.\mathbf{u}$ was defined for $\mathbf{u} \in A^n$.
The set of all $\mathbf{u} \in A^n$ such that $\mathbf{H}.\mathbf{u} = \mathbf{o}$ is a subspace of the space A^n with dimension equal to $n - \text{rank}(\mathbf{H}) = \dim(\mathcal{K}) = k$, from where $\text{rank}(\mathbf{H}) = n - k$. Hence it suffices to search the check matrix $\mathbf{H}$ as a matrix of the type $((n-k)\times n)$ with $n - k$ linearly independent rows. Let $\mathbf{h}^T$ is arbitrary row of the matrix $\mathbf{H}$. Then every code word $\mathbf{u} \in \mathcal{K}$ has to satisfy:

$$\mathbf{u}^T.\mathbf{h} = u_1h_1 + u_2h_2 + \cdots + u_nh_n = 0 \ . \tag{4.49}$$

We could write out the system of $p^k = |\mathcal{K}|$ linear equations of the type (4.49), one for every code word $\mathbf{u} \in \mathcal{K}$. Such system of equation would contain too much linearly dependent equations. Suppose that (4.49) holds for all vectors of a basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$ of the subspace $\mathcal{K}$. Then (4.49) has to hold for all vectors of the linear subspace $\mathcal{K}$. That is why it suffices to solve the following system of equations:

$$\left.\begin{array}{rcl} \mathbf{b}_1^T.\mathbf{h} & = & 0 \\ \mathbf{b}_2^T.\mathbf{h} & = & 0 \\ \ldots & & \\ \mathbf{b}_k^T.\mathbf{h} & = & 0 \end{array}\right\} ,$$

in matrix notation:

$$\mathbf{G}.\mathbf{h} = \mathbf{o} \ , \tag{4.50}$$

where $\mathbf{G}$ is the generating matrix with rows $\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_k^T$.
Since the rank of matrix $\mathbf{G}$ is k, the set of all solutions of the system (4.50) is a subspace with dimension $(n - k)$ and that is why it is possible to find $(n - k)$ linearly independent solutions $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n-k}$ of the system (4.50) which will be the rows of required check matrix $\mathbf{H}$, i. e.,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \ldots \\ \mathbf{h}_{n-k}^T \end{bmatrix} .$$

Note that

$$\mathbf{G}.\mathbf{H}^T = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \ldots \\ \mathbf{b}_k^T \end{bmatrix}_{k\times n} \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_{n-k} \end{bmatrix}_{n\times(n-k)} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{k\times(n-k)} .$$

Let us have a matrix $\mathbf{H}$ of the type $((n-k)\times n)$, let $\operatorname{rank}(\mathbf{H}) = (n-k)$ and let $\mathbf{G}.\mathbf{H}^T = \mathbf{O}_{k\times(n-k)}$, where $\mathbf{O}_{k\times(n-k)}$ is the null matrix of the type $(k\times(n-k))$. Denote by $\mathcal{N}\subseteq A^n$ the linear subspace of all solutions of the equation $\mathbf{Hu} = \mathbf{o}$. Since for all vectors of the basis of the code $\mathbf{K}$ is $\mathbf{H}.\mathbf{b}_i = \mathbf{o}$, $i = 1, 2, \dots k$, the same holds for arbitrary code word $\mathbf{u}\in\mathcal{K}$, $\mathbf{u} = \sum_{i=1}^{k} u_i\mathbf{b}_i$:

$$\mathbf{H}.\mathbf{u} = \mathbf{H}.\sum_{i=1}^{k} u_i\mathbf{b}_i = \sum_{i=1}^{k}\mathbf{H}.(u_i\mathbf{b}_i) = \sum_{i=1}^{k} u_i(\mathbf{H}.\mathbf{b}_i) = \sum_{i=1}^{k} u_i.\mathbf{o} = \mathbf{o}\ .$$

We have just proven $\mathcal{K}\subseteq\mathcal{N}$.
Since $\operatorname{rank}(\mathbf{H}) = (n-k)$, $\dim(\mathcal{N}) = n - \operatorname{rank}(\mathbf{H}) = k$. Since $\mathcal{K}\subseteq\mathcal{N}$, the basis $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ is the basis of the subspace $\mathcal{N}$ and hence $\mathcal{K} = \mathcal{N}$.

Now we can formulate these proven facts in the following theorem.

Theorem 4.18. *Let $\mathcal{K}$ is a linear (n,k)-code with a generating matrix $\mathbf{G}$ of the type $(k\times n)$. Then the matrix $\mathbf{H}$ of the type $((n-k)\times n)$ is the check matrix of the code $\mathcal{K}$ if and only if*

$$\dim(\mathbf{H}) = (n-k) \quad a \quad \mathbf{G}.\mathbf{H}^T = \mathbf{O}_{k\times(n-k)}, \tag{4.51}$$

where $\mathbf{O}_{k\times(n-k)}$ is the null matrix of the type $(k\times(n-k))$.

The situation is much more simple for systematic codes as the next theorem says.

Theorem 4.19. *A linear (n,k)-code $\mathcal{K}$ with the generating matrix of the type $\mathbf{G} = \left[\ \mathbf{E}_{k\times k}\ \middle|\ \mathbf{B}\ \right]$ has the check matrix $\mathbf{H} = \left[\ -\mathbf{B}^T\ \middle|\ \mathbf{E}_{(n-k)\times(n-k)}\ \right]$.*

Proof. Denote $m = n-k$. Then we can write:

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \dots \\ \mathbf{b}_p^T \\ \dots \\ \mathbf{b}_k^T \end{bmatrix} = \left[\begin{array}{cccccc|cccccc} 1 & 0 & \dots & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1q} & \dots & b_{1m} \\ 0 & 1 & \dots & 0 & \dots & 0 & b_{21} & b_{22} & \dots & b_{1q} & \dots & b_{2m} \\ \dots & & & & & & & & & & & \\ 0 & 0 & \dots & 1 & \dots & 0 & b_{p1} & b_{p2} & \dots & b_{pq} & \dots & b_{pm} \\ \dots & & & & & & & & & & & \\ 0 & 0 & \dots & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kq} & \dots & b_{km} \end{array}\right],$$

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}_1^T\\ \mathbf{h}_2^T\\ \dots\\ \mathbf{h}_q^T\\ \dots\\ \mathbf{h}_m^T\end{bmatrix}=\left[\begin{array}{cccccc|cccccc} -b_{11} & -b_{21} & \dots & -b_{p1} & \dots & -b_{k1} & 1 & 0 & \dots & 0 & \dots & 0\\ -b_{12} & -b_{22} & \dots & -b_{p2} & \dots & -b_{k2} & 0 & 1 & \dots & 0 & \dots & 0\\ \dots & & & & & & & & & & & \\ -b_{1q} & -b_{2q} & \dots & -b_{pq} & \dots & -b_{kq} & 0 & 0 & \dots & 1 & \dots & 0\\ \dots & & & & & & & & & & & \\ -b_{1m} & -b_{2m} & \dots & -b_{pm} & \dots & -b_{km} & 0 & 0 & \dots & 0 & \dots & 1\end{array}\right].$$

It holds for $\mathbf{b}_p$, $\mathbf{h}_q$:

$$\begin{array}{lcl} \mathbf{b}_p^T & = & [\begin{array}{cccccccccccc} 0 & 0 & \dots & 1 & \dots & 0 & b_{p1} & b_{p2} & \dots & b_{pq} & \dots & b_{pm}\end{array}]\\ \mathbf{h}_q^T & = & [\begin{array}{cccccccccccc} -b_{1q} & -b_{2q} & \dots & -b_{pq} & \dots & -b_{kq} & 0 & 0 & \dots & 1 & \dots & 0\end{array}]\end{array}$$

and that is why $\mathbf{b}_p^T.\mathbf{h}_q=(-b_{pq}+b_{pq})=0$ for every $p,\ q\in\{1,2,\dots,n\}$ what implies

$$\mathbf{G}.\mathbf{H}^T=\mathbf{O}_{k\times(n-k)}.$$

Since the matrix $\mathbf{H}$ with $m=n-k$ rows contains the submatrix $\mathbf{E}_{(n-k)\times(n-k)}$ it holds $\text{rank}(H)=n-k$. The matrix $\mathbf{H}$ is by theorem 4.18 the check matrix of the code $\mathcal{K}$.

Definition 4.24. Let $\mathcal{K}\subseteq A^n$ be a linear (n,k)-code. The **dual code** $\mathcal{K}^\perp$ of the code $\mathcal{K}$ is defined by equation:

$$\mathcal{K}^\perp=\{\mathbf{v}\mid \mathbf{a}.\mathbf{v}=0\ \forall\mathbf{a}\in\mathcal{K}\}.$$

Theorem 4.20. *Let $\mathcal{K}\subseteq A^n$ be a linear (n,k)-code with the generating matrix $\mathbf{G}$ and the check matrix $\mathbf{H}$. Then the dual code $\mathcal{K}^\perp$ is a linear $(n,n-k)$-code with the generating matrix $\mathbf{H}$ and the check matrix $\mathbf{G}$.*

Proof. It holds $\mathbf{v}\in\mathcal{K}^\perp$ if and only if

$$\mathbf{G}.\mathbf{v}=\mathbf{o}. \tag{4.52}$$

Since $\mathcal{K}^\perp$ is the set of all solutions of the equation (4.52) and $\text{rank}(\mathbf{G})=k$, $\mathcal{K}^\perp$ is a $(n-k)$-dimensional subspace of A^n – i. e., it is a linear $(n,(n-k))$-code with check matrix $\mathbf{G}$.

Since $\mathbf{H}.\mathbf{G}^T=\big((\mathbf{G}^T)^T.\mathbf{H}^T\big)^T=\big(\mathbf{G}.\mathbf{H}^T\big)^T=\mathbf{O}^T_{k\times(n-k)}=\mathbf{O}_{(n-k)\times k}$, every row of the matrix $\mathbf{H}$ is orthogonal to the subspace $\mathcal{K}$ and hence it is a code word of the code $\mathcal{K}^\perp$. Since $\text{rank}(\mathbf{H})=(n-k)$, the set of rows of the matrix $\mathbf{H}$ is the basis of the whole subspace $\mathcal{K}^\perp$, i. e., matrix $\mathbf{H}$ is the generating matrix of the code $\mathcal{K}^\perp$.

Example 4.30. The dual code of the binary repeating code $\mathcal{K}$ of the length 5 is the code containing all binary words $v_1v_2\dots v_n$ such that

$$v_1 + v_2 + v_3 + v_4 + v_5 = 0.$$

The code $\mathcal{K}^\perp$ is the code with even parity check.

Example 4.31. The dual code of the binary doubling code $\mathcal{K}$ is $\mathcal{K}$, hence $\mathcal{K}^\perp = \mathcal{K}$. The generating matrix of $\mathcal{K}$ is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

It is easy to show that $\mathbf{G}.\mathbf{G}^T = \mathbf{O}_{3\times 3}$ – the generating matrix of code $\mathcal{K}$ is also its check matrix.

4.13 Linear codes and error detecting

In definition 4.7 (page 84) we have defined error detection as follows: the code $\mathcal{K}$ detects t-tuple simple errors, if for every code word $\mathbf{u}$ and every word $\mathbf{w}$ such that $0 < d(\mathbf{u}, \mathbf{w}) \leq t$, the word $\mathbf{w}$ is a non code word. Theory of linear codes offers the way of modelling mechanism of error origination as the addition of an **error word** $\mathbf{e} = e_1e_2\dots e_n$ to the transmitted word $\mathbf{v} = v_1v_2\dots v_n$. Then we receive the word $\mathbf{w} = w_1w_2\dots w_n = \mathbf{v} + \mathbf{e}$ instead of transmitted word $\mathbf{v}$.

Definition 4.25. We say that the linear code $\mathcal{K}$ **detects error word e**, if $\mathbf{v} + \mathbf{e}$ is a non code word for every code word $\mathbf{v}$.

Definition 4.26. Hamming weight $\|\mathbf{a}\|$ of the word $\mathbf{a} \in A^n$ is the number of non zero characters of the word $\mathbf{a}$.

Theorem 4.21. *All code words of binary linear code $\mathcal{K}$ have even Hamming weight, or the number of words of even Hamming weight of $\mathcal{K}$ equals to the number of words of $\mathcal{K}$ of odd Hamming weight.*

Proof. Let $\mathbf{v}$ be a code word of odd Hamming weight. Define a mapping $f : \mathcal{K} \to \mathcal{K}$ by the following formula

$$f(\mathbf{w}) = \mathbf{w} + \mathbf{v}\ .$$

The mapping f is one to one mapping assigning to every word of even weight a word of odd weight and vice versa. Therefore, the number of words of even Hamming weight of $\mathcal{K}$ equals to the number of words of $\mathcal{K}$ of odd Hamming weight. ■

Note that a linear code $\mathcal{K}$ detects t-tuple simple errors if and only if it detects all error words having Hamming weight less of equal to t.

The minimum distance of a code $\Delta(\mathcal{K})$ has crucial importance for error detection and error correction. $\Delta(\mathcal{K})$ was defined by definition 4.7 (page 84) as the minimum of Hamming distances of all pairs of different code words of the code $\mathcal{K}$. Denote $d = \Delta(\mathcal{K})$. Then the code $\mathcal{K}$ detects all $(d-1)$-tuple errors and corrects all t-tuple errors for $t < \dfrac{d}{2}$ (see the theorem 4.12 – page 103).

A linear code $\mathcal{K}$ allows even simpler calculation of the minimal distance $\Delta(\mathcal{K})$ of $\mathcal{K}$.

Theorem 4.22. *Let $\mathcal{K}$ be a linear code. The minimum distance $\Delta(\mathcal{K})$ of $\mathcal{K}$ equals to the minimum of Hamming weights of all non zero code words of $\mathcal{K}$, i. e.,*

$$\Delta(\mathcal{K}) = \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}}\{\|\mathbf{u}\|\} \ .$$

Proof.
1. Let $\mathbf{u},\ \mathbf{v} \in \mathcal{K}$ be two code words such that $d(\mathbf{u},\mathbf{v}) = \Delta(\mathcal{K})$. Let $\mathbf{w} = \mathbf{u} - \mathbf{v}$. The word $\mathbf{w}$ has exactly as many characters different from zero character, as in how many positions the words $\mathbf{u},\ \mathbf{v}$ differ. Therefore:

$$\min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}}\{\|\mathbf{u}\|\} \leq \|\mathbf{w}\| = d(\mathbf{u},\mathbf{v}) = \Delta(\mathcal{K}) \ . \tag{4.53}$$

2. Let $\mathbf{w} \in \mathcal{K}$ be a code word such that $\|\mathbf{w}\| = \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}}\{\|\mathbf{u}\|\}$. Then:

$$\Delta(\mathcal{K}) \leq d(\mathbf{o},\mathbf{v}) = \|\mathbf{w}\| \leq \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}}\{\|\mathbf{u}\|\} \ . \tag{4.54}$$

The desired assertion of the theorem follows from (4.53) and (4.54). ■

Definition 4.27. Let $\mathbf{H}$ be the check matrix of a linear code $\mathcal{K}$, let $\mathbf{v} = v_1v_2\dots v_n \in A^n$ be an arbitrary word of the length n of alphabet A. **Syndrome of the word v** is a word $\mathbf{s} = s_1s_2\dots s_n$ satisfying the equation:

$$\mathbf{H}.\begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \dots \\ s_n \end{bmatrix}, \qquad \text{shortly} \quad \mathbf{H}.\mathbf{v} = \mathbf{s} \ .$$

Having received a word $\mathbf{w}$ we can calculate its syndrome $\mathbf{s} = \mathbf{Hw}$. If $\mathbf{s} \neq \mathbf{o}$ we know that an error occurred. Moreover, we know that the syndrome of the received word $\mathbf{w} = \mathbf{v} + \mathbf{e}$ (where $\mathbf{v}$ was the transmitted code word) is the same as the symbol of the error word $\mathbf{e}$ since

$$\mathbf{Hw} = \mathbf{H}(\mathbf{v} + \mathbf{e}) = \mathbf{Hv} + \mathbf{He} = \mathbf{o} + \mathbf{He} = \mathbf{He}\ .$$

Since the code $\mathcal{K}$ is the subspace of all solutions of the equation $\mathbf{H} = \mathbf{o}$, every solution of the equation $\mathbf{He} = \mathbf{s}$ is in the form $\mathbf{e} + \mathbf{k}$ where $\mathbf{k} \in \mathcal{K}$. The set of all words of this form will be denoted by $\mathbf{e} + \mathcal{K}$, i. e.:

$$\mathbf{e} + \mathcal{K} = \{\mathbf{w} \mid \mathbf{w} = \mathbf{e} + \mathbf{k},\quad \mathbf{k} \in \mathcal{K}\}.$$

Theorem 4.23. *Let $\mathcal{K}$ be a linear code with the check matrix $\mathbf{H}$ and minimum distance $\Delta(\mathcal{K})$. Let d be the minimum of the number of linearly dependent columns*[5] *of the check matrix $\mathbf{H}$. Then:*

$$d = \Delta(\mathcal{K})\ .$$

Proof. According to the theorem 4.22 $\Delta(\mathcal{K})$ equals to the minimum weight of non zero code words. Let d be the minimum number of linearly dependent columns of check matrix $\mathbf{H}$.

Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n$ are the columns of the check matrix $\mathbf{H}$, i. e.,

$$\mathbf{H} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_n \end{bmatrix}.$$

Denote by $\mathbf{u} \in \mathcal{K}$ the non zero code word with the minimum Hamming weight $\|\mathbf{u}\| = t$. The word $\mathbf{u}$ has characters $u_{i_1}, u_{i_2}, \dots, u_{i_t}$ on the positions $i_1, i_2, \dots, i_t$ and the character 0 on other positions, i. e.,

$$\mathbf{u}^T = \begin{bmatrix} 0 & 0 & \dots & 0 & u_{i_1} & 0 & \dots & 0 & u_{i_2} & 0 & \dots & \dots & 0 & u_{i_t} & 0 & \dots & 0 & 0 \end{bmatrix}.$$

The word $\mathbf{u}$ is the code word, that is why $\mathbf{Hv} = \mathbf{o}$, i. e.:

$$\mathbf{Hu} = \sum_{i=1}^{n} u_i.\mathbf{c}_i = u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2} + \dots + u_{i_t}\mathbf{u}_{i_t} = \mathbf{o}\ . \tag{4.55}$$

[5] Let d be such number that in the check matrix $\mathbf{H}$ there exist d linearly dependent columns but every $(d-1)$-tuple of columns of $\mathbf{H}$ is the set of linearly independent columns.

Since all coefficients u_{i_j} are non zero characters, the columns $\mathbf{c}_{i_1}, \mathbf{c}_{i_2}, \dots, \mathbf{c}_{i_t}$ are linearly dependent. We have just proven:

$$d \leq \Delta(\mathcal{K}) \ . \tag{4.56}$$

Let us have d linearly dependent columns $\mathbf{c}_{i_1}, \mathbf{c}_{i_2}, \dots, \mathbf{c}_{i_d}$. Then there exist numbers $u_{i_1}, u_{i_2}, \dots, u_{i_d}$ such that at least one of them is different from zero and

$$u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2} + \cdots + u_{i_d}\mathbf{c}_{i_d} = \mathbf{o} \ .$$

Let us define the word $\mathbf{u}$ that has characters $u_{i_1}, u_{i_2}, \dots, u_{i_d}$ on positions $i_1, i_2, \dots, i_d$ and zero character on other positions, i. e.:

$$\mathbf{u}^T = \begin{bmatrix} 0 & 0 & \dots & 0 & u_{i_1} & 0 & \dots & 0 & u_{i_2} & 0 & \dots & \dots & 0 & u_{i_t} & 0 & \dots & 0 & 0 \end{bmatrix} .$$

Then

$$\mathbf{Hu} = \sum_{i=1}^{n} u_i.\mathbf{c}_i = u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2} + \cdots + u_{i_t}\mathbf{c}_{i_d} = \mathbf{o} \ , \tag{4.57}$$

and hence $\mathbf{u}$ is a non zero word with Hamming weight $\|\mathbf{u}\| \leq d$. We have proven:

$$\Delta(\mathcal{K}) \leq d \ .$$

The last inequality with (4.56) gives desired assertions of theorem. ■

Theorem 4.24. *A linear code detects t-tuple simple errors if and only if every t columns of the check matrix of $\mathcal{K}$ are linearly independent.*

Proof. Denote $d = \Delta(\mathcal{K})$. By the last theorem 4.23 there exist d linearly dependent columns in check matrix $\mathbf{H}$ of $\mathcal{K}$ but for $t < d$ every t columns are linearly independent.

If the code $\mathcal{K}$ detects t-tuple errors then $t < d$ and (by theorem 4.23) every t columns of $\mathbf{H}$ are linearly independent.

If every t columns of check matrix $\mathbf{H}$ are linearly independent (again by theorem 4.23), it holds $t < d$ and that is why the code $\mathcal{K}$ detects t errors. ■

4.14 Standard code decoding

In previous section, we have shown how to determine the maximum number t of errors which a linear code $\mathcal{K}$ is capable to detect, and how to decide whether the received word was transmitted without errors or not – of course provided that the number of errors is not greater than t.

Having received a non code word $\mathbf{w}$ we would like to assign it the code word $\mathbf{v}$ which was probably transmitted and from which the received word $\mathbf{w}$ originated by effecting several errors – again provided that the number of errors occurred is limited to some small number. For this purpose the decoding δ of the code $\mathcal{K}$ was defined (see section 4.10, definition 4.16, page. 105) as a function whose codomain is a subset of A^n, contains $\mathcal{K}$ and which assigns to every word from its codomain a code word, and which is identity on $\mathcal{K}$ (for all $\mathbf{v} \in \mathcal{K}$ it holds $\delta(\mathbf{v}) = \mathbf{v}$).

If the word $\mathbf{v}$ was transmitted and errors represented by the error word $\mathbf{e}$ occurred, we receive the word $\mathbf{e} + \mathbf{v}$. If $\delta(\mathbf{e} + \mathbf{v}) = \mathbf{v}$ we have decoded correctly.

Definition 4.28. We say that a linear code $\mathcal{K}$ with decoding δ **corrects the error word e** if for all $\mathbf{v} \in \mathcal{K}$ it holds:

$$\delta(\mathbf{e} + \mathbf{v}) = \mathbf{v} \ .$$

Definition 4.29. Let $\mathcal{K} \subseteq A^n$ be a linear code with code alphabet A. Let us define for every $\mathbf{e} \in A^n$:

$$\mathbf{e} + \mathcal{K} = \{\mathbf{e} + \mathbf{v} \mid \mathbf{v} \in \mathcal{K}\} \ .$$

The set $\mathbf{e} + \mathcal{K}$ is called **class of the word e according to the code $\mathcal{K}$**.

Theorem 4.25. *Let $\mathcal{K} \subseteq A^n$ be a linear (n, k)-code with code alphabet A, $|A| = p$. For arbitrary words $\mathbf{e}$, $\mathbf{e}' \in A^n$ it holds:*

(i) If $\mathbf{e} - \mathbf{e}'$ is a code word then $\mathbf{e} + \mathcal{K} = \mathbf{e}' + \mathcal{K}$.

(ii) If $\mathbf{e} - \mathbf{e}'$ is not a code word then $\mathbf{e} + \mathcal{K}$, $\mathbf{e}' + \mathcal{K}$ are disjoint.

(iii) The number of words of every class is equal to the number of all code words, i. e., $|\mathbf{e} + \mathcal{K}| = |\mathcal{K}| = p^k$ and the number of all classes is p^{n-k}.

Proof.
(i) Let $(\mathbf{e}-\mathbf{e}')\in\mathcal{K}$.
Let $\mathbf{v}\in\mathcal{K}$, and hence $(\mathbf{e}+\mathbf{v})\in(\mathbf{e}+\mathcal{K})$. Set $\mathbf{u}=\mathbf{v}+(\mathbf{e}-\mathbf{e}')$. $\mathcal{K}$ is a linear space and $(\mathbf{e}-\mathbf{e}')\in\mathcal{K}$. Therefore $\mathbf{u}\in\mathcal{K}$ what implies $(\mathbf{e}'+\mathbf{u})\in(\mathbf{e}'+\mathcal{K})$. Now we can write $\mathbf{e}'+\mathbf{u}=\mathbf{e}'+\mathbf{v}+(\mathbf{e}-\mathbf{e}')=\mathbf{e}+\mathbf{v}$. That is why $(\mathbf{e}+\mathbf{v})\in(\mathbf{e}'+\mathcal{K})$. We have shown that $(\mathbf{e}+\mathcal{K})\subseteq(\mathbf{e}'+\mathcal{K})$. The reverse inclusion can be shown analogically. Thus $(\mathbf{e}+\mathcal{K})=(\mathbf{e}'+\mathcal{K})$.

(ii) Let $(\mathbf{e}-\mathbf{e}')\notin\mathcal{K}$.
Suppose that there is a word $\mathbf{w}\in(\mathbf{e}+\mathcal{K})\cap(\mathbf{e}'+\mathcal{K})$. Then

$$\begin{aligned}\mathbf{w}&=\mathbf{e}+\mathbf{v}\ ,\\ \mathbf{w}&=\mathbf{e}'+\mathbf{v}',\end{aligned}$$

for some code words $\mathbf{v}$, $\mathbf{v}'\in\mathcal{K}$. From two last equations it follows $\mathbf{e}+\mathbf{v}=\mathbf{e}'+\mathbf{v}'$ and further $\mathbf{e}-\mathbf{e}'=\mathbf{v}'-\mathbf{v}\in\mathcal{K}$ (since both words $\mathbf{v}$, $\mathbf{v}'$ are vectors of linear space $\mathcal{K}$) which is in contradiction with assumption of (ii).

(iii) We have shown that a linear (n,k)-code with p-character code alphabet has p^k code words (see the text following definition 4.20, page 112). We want to show that $|\mathbf{e}+\mathcal{K}|=|\mathcal{K}|=p^k$. It suffices to show that if $\mathbf{u},\ \mathbf{w}\in\mathcal{K}$, $\mathbf{u}\neq\mathbf{w}$ then $\mathbf{e}+\mathbf{u}\neq\mathbf{e}+\mathbf{w}$. If $\mathbf{e}+\mathbf{u}=\mathbf{e}+\mathbf{w}$ then (after subtracting $\mathbf{e}$ from both sides of the equation) $\mathbf{u}=\mathbf{w}$. Therefore, all classes of words according to the code $\mathcal{K}$ have the same number of elements p^k.
Since the union of all clases of words according to the code $\mathcal{K}$ is A^n and $|A^n|=p^k$, the number of all classes according to the code $\mathcal{K}$ is equal to

$$\frac{|A^n|}{|\mathcal{K}|}=\frac{p^n}{p^k}=p^{n-k}.$$

■

Definition 4.30. Standard decoding of a linear code $\mathcal{K}$. Define a complete decoding $\delta:A^n\to\mathcal{K}$ of a code $\mathcal{K}$ as follows: Choose one representative from every class according to the code $\mathcal{K}$ so that its weight is minimal in its class. (The choice does not need to be unique – several words with the same minimum weight can exist in one class.) Then every received word $\mathbf{w}\in A^n$ is decoded as $\mathbf{v}=\mathbf{w}-\mathbf{e}$ where error word $\mathbf{e}$ is the representative of the class of the word $\mathbf{w}$:

$$\delta(\mathbf{w})=\mathbf{w}-[\text{representative of the class }(\mathbf{w}+\mathcal{K})].$$

Example 4.32. Binary $(4,3)$-code $\mathcal{K}$ of even parity has two classes:

$$\begin{array}{lllllllll} 0000+\mathcal{K} & =\{0000 & 0011 & 0101 & 0110 & 1001 & 1010 & 1100 & 1111\} \\ 0001+\mathcal{K} & =\{0001 & 0010 & 0100 & 0111 & 1000 & 1011 & 1101 & 1110\} \end{array}$$

The class $0000+\mathcal{K}$ has an unique representative – the word 0000. The class $0001+\mathcal{K}$ can have as the representative an arbitrary word from the following words 0001, 0010, 0100, 1000. According to our choice of representative the standard decoding corrects one simple error on the forth, third, second or first position of the received word.

If the error occurs on other places the standard decoding does not decode correctly. This is not a surprising discovery for us since we know that the minimum distance of the even parity code is 2 and hence it cannot correct all single simple errors.

Theorem 4.26. *Standard decoding δ corrects exactly those error words that are representatives of classes, i. e.,*

$$\delta(\mathbf{v}+\mathbf{e}) = \mathbf{v} \quad \textit{for all } \mathbf{v} \in \mathcal{K}$$

if and only if the error word $\mathbf{e}$ is the representative of some class according to the code $\mathcal{K}$.

Proof. If the word $\mathbf{e}$ is the representative of its class and $\mathbf{v} \in \mathcal{K}$ then the word $\mathbf{v}+\mathbf{e}$ is an element of the class $\mathbf{e}+\mathcal{K}$. By definition of standard decoding $\delta(\mathbf{e}+\mathbf{v}) = \mathbf{e}+\mathbf{v}-\mathbf{e} = \mathbf{v}$ – standard decoding δ corrects the error word $\mathbf{e}$ (see definition 4.28).

Let the word $\mathbf{e}'$ is not the representative of its class whose representative is the word $\mathbf{e} \neq \mathbf{e}'$. It holds $(\mathbf{e}-\mathbf{e}') \in \mathcal{K}$. Let $\mathbf{v} \in \mathcal{K}$, then the word $\mathbf{v}+\mathbf{e}'$ is an element of the class $\mathbf{e}+\mathcal{K}$ and is decoded as $\delta(\mathbf{v}+\mathbf{e}') = \mathbf{v}+\mathbf{e}'-\mathbf{e} \neq \mathbf{v}$. If $\mathbf{e}'$ is not the representative of its class, the standard decoding does not correct the error word $\mathbf{e}'$. ■

Theorem 4.27. *Standard decoding δ is an optimal decoding in the following meaning: There exists no decoding δ^* such that δ^* corrects the same error words as δ, and moreover several another error words.*

Proof. Let $\mathbf{e}' \in (\mathbf{e} + \mathcal{K})$, let $\mathbf{e}$ be the representative of the class $\mathbf{e} + \mathcal{K}$, let $\mathbf{e} \neq \mathbf{e}'$. The word $\mathbf{v} = \mathbf{e}' - \mathbf{e}$ is a code word not equal to zero word $\mathbf{o}$. If an error specified by the error word $\mathbf{e}$ occurs after the word $\mathbf{v}$ was transmitted, the word $\mathbf{v} + \mathbf{e} = \mathbf{e}' - \mathbf{e} + \mathbf{e} = \mathbf{e}'$ is received. Since δ corrects all error words that are representatives of classes, it holds: $\delta(\mathbf{v} + \mathbf{e}) = \delta(\mathbf{e}') = \mathbf{v}$. Decoding δ^* corrects the same words as δ (and maybe several others), therefore, it holds $\delta^*(\mathbf{e}') = \mathbf{v}$.

Can the decoding δ^* correct the word $\mathbf{e}'$? If yes, then it has to hold: $\delta^*(\mathbf{o} + \mathbf{e}') = \mathbf{o}$, what is in contradiction with $\delta^*(\mathbf{e}') = \mathbf{v} \neq \mathbf{o}$. ■

Theorem 4.28. *Let $d = \Delta(\mathcal{K})$ be the minimum distance of a linear code $\mathcal{K}$, $t < \dfrac{d}{2}$. Then the standard decoding corrects all t-tuple simple errors.*

Proof. Let $\mathbf{e}$ be a word of the weight $\|\mathbf{e}\| = t < \dfrac{d}{2}$. Let $\mathbf{v} \in (\mathbf{e} + \mathcal{K})$, $\mathbf{v} \neq \mathbf{e}$, $\mathbf{v} = \mathbf{e} + \mathbf{u}$, $\mathbf{u} \in \mathcal{K}$. Then $\|\mathbf{u}\| \geq d$, $\|\mathbf{e}\| = t < \dfrac{d}{2}$. Therefore, the number of non zero characters of the word $\mathbf{v} = \mathbf{e} + \mathbf{u}$ is at least $d - t$ – i. e., $\|\mathbf{v}\| > d - t > t$. Hence, every word $\mathbf{e}$ with Hamming weight less than $\dfrac{d}{2}$ is the (unique) representative of some class according to the code $\mathcal{K}$.
By the theorem 4.26, the standard decoding corrects all error words that are representatives of all classes, therefore, it corrects all error words of Hamming weight less than $\dfrac{d}{2}$ what is equivalent with the fact that standard decoding corrects all t-tuple simple errors. ■

The principle of standard decoding is the determining which class of words according to the code $\mathcal{K}$ contains the decoded word. For this purpose the decoding algorithm has to search the decoded word $\mathbf{w}$ in so called **Slepian's table** of all words of the length n of alphabet A.

It is the table which has the number m of columns equal to the number of classes of words according to the code $\mathcal{K}$ – $m = p^{n-k}$, and the number q of rows equal to the number of code words – $q = p^k$. In every column, there are all words of one class, in the first row of the table, there are representatives of corresponding classes.

After determining which column contains the decoded word $\mathbf{w}$ we decode in this way that we subtract from $\mathbf{w}$ the word in the first row of the corresponding column.

	Class $\mathbf{e}_1+\mathcal{K}$	Class $\mathbf{e}_2+\mathcal{K}$		Class $\mathbf{e}_m+\mathcal{K}$
representative	$\mathbf{e_1}=\mathbf{e_1}+\mathbf{o}$	$\mathbf{e_2}=\mathbf{e_2}+\mathbf{o}$	...	$\mathbf{e_m}=\mathbf{e_m}+\mathbf{o}$
elements of classes	$\mathbf{e_1}+\mathbf{u_1}$	$\mathbf{e_2}+\mathbf{u_1}$	...	$\mathbf{e_m}+\mathbf{u_1}$
	$\mathbf{e_1}+\mathbf{u_2}$	$\mathbf{e_2}+\mathbf{u_2}$	...	$\mathbf{e_m}+\mathbf{u_2}$
	...	...	...	...
	...	...	...	...
	...	...	...	...
	$\mathbf{e_1}+\mathbf{u_q}$	$\mathbf{e_2}+\mathbf{u_q}$	...	$\mathbf{e_m}+\mathbf{u_q}$

(4.58)

Slepian's table, $m = p^{n-k}$, $q = |\mathcal{K}| = p^k$.

Slepian's table has p^n elements. In worst case whole the table has to be searched. The size of this table for often used 64-bit binary codes is $2^{64} > 10^{19}$. Clever implementation replaces full search by binary search and reduces the number of accesses to the table to 64, but memory requirements remain enormous.

The complexity of this problem can be reduced significantly if we remember that all words of one class $\mathbf{e}+\mathcal{K}$ have the same syndrome as its representative $\mathbf{e}$. Really, it holds for $\mathbf{v} \in \mathcal{K}$ and the check matrix $\mathbf{H}$ of the code $\mathcal{K}$:

$$\mathbf{H}.(\mathbf{e}+\mathbf{v}) = \mathbf{H}.\mathbf{e} + \mathbf{H}.\mathbf{v} = \mathbf{H}.\mathbf{e} + \mathbf{o} = \mathbf{H}.\mathbf{e}\ .$$

Therefore, the table with only two rows suffices instead of Slepian's table. This table contains representatives $\mathbf{e_1}, \mathbf{e_2}, \dots, \mathbf{e_m}$ of classes in the first row and corresponding syndromes $\mathbf{s_1}, \mathbf{s_2}, \dots, \mathbf{s_m}$

representative	$\mathbf{e}_1$	$\mathbf{e}_2$	...	$\mathbf{e}_m$
syndrome	$\mathbf{s}_1$	$\mathbf{s}_2$	...	$\mathbf{s}_m$

(4.59)

Now the decoding procedure can be reformulated as follows: Calculate the syndrome of the received word $\mathbf{w}$: $\mathbf{s} = \mathbf{H}.\mathbf{w}$. Find this syndrome $\mathbf{s}$ in the second row of the table (4.59) and use the corresponding representative $\mathbf{e}$ from the first row of this table and decode:

$$\delta(\mathbf{w}) = \mathbf{w} - \mathbf{e}\ .$$

The table (4.59) has p^{n-k} columns and only two rows – its size is significantly less than that of the original Slepian's table. Moreover, we can await that even by large length n of a linear block code $\mathcal{K}$ the number $n-k$ will not rise too much since it means the number of check digits and our effort is to maintain a good information ratio.

4.15 Hamming codes

Theorem 4.29. *A linear code with alhabet with p characters corrects one simple error if and only if none of the columns of its check matrix is a scalar multiple of another column.*
Specially a binary code corrects one simple error if and only if its check matrix contains mutually different non zero columns.

Proof. We know that a code $\mathcal{K}$ corrects one error if and only if $\Delta(\mathcal{K}) \geq 3$ what by theorem 4.23 (page 123) occurs if and only if arbitrary two columns of its check matrix $\mathbf{H}$ are linearly independent.

Two vectors $\mathbf{u}$, $\mathbf{v}$ are independent in general case if and only if none of them is a scalar multiple of another. In the case of binary alphabet if and only if both vectors $\mathbf{u}$, $\mathbf{v}$ are non zero and different. ■

Definition 4.31. A binary linear (n, k)-code is called **Hamming code**, if its check matrix $\mathbf{H}$ has $(2^{(n-k)} - 1)$ columns – all non zero binary words of the length $n - k$ every one of them occurs as a column of the matrix $\mathbf{H}$ exactly once.

Check matrix $\mathbf{H}$ of a linear (n, k)-code has n columns, that is why

$$n = 2^{(n-k)} - 1.$$

Therefore Hamming codes exist only for the following (n, k):

$$(n, k) = (3, 1),\ (7, 4),\ (15, 11),\ (31, 26), \dots (2^m - 1, 2^m - m - 1), \dots\ .$$

Note that the information ratio (4.43) (page 107) converges to 1 with $m \to \infty$. For example for $m = 6$ Hamming $(63, 57)$-code has information ratio $\dfrac{57}{63} > 0.9$.

Definition 4.32. Decoding of Hamming code. Let $\mathcal{K}$ be a Hamming (n, k)-code where $n = 2^m - 1$, $k = 2^m - m - 1$ with check matrix $\mathbf{H}$. Suppose that the columns of the matrix are ordered such that the first column is binary representation of number 1, the second column is binary representation of 2 etc. After receiving a word $\mathbf{w}$ we calculate its syndrome $\mathbf{s} = \mathbf{Hw}$. If $\mathbf{s} = \mathbf{o}$, the word $\mathbf{w}$ is a code word and remains unchanged. If $\mathbf{s} \neq \mathbf{o}$, the word $\mathbf{s}$ is the binary representation of a number i and we change the character on i-th position of the received word $\mathbf{w}$. Formally:

$$\delta(\mathbf{w}) = \begin{cases} \mathbf{w}, & \text{if } \mathbf{s} = \mathbf{o} \\ \mathbf{w} - \mathbf{e_i}, & \text{if } \mathbf{s} \text{ is the binary representation of the number } i, \end{cases} \tag{4.60}$$

where $\mathbf{e}_i$ is the word having character 1 on the position i and characters 0 on all other positions.

Theorem 4.30. *The decoding δ defined in* (4.60) *corrects one simple error. More precisely: If the word $\mathbf{w}$ differs from a code word $\mathbf{v}$ at most at one position then $\delta(\mathbf{w}) = \mathbf{v}$.*

Proof. If $\mathbf{w} = \mathbf{v}$ then $\mathbf{w}$ is a code word and $\mathbf{Hw} = \mathbf{Hv} = \mathbf{o}$ holds. In this case $\delta(\mathbf{w}) = \mathbf{w} = \mathbf{v}$.

Let the words $\mathbf{v}$, $\mathbf{w}$ differ exactly at one position i, i. e., $\mathbf{w} = \mathbf{v} + \mathbf{e_i}$ where $\mathbf{e}_i$ is the word containing exactly one character 1 on the position i, $i \in \{1, 2, \dots, n\}$. Then

$$\mathbf{Hw} = \mathbf{H}(\mathbf{v} + \mathbf{e_i}) = \mathbf{Hv} + \mathbf{He_i} = \mathbf{He_i} \ .$$

Then $\mathbf{He_i}$ is i-th column of the matrix $\mathbf{H}$ and this column is the binary representation of number i. Therefore, the decoding $\delta(\mathbf{w}) = \mathbf{w} - \mathbf{e_i} = \mathbf{v}$ decodes correctly. ■

The most economic error correcting codes are perfect codes. By definition 4.15 (page 103) a block code $\mathcal{K}$ of the length n is t-perfect if the set of balls $\{B_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ is a partition of the set A^n of all words of the length n.

Theorem 4.31. *A linear code $\mathcal{K}$ is t-perfect if and only if the set of all words of the weight less or equal to t is the system of all representatives of all classes of words according to the code $\mathcal{K}$.*

Proof. First note that every word $\mathbf{a} \in A^n$ can be the representative of some class according to the code $\mathcal{K}$ – namely that of the class $\mathbf{a} + \mathcal{K}$.

In order to prove that the set of all words with weight less or equal to t is the set of all representatives of all classes, we have to prove two facts:

- every class contains a word with Hamming weight less or equal to t
- if $\mathbf{e_1}$, $\mathbf{e_2}$ are two words such that $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$, then $\mathbf{e}_1 + \mathcal{K}$, $\mathbf{e}_2 + \mathcal{K}$ are two different classes, i. e., $\mathbf{e}_2 \notin (\mathbf{e}_1 + \mathcal{K})$

1. Let $\mathcal{K}$ be a t-perfect linear code – i. e., for every word $\mathbf{a} \in A^n$ there exists exactly one code word $\mathbf{b} \in \mathcal{K}$ such that the distance of words $\mathbf{a}$, $\mathbf{b}$ is less or equal to t, i. e., $d(\mathbf{a},\ \mathbf{b}) \leq t$. Denote $\mathbf{e} = \mathbf{a} - \mathbf{b}$. Since the Hamming distance of words $\mathbf{a}$, $\mathbf{b}$ is less or equal to t it holds $\|\mathbf{e}\| \leq t$ and $\mathbf{a} = \mathbf{e} + \mathbf{b}$.
Every class $\mathbf{a} + \mathcal{K}$ has a representative $\mathbf{e}$ with Hamming weight less or equal to t.

Let $\mathbf{e}_1$, $\mathbf{e}_2$ are two words such that $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$ and $\mathbf{e}_2 \in (\mathbf{e}_1 + \mathcal{K})$. Then $\mathbf{e_2} - \mathbf{e_1} \in \mathcal{K}$ and $\|\mathbf{e_2} - \mathbf{e_1}\| \leq 2t$. The last inequality implies that $\Delta(\mathcal{K}) \leq 2t$ which is in contradiction with the assumption that $\mathcal{K}$ corrects t simple errors. By the theorem 4.12 (page 103) the code $\mathcal{K}$ corrects t errors if and only if $\Delta(\mathcal{K}) \geq 2t + 1$.

2. Let the set of all words of the Hamming weight less or equal to t is the system of all representatives of all classes of words according to the code $\mathcal{K}$.
At first we show that $\Delta(\mathcal{K}) \geq 2t + 1$. Suppose that there is a non zero word $\mathbf{a} \in \mathcal{K}$ such that $\|\mathbf{a}\| < 2t + 1$ Then it is possible to write $\mathbf{a} = \mathbf{e_1} - \mathbf{e_2}$ where $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$ and $\mathbf{e_1} \neq \mathbf{e_2}$. By assertion (i) of the theorem 4.25 (page 125) it holds $(\mathbf{e}_1 + \mathcal{K}) = (\mathbf{e}_2 + \mathcal{K})$, which is in contradiction with the assumption that $\mathbf{e_1}$, $\mathbf{e_2}$ are representatives of different classes. If $\Delta(\mathcal{K}) \geq 2t + 1$ then the balls $\{B_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ are mutually disjoint.
Finally we show that for every $\mathbf{a} \in A^n$ there exists a ball $B_t(\mathbf{b})$, $\mathbf{b} \in \mathcal{K}$ such that $\mathbf{a} \in B_t(\mathbf{b})$. By the assumption there exists $\mathbf{e} \in A^n$, $\|\mathbf{e}\| \leq t$ such that $\mathbf{a} \in (\mathbf{e} + \mathcal{K})$. Hence we can write $\mathbf{a} = \mathbf{e} + \mathbf{b}$ for some $\mathbf{b} \in \mathcal{K}$. Therefore, $\mathbf{a} - \mathbf{b} = \mathbf{e}$, $d(\mathbf{a}, \mathbf{b}) = \|(\mathbf{a} - \mathbf{b})\| = \|\mathbf{e}\| \leq t$ and thus $\mathbf{a} \in B_t(\mathbf{b})$.
The system of balls $\{B_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ is a partition of the set A^n – the code $\mathcal{K}$ is t-perfect. ■

Theorem 4.32. *All Hamming binary codes are* 1*-perfect. Every* 1*-perfect binary linear code is a Hamming code.*

Proof. Let $\mathcal{K}$ be a Hamming linear (n, k)-code with $n = 2^m - 1$ and $k = 2^m - m - 1$, let $\mathbf{H}$ be the check matrix of $\mathcal{K}$. The Hamming code $\mathcal{K}$ has $n - k = m$ check characters and by the assertion (iii) of the theorem 4.25 (page 125) has $2^{(n-k)} = 2^m$ classes.
Denote $\mathbf{e}_0 = \mathbf{o}$ the zero word of the length $2^m - 1$ and for $i = 1, 2, \dots, 2^m - 1$

$$\mathbf{e}_i = \begin{bmatrix} 0 & 0 & \dots & 0 & \underbrace{1}_{i\text{-th position}} & 0 \dots & 0 \end{bmatrix}.$$

All $\mathbf{e}_i$ for $i = 1, 2, \dots, 2^m - 1$ are non-code words with the Hamming weight equal to 1.

Examine the classes $\mathbf{e}_i + \mathcal{K}$ for $i = 0, 1, 2, \dots, 2^m - 1$. The class $\mathbf{e}_0 + \mathcal{K}$ is equal to the set of code words $\mathcal{K}$ and that is why it is different from all other classes.
Suppose that the classes $\mathbf{e}_i + \mathcal{K}$, $\mathbf{e}_j + \mathcal{K}$ are equal for $i \neq j$. Then $\mathbf{e_i} - \mathbf{e_j} \in \mathcal{K}$, what implies that $\mathbf{H}(\mathbf{e}_i - \mathbf{e}_j) = \mathbf{o} = \mathbf{c}_i - \mathbf{c}_j$ where $\mathbf{c}_i$ and $\mathbf{c}_j$ are i-th and j-th

column of **H**. Since the check matrix of a Hamming code cannot contain two equal columns, the classes $\mathbf{e}_i + \mathcal{K}$, $\mathbf{e}_j + \mathcal{K}$ are different.

Since, as we have shown, the Hamming code $\mathcal{K}$ has 2^m classes and that all classes of the type $\mathbf{e}_i+\mathcal{K}$ for $i = 0, 1, 2, \dots, 2^m-1$ are different, there is no other class. The set of all words of the length less or equal to 1 creates the system of all representatives of all classes according to $\mathcal{K}$, that is why the code $\mathcal{K}$ is 1-perfect.

Let us have a 1-perfect linear (n, k) code $\mathcal{K}$ with $m = (n-k)$ check characters. The code $\mathcal{K}$ has 2^m classes of words by the assertion (iii) of the theorem 4.25 (page 125).

Denote by **H** the check matrix of $\mathcal{K}$. The matrix **H** has n rows and m columns. By the theorem 4.29 all columns of **H** have to be mutually different and non-zero – then $n \le 2^m - 1$. The code $\mathcal{K}$ is 1-perfect. By the theorem 4.31 (page 131) all binary words of the length n with the weight 1 or 0 are exactly all representatives of all classes. The number of such words is $n+1$ (zero word and all words of the type $\mathbf{e}_i$ with exactly one character 1 on position i). Therefore, it holds:

$$n + 1 = 2^m,$$

and

$$n = 2^m - 1.$$

The type of the check matrix of the code $\mathcal{K}$ is $(2^m - 1) \times m$ and its column are exactly all nonzero words of the length m. Hence $\mathcal{K}$ is a Hamming code. ■

Definition 4.33. Extended Hamming binary code is a binary code which originated by adding parity bit to all code words of a Hamming code.

The extended Hamming code is the linear $(2^m, 2^m - m - 1)$-code of all words $\mathbf{v} = v_1v_2 \dots v_{2^m}$ such that $v_1v_2 \dots v_{2^m-1}$ is a word of a Hamming code and $v_1 + v_2 + \cdots + v_{2^m} = 0$. The minimum distance of an extended Hamming code is 4. This code corrects single errors and detects triple errors.

Remark. Theorem 4.29 gives a hint how to define a p-character Hamming code as the code with check matrix **H** of the type $(n \times m)$ such that

(i) none column is a scalar multiple of other column

(ii) for every non zero word $\mathbf{a} \in A^m$ there exists a column $\mathbf{c}$ of **H** such $\mathbf{a}$ is a scalar multiple of $\mathbf{c}$

The matrix $\mathbf{H}$ can be constructed from all nonzero columns of the length m whose the first nonzero character is 1. It can be shown that p-ary Hamming codes have a lot of properties similar to binary Hamming codes, e. g. all Hamming codes are 1-perfect.

4.16 Golay code*

Denote by $\mathbf{B}$ the square matrix of the type 11×11 whose the first row contains the binary word 11011100010 and next rows are right rotations of the first one, i. e.,

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & & 1 & 1 & 1 & & & & 1 & \\ & 1 & 1 & & 1 & 1 & 1 & & & & 1 \\ 1 & & 1 & 1 & & 1 & 1 & 1 & & & \\ & 1 & & 1 & 1 & & 1 & 1 & 1 & & \\ & & 1 & & 1 & 1 & & 1 & 1 & 1 & \\ & & & 1 & & 1 & 1 & & 1 & 1 & 1 \\ 1 & & & & 1 & & 1 & 1 & & 1 & 1 \\ 1 & 1 & & & & 1 & & 1 & 1 & & 1 \\ 1 & 1 & 1 & & & & 1 & & 1 & 1 & \\ & 1 & 1 & 1 & & & & 1 & & 1 & 1 \\ 1 & & 1 & 1 & 1 & & & & 1 & & 1 \end{bmatrix}. \tag{4.61}$$

Binary word 11011100010 has on i-th position 1 if and only if $i-1$ is a square modulo 11, i. e., if $i-1 = 0^2$, 1^2, 2^2, 3^2, $4^2 \equiv 5$ a $5^2 \equiv 3$. In this section 4.16 we will suppose that the matrix $\mathbf{B}$ is given by (4.61).

Definition 4.34. Golay code G_{23} is the systematic binary code of the length 23 with generating matrix $\mathbf{G}_{23}$ defined as follows:

$$\mathbf{G}_{23} = \left[\begin{array}{c|c} & \mathbf{B}_{11\times 11} \\ \mathbf{E}_{12\times 12} & \\ \cline{2-2} & 11\ldots 11 \end{array} \right],$$

where $\mathbf{E}_{12\times 12}$ is the unit matrix of the type 12×12, $\mathbf{B}_{11\times 11}$ is the square matrix of the type 11×11 defined in (4.61).

The **Golay code** G_{24} is the systematic binary code of the length 24 with generating matrix $\mathbf{G}_{24}$ which originates from matrix $\mathbf{G}_{23}$ by adding the column

$11\dots10$, i. e.,

$$\mathbf{G}_{24} = \left[\begin{array}{c|c|c} & & 1 \\ & & 1 \\ \mathbf{E}_{12\times12} & \mathbf{B}_{11\times11} & \dots \\ & & 1 \\ \cline{2-3} & 11\dots11 & 0 \end{array}\right]$$

$$\begin{bmatrix}
1 & & & & & & & & & & & & 1 & 1 & & 1 & 1 & 1 & & & & 1 & & 1 \\
& 1 & & & & & & & & & & & & 1 & 1 & & 1 & 1 & 1 & & & & 1 & 1 \\
& & 1 & & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & & & 1 \\
& & & 1 & & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & & 1 \\
& & & & 1 & & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & 1 \\
& & & & & 1 & & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & 1 \\
& & & & & & 1 & & & & & & 1 & & & & 1 & & 1 & 1 & & 1 & 1 & 1 \\
& & & & & & & 1 & & & & & 1 & 1 & & & & 1 & & 1 & 1 & & 1 & 1 \\
& & & & & & & & 1 & & & & 1 & 1 & 1 & & & & 1 & & 1 & 1 & & 1 \\
& & & & & & & & & 1 & & & & 1 & 1 & 1 & & & & 1 & & 1 & 1 & 1 \\
& & & & & & & & & & 1 & & 1 & & 1 & 1 & 1 & & & & 1 & & 1 & 1 \\
& & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 &
\end{bmatrix}$$

Generating matrix of Golay code $\mathbf{G}_{24}$.

The properties of codes G_{24}, G_{23}.

- Golay code G_{24} has 12 information characters and 12 check characters.
- The dual code to the Golay code G_{24} is G_{24} itself. The generating matrix of G_{24} is also its check matrix[6].
- The minimum distance of the code G_{24} is 8.
- The Golay code G_{23} is a 3-perfect $(23, 12)$-code.

Theorem 4.33. Tietvinen, Van Lint. *The only nontrivial perfect binary codes are:*

a) Hamming codes correcting single errors,

b) Golay code G_{23} correcting triple errors and codes equivalent with G_{23},

[6] It suffices to verify that the scalar multiple of arbitrary two different rows of generating matrix $\mathbf{G}_{24}$ equals to 0.

c) *repeating codes of the odd length* $2t + 1$ *correcting* t*-tuple errors for* $t = 1, 2, 3, \dots$.

The reader can find proofs of this theorem and other properties of Golay codes in [1].

Another interesting code is the Golay perfect ternary $(11, 6)$-code which corrects 3 errors. Its generating matrix is in the form:

$$\mathbf{G}_{11} = \left[\begin{array}{c|c} \mathbf{E}_{6\times 6} & \begin{array}{c} \mathbf{D}_{5\times 5} \\ \hline 11\dots 11 \end{array} \end{array} \right],$$

where $\mathbf{E}_{6\times 6}$ is the unit matrix of the type 6×6 and where $\mathbf{D}_{5\times 5}$ is the matrix whose rows are all right cyclic rotations of the word 01221. Golay code G_{11} (and equivalent codes), Hamming codes and repeating codes of odd length are the only ternary nontrivial perfect codes.

In the case of code alphabet with more than 3 characters the only nontrivial perfect codes are Hamming codes and repeating codes of odd length $2t + 1$.

At the end of this chapter, it is necessary to say that it contains only an introduction to the coding theory and practice. A lot of topics of coding theory and coding methods could not be included because the size of this publication is limited and many omitted subjects require deeper knowledge of notions of finite algebra as rings of polynomial, Boolean polynomial, finite fields, etc. Such themes are e. g., cyclic codes, Reed-Muller codes, BCH codes, etc. The interested reader can find more about coding theory in [1], [2], [11]. Nevertheless, I hope that the knowledge of this chapter can help the reader in orientation in coding field of interest.

Chapter 5

Communication channels

5.1 Informal notion of a channel

A communication channel is a communication device with two ends, an input end and an output one. The input accepts the characters of some input alphabet Y and delivers the characters of an output alphabet Z. In most cases $Y = Z$, but there are cases when a channel works with different input and output alphabets. That is why we will distinguish input alphabet and output alphabet.

Example 5.1. Let $Y = \{0, 1\}$ be the input alphabet Y of a channel represented by voltage level $0 = L$-(low – e. g., 0.7 V) and $1 = H$-(high – e. g., 5.5 V). These voltage levels can slightly change during transmission, therefore, we can represent the voltage range $\langle 0.7, 2.3\rangle$ as character 0 and voltage range $\langle 3.9, 5.5\rangle$ as character 1 and the voltage range $(2.3, 3, 9)$ will be represented as erroneous character "*". The output alphabet will be $Z = \{0, 1, *\}$.

Example 5.2. Let the input alphabet Y of a channel be the set of all 8-bit binary numbers with even parity. If the channel is a noisy channel, the output can deliver any 8-bit number. The output alphabet Z is in this case the set of all 8-bit numbers.

The input of a channel accepts a sequence of characters $y_1, y_2, y_3, \dots$ in discrete time moments $i = 1, 2, 3, \dots$, and it delivers a sequence of output characters in the corresponding time moments, i. e., if the character y_i appears on the input, the character z_i appears on the output in the corresponding time moment. The assumption of the simultaneous appearance of input character and corresponding output character on the input and output contradicts physical law by which the speed of even the fastest particles – photons – is limited, but the delay is in most cases negligible for our purposes.

5.2 Noiseless channel

The simplest case of communication channel is **memoryless noiseless channel** where the received character z_i in time i depends only on the transmitted character y_i in the corresponding time[1] , – i. e.:

$$z_i = f_i(y_i) \ ,$$

In a **noiseless channel with memory** the character z_i received in time i uniquely depends on transmitted word $y_1, y_2, \dots, y_i$ in time moments[2] $i = 1, 2, \dots, i$, i. e.,

$$z_i = F_i(y_1, y_2, \dots, y_i) \ ,$$

Another type of communication channel is the **noiseless channel with finite memory**, where the output character z_i depends only on the last m transmitted characters, i. e.,

$$z_i = F_i(y_{i-m+1}, y_{i-m+2}, \dots, y_i) \ .$$

We will require that channels have one obvious property, namely that the output character z_i does not depend on any input character y_{i+k}, $k > 0$. Any character received at time i depends only on characters transmitted in time moments $1, 2, \dots, i$, but it does not depend on any character transmitted after time i. We say that a channel is not predictive.

Noiseless channel is uniquely defined by the system of functions $\{f_i\}_{i=1,2,\dots}$, resp. $\{F_i\}_{i=1,2,\dots}$.

[1] The most common case is when $Y = Z$ and f_i is the identity on Y for every i. In general case the function f_i can depend on time moment i.

[2] For example the key ⟨`CapsLock`⟩ causes that after its hitting, the keyboard transmits upper case letters and another pressing returns the keyboard to lower case mode. This channel remembers forever that the key ⟨`CapsLock`⟩ was transmitted.
Similarly the input ⟨`Alt`⟩/⟨`Shift`⟩ under OS Windows switches between US and national keyboard.

5.3 Noisy communication channels

In real situations a noiseless channel is rather an exception than a rule. What makes our life interesting in modern time is "channel noise" – you cannot be dead certain what the output will be for a given input. Industrial interference, weather impact, static electricity, birds flying around antennas and many other negative effects are the causes of transmission failures[3].

After transmitting an input word $y_1, y_2, \dots, y_i$, we can receive, owing to noise, an arbitrary word $z_1, z_2, \dots, z_i$, of course, every one with a different probability. The conditional probability of receiving the word $z_1, z_2, \dots, z_i$ given the input word $y_1, y_2, \dots, y_i$ was transmitted will be denoted by

$$\nu(z_1, z_2, \dots, z_i | y_1, y_2, \dots, y_i) \ .$$

Since the input alphabet Y, output alphabet Z and the function

$$\nu : \bigcup_{i=1}^{\infty} (Z^i \times Y^i) \to \langle 0, 1 \rangle$$

fully characterize the communication channel we can define:

Definition 5.1. The **communication channel** $\mathcal{C}$ is an ordered triple $\mathcal{C} = (Y, Z, \nu)$ where Y is an input alphabet, Z is an output alphabet and $\nu : \bigcup_{i=1}^{\infty} (Z^i \times Y^i) \to \langle 0, 1 \rangle$, $\nu(z_1, z_2, \dots, z_i | y_1, y_2, \dots, y_i)$ is the conditional probability of the event that the word $z_1, z_2, \dots, z_i$ occurs on the output given the input word is $y_1, y_2, \dots, y_i$.

Denote $\nu_i(z_i | y_1, y_2, \dots, y_i)$ the conditional probability of the event that the character z_i occurs on the output in time moment i given the word $y_1, y_2, \dots, y_i$ is on the input of the channel. Then

$$\nu_i(z_i | y_1, y_2, \dots, y_i) = \sum_{z_1, z_2, \dots, z_{i-1}} \nu(z_1, z_2, \dots, z_i | y_1, y_2, \dots, y_i).$$

We say that the channel $\mathcal{C}$ is **memoryless channel**, if $\nu_i(z_i | y_1, y_2, \dots, y_i)$ depends only on y_i, i. e., if

$$\nu_i(z_i | y_1, y_2, \dots, y_i) = \nu_i(z_i | y_i).$$

[3] A human can be also considered a transmission channel. He reads numbers (of goods, bank accounts, railway cars, personal identification numbers etc.) or a text and transmits character in such a way that he types them on a keyboard into a registration cash desk or a computer. Humans make errors that is why this channel is a noisy channel. Error correction codes are often used in noisy channels in order to ensure reliable communication.

If moreover $\nu_i(z_i|y_i)$ does not depend on i, i. e., if $\nu_i(z_i|y_i) = \nu(z_i|y_i)$, we say that $\mathcal{C}$ is **stationary memoryless channel**.
If

$$\nu(z_1, z_2, \dots, z_i|y_1, y_2, \dots, y_i) = \nu(z_1|y_1)\nu(z_2|y_2)\dots\nu(z_i|y_i) = \prod_{k=1}^{i} \nu(z_k|y_k),$$

we say that $\mathcal{C}$ is the **stationary independent channel**.

5.4 Stationary memoryless channel

Let us have a stationary memoryless channel with input alphabet $A = \{a_1, a_2, \dots, a_n\}$ and output alphabet $B = \{b_1, b_2, \dots, b_r\}$. Denote $q_{ij} = \nu(b_j|a_i)$ the conditional probability that the character b_j occurs on the output given the input character is a_i.
Numbers q_{ij} are called **transition probabilities** and the matrix of the type $n \times r$

$$\mathbf{Q} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1r} \\ q_{21} & q_{22} & \cdots & q_{2r} \\ \cdots & \cdots & \cdots & \cdots \\ q_{n1} & q_{n2} & \cdots & q_{nr} \end{pmatrix}$$

is **matrix of transition probabilities**. Note that the sum of elements of every row of the matrix $\mathbf{Q}$ equals to 1, i. e., $\sum_{j=1}^{r} q_{kj} = 1$ for every $k = 1, 2, \dots, n$.

Let $p_i = P(a_i)$ be the probability of the event that the character a_i occurs on the input of the channel. The joint probability $P(a_i \cap b_j)$ of the event, that the character a_i occurs on channel input and at the same time the character b_j occurs on channel output is:

$$P(a_i \cap b_j) = p_i q_{ij}.$$

The probability $P(b_j)$ that b_j occurs on the output can be calculated as the sum of probabilities: $P(a_1 \cap b_j) + P(a_2 \cap b_j) + \dots + P(a_n \cap b_j)$, i. e.,

$$P(b_j) = \sum_{t=1}^{n} p_t q_{tj}.$$

The occurrence of the character a_i on the channel input, resp., the occurrence of the character b_j on the channel output can be considered as the result of experiments

$$\begin{aligned}\mathbf{A} &= \{\{a_1\},\{a_2\},\dots,\{a_n\}\},\\ \mathbf{B} &= \{\{b_1\},\{b_2\},\dots,\{b_r\}\}.\end{aligned}$$

The person who receives messages wants to know what character was transmitted – the result of the experiment **A**. However, he knows only the result of the experiment **B**. We have shown in section 2.7 that the mean value of information about experiment **A** contained in experiment **B** can be expressed as the mutual information $I(\mathbf{A},\mathbf{B})$ of experiments **A**, **B** for which we make use of the formula (2.14) from the theorem 2.14 (page 47)

$$I(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j).\log_2\left(\frac{P(A_i\cap B_j)}{P(A_i).P(B_j)}\right). \tag{5.1}$$

The formula (5.1) can be rewritten in terms of probabilities p_i, q_{ij} as follows:

$$\begin{aligned}I(\mathbf{A},\mathbf{B}) &= \sum_{i=1}^{n}\sum_{j=1}^{r} P(a_i\cap b_j)\log_2\frac{P(a_i\cap b_j)}{P(a_i)P(b_j)}\\ &= \sum_{i=1}^{n}\sum_{j=1}^{r} p_i q_{ij}\log_2\frac{p_i q_{ij}}{p_i\sum_{t=1}^{n}p_t q_{tj}}\\ &= \sum_{i=1}^{n}p_i\sum_{j=1}^{r} q_{ij}\log_2\frac{q_{ij}}{\sum_{t=1}^{n}p_t q_{tj}}. \qquad (5.2)\end{aligned}$$

If the experiment **A** will be independently repeated many times (i. e., if the outputs of a stationary memoryless source (A^*,P) with character probabilities p_i, $i=1,2,\dots,n$ occur on the input of the channel), the expression (5.1) resp., (5.2) is the mean value of information per character transmitted through the channel.

Symmetric binary channel is a channel with input alphabet $A = \{0,1\}$, output alphabet $B = \{0,1\}$, and matrix of transmission probabilities

$$\mathbf{Q} = \begin{pmatrix} q & 1-q \\ 1-q & q \end{pmatrix}, \tag{5.3}$$

where $0 \leq q \leq 1$. In this case $n = 2$ and $r = 2$.

Note that for $q = 1/2$ is

$$\mathbf{Q} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}, \tag{5.4}$$

and that is why

$$\begin{aligned} I(\mathbf{A},\mathbf{B}) &= \sum_{i=1}^{2} p_i \sum_{j=1}^{2} \frac{1}{2} \log_2 \frac{1/2}{\sum_{t=1}^{2} p_t.1/2} \\ &= \sum_{i=1}^{2} p_i \sum_{j=1}^{2} \frac{1}{2} \log_2 \frac{1/2}{(1/2).\sum_{t=1}^{2} p_t} \\ &= \sum_{i=1}^{2} p_i \sum_{j=1}^{2} \frac{1}{2} \log_2 1 = 0 \end{aligned}$$

for arbitrary values of probabilities p_1, p_2. The channel transmits no information in this case.

Let us return to general stationary memoryless channel and let us search for probabilities $p_1, p_2, \ldots, p_n$ which maximize the amount of transferred information. This problem can be formulated as a problem to maximize the function (5.2) subject to constraints $\sum_{i=1}^{n} p_i = 1$ and $p_i \geq 0$ for $i = 1, 2, \ldots, n$. To solve this problem Lagrange multipliers method can be applied.

Set

$$\begin{aligned}F(p_1, p_2, \dots, p_n) = I(A,B) + \lambda\Big(1 - \sum_{i=1}^{n} p_i\Big) = \\ = \sum_{i=1}^{n} p_i \sum_{j=1}^{r} q_{ij} \underbrace{\log_2 \frac{q_{ij}}{\sum_{t=1}^{n} p_t q_{tj}}}_{(*)} + \lambda\Big(1 - \sum_{i=1}^{n} p_i\Big) \ .\end{aligned} \tag{5.5}$$

Partial derivative of the term (*) in (5.5) is calculated as follows:

$$\begin{aligned}\frac{\partial}{\partial p_k} \log_2 \frac{q_{ij}}{\sum_{t=1}^{n} p_t q_{tj}} = \frac{\partial}{\partial p_k} \log_2(e) \cdot \ln \frac{q_{ij}}{\sum_{t=1}^{n} p_t q_{tj}} = \\ = \log_2(e) \cdot \frac{\sum_{t=1}^{n} p_t q_{tj}}{q_{ij}} \cdot \frac{q_{ij}}{-\Big(\sum_{t=1}^{n} p_t q_{tj}\Big)^2} \cdot q_{kj} = -\log_2(e) \cdot \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} \ .\end{aligned}$$

Then it holds for partial derivative of F with respect to k-th variable:

$$\begin{aligned}\frac{\partial F}{\partial p_k} &= \frac{\partial}{\partial p_k}\left(I(A,B) + \lambda\Big(1 - \sum_{i=1}^{n} p_i\Big)\right) = \frac{\partial}{\partial p_k}\Big(I(A,B)\Big) - \lambda \\ &= \sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} - \log_2 e \sum_{i=1}^{n} p_i \sum_{j=1}^{r} \frac{q_{ij} q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} - \lambda \\ &= \sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} - \log_2 e \sum_{j=1}^{r} \frac{\sum_{i=1}^{n} p_i q_{ij}}{\sum_{t=1}^{n} p_t q_{tj}} q_{kj} - \lambda\end{aligned} \tag{5.6}$$

$$\begin{aligned} &= \sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} - \log_2 e \sum_{j=1}^{r} q_{kj} - \lambda \\ &= \sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} - \underbrace{(\log_2 e + \lambda)}_{\gamma} \ .\end{aligned} \tag{5.7}$$

Denote $(\log_2 e + \lambda) = \gamma$ and set all partial derivatives equal to 0. The result is the following system of equations for unknown $p_1, p_2, \ldots, p_n$ and γ:

$$\sum_{i=1}^{n} p_i = 1 \tag{5.8}$$

$$\sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} = \gamma \quad \text{for } k = 1, 2, \ldots, n \ . \tag{5.9}$$

It can be shown that the function $I(\mathbf{A}, \mathbf{B})$ of variables $p_1, p_2, \ldots, p_n$ in formula (5.2) is concave and that fulfilling of equations suffices for the maximality of information $I(\mathbf{A}, \mathbf{B})$. (see [7], part 3.4).
The equations (5.8) and (5.9) are called **capacity equations for the channel**.

Please observe that after substitution[4]

$$\sum_{j=1}^{r} q_{kj} \log_2 \frac{q_{kj}}{\sum_{t=1}^{n} p_t q_{tj}} = \gamma \quad \text{for } k = 1, 2, \ldots, n \ ,$$

into formula (5.2) we obtain

$$I(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{n} p_i \sum_{j=1}^{r} q_{ij} \log_2 \frac{q_{ij}}{\sum_{t=1}^{n} p_t q_{tj}} = \sum_{i=1}^{n} p_i \gamma = \gamma \sum_{i=1}^{n} p_i = \gamma \ .$$

If γ is the solution of the system (5.8) and (5.9) then the value of variable γ equals to the maximum amount of information which can be transmitted through the channel. This number will be considered as the capacity of the stationary memoryless channel. Theory of information studies more general types of communication channels and several different ways of defining channel capacity as we will see in section 5.6.

[4] γ is the solution of the system (5.8) and (5.9).

The capacity equations (5.3) for symmetric binary channel with the matrix **Q** (5.3) can be rewritten into the form:

$$\begin{aligned} p_1 + p_2 &= 1 \qquad (5.10)\\ q\log_2 \frac{q}{p_1 q + p_2(1-q)} + (1-q)\log_2 \frac{1-q}{p_1(1-q)+p_2 q} &= \gamma \qquad (5.11)\\ (1-q)\log_2 \frac{1-q}{p_1 q + p_2(1-q)} + q\log_2 \frac{q}{p_1(1-q)+p_2 q} &= \gamma\ . \qquad (5.12) \end{aligned}$$

Right sides of (5.11) and (5.12) are equal what implies the equality of left sides. After subtracting $q\log_2 q$ and $(1-q)\log_2(1-q)$ from both sides of this equality we get

$$\begin{aligned} &q\log_2[p_1 q + p_2(1-q)] + (1-q)\log_2[p_1(1-q)+p_2 q] = \\ &\qquad = (1-q)\log_2[p_1 q + p_2(1-q)] + q\log_2[p_1(1-q)+p_2 q]\ , \end{aligned}$$

from where

$$(2q-1)\log_2[p_1 q + p_2(1-q)] = (2q-1)\log_2[p_1(1-q)+p_2 p]\ . \qquad (5.13)$$

If $2q = 1$ then $q = 1/2$ and $I(\mathbf{A},\mathbf{B}) = 0$ regardless of the values of probabilities p_1, p_2.
If $q \neq 1/2$ then we have from (5.13) step by step:

$$\begin{aligned} p_1 q + p_2(1-q) &= p_1(1-q) + p_2 q\\ (2q-1)p_1 &= (2q-1)p_2\\ p_1 &= p_2. \qquad (5.14) \end{aligned}$$

Finally from (5.10) and (5.14) follows:

$$p_1 = p_2 = \frac{1}{2},$$

and after substituting p_1, p_2 in (5.11) or (5.12) we have

$$\gamma = q\log_2(2q) + (1-q)\log_2 2(1-q). \qquad (5.15)$$

The capacity of the symmetric binary channel $\mathcal{C}$ with matrix **Q** is given by the formula (5.3). Channel $\mathcal{C}$ transfers maximum amount of information for stationary binary independent source with equal probabilities $p_1 = p_2 = 1/2$ of both characters.

5.5 The amount of transferred information

Attach a source $\overline{\mathcal{S}} = (Y^*, \mu)$ to the input of a channel $\mathcal{C} = (Y, Z, \nu)$. Remember that the probability of transmitting the word $\mathbf{y} = (y_1, y_2, \dots, y_n)$ is $\mu(y_1, y_2, \dots, y_i)$. If the input of the channel $\mathcal{C}$ accepts input words from the source $\overline{\mathcal{S}}$, the output of the channel $\mathcal{C}$ can be regarded as a source denoted by $\mathcal{R} = \mathcal{R}(\mathcal{C}, \overline{\mathcal{S}})$ with alphabet Z and probability function π for which it holds

$$\begin{aligned} &\pi(\mathbf{z}) = \pi(z_1, z_2, \dots, z_n) = \\ &= \sum_{\mathbf{y} \in Y^n} \nu(\mathbf{z}|\mathbf{y})\mu(\mathbf{y}) = \sum_{y_1 y_2 \dots y_n \in Y^n} \nu(z_1, z_2, \dots, z_n | y_1, y_2, \dots, y_n) \cdot \mu(y_1, y_2, \dots, y_n). \end{aligned}$$

Together with the output source $\mathcal{R} = \mathcal{R}(\mathcal{C}, \overline{\mathcal{S}})$ we can define a so called double source $\mathcal{D} = ((Y \times Z)^*, \psi))$ depending on the source $\overline{\mathcal{S}}$ and the channel $\mathcal{C}$ which simulates a simultaneous appearing of the couples (y_i, z_i) of input and output characters on both ends of the channel $\mathcal{C}$.

If we identify the word $(y_1, z_1)(y_2, z_2) \dots (y_n, z_n)$ with the ordered couple

$$(\mathbf{y}, \mathbf{z}) = ((y_1, y_2, \dots, y_n), (z_1, z_2, \dots, z_n)),$$

we can express the probability

$$\psi\big((y_1, z_1)(y_2, z_2) \dots (y_n, z_n)\big) = \psi\big((y_1, y_2, \dots, y_n), (z_1, z_2, \dots, z_n)\big) = \psi(\mathbf{y}, \mathbf{z})$$

as follows:

$$\begin{aligned} \psi(\mathbf{y}, \mathbf{z}) = \psi\big((y_1, z_1)(y_2, z_2) \dots (y_n, z_n)\big) = \psi\big((y_1, y_2, \dots, y_n), (z_1, z_2, \dots, z_n)\big) = \\ = \nu(\mathbf{z}|\mathbf{y}) \cdot \mu(\mathbf{y}) = \nu(z_1, z_2, \dots, z_n | y_1, y_2, \dots, y_n) \cdot \mu(y_1, y_2, \dots, y_n). \end{aligned}$$

So we will work with three sources – the input source $\overline{\mathcal{S}}$, the output source $\mathcal{R} = \mathcal{R}(\mathcal{C}, \overline{\mathcal{S}})$ and the double source $\overline{\mathcal{D}}$. Fixate n and denote by $\mathbf{A}_n$, $\mathbf{B}_n$ the following partitions of the set $Y^n \times Z^n$:

$$\begin{aligned}&\{\mathbf{y}\}\times Z^n=\{(y_1,y_2,\dots,y_n)\}\times Z^n,\ \mathbf{y}=(y_1,y_2,\dots,y_n)\in Y^n,\quad \text{resp.,}\\&Y^n\times\{\mathbf{z}\}=Y^n\times\{(z_1,z_2,\dots,z_n)\},\ \mathbf{z}=(z_1,z_2,\dots,z_n)\in Z^n,\end{aligned}$$

i. e.,

$$\begin{aligned}\mathbf{B}_n &= \big\{\{\mathbf{y}\times Z^n\}\mid \mathbf{y}\in Y^n\big\}=\big\{\{(y_1,\dots,y_n)\}\times Z^n\mid (y_1,\dots,y_n)\in Y^n\big\}\\ \mathbf{A}_n &= \big\{\{Y^n\times\mathbf{z}\ \}\mid \mathbf{z}\in Z^n\big\}=\big\{Y^n\times\{(z_1,\dots,z_n)\}\mid (z_1,\dots,z_n)\in Z^n\big\}\end{aligned}$$

Further define the combined experiment $\mathbf{D}_n=\mathbf{A}_n\wedge\mathbf{B}_n$. It holds:

$$\begin{aligned}&\mathbf{D}_n=\{(\mathbf{y},\mathbf{z})\mid \mathbf{y}\in Y^n,\mathbf{z}\in Z^n\}=\\&=\{((y_1,y_2,\dots,y_n),(z_1,z_2,\dots,z_n))|(y_1,y_2,\dots,y_n)\in Y^n,(z_1,z_2,\dots,z_n)\in Z^n\}\,.\end{aligned}$$

The answer about the result of the experiment $\mathbf{B}_n$ tells us what word was transmitted. We cannot know this answer on the receiving end of the channel. What we know is the result of the experiment $\mathbf{A}_n$. Every particular result $Y^n\times\{z_1,z_2,\dots,z_n\}$ of the experiment $\mathbf{A}_n$ will change the entropy $H(\mathbf{B}_n)$ of the experiment $\mathbf{B}_n$ to the value $H(\mathbf{B}_n|Y^n\times\{z_1,z_2,\dots,z_n\})$. The mean value of entropy of the experiment $\mathbf{B}_n$ after executing the experiment $\mathbf{A}_n$ is $H(\mathbf{B}_n|\mathbf{A}_n)$. The execution of the experiment $\mathbf{A}_n$ changes the entropy $H(\mathbf{B}_n)$ to $H(\mathbf{B}_n|\mathbf{A}_n)$. The difference $H(\mathbf{B}_n)-H(\mathbf{B}_n|\mathbf{A}_n)=I(\mathbf{A}_n,\mathbf{B}_n)$ is the mean value of information about the experiment $\mathbf{B}_n$ obtained by executing the experiment $\mathbf{A}_n$.

By the formula (2.35), theorem 2.13 (page 47) it holds:

$$I(\mathbf{A},\mathbf{B})=H(\mathbf{A})+H(\mathbf{B})-H(\mathbf{A}\wedge\mathbf{B})$$

For our special case:

$$I(\mathbf{A}_n,\mathbf{B}_n)=H(\mathbf{A}_n)+H(\mathbf{B}_n)-H(\mathbf{D}_n)$$

We know that it holds for the entropy of input source $\overline{\mathcal{S}}$, output source $\mathcal{R}(\mathcal{C},\overline{\mathcal{S}})$ and double source $\overline{\mathcal{D}}$:

$$\begin{aligned}
H(\overline{\mathcal{S}}) &= \lim_{n\to\infty}\frac{1}{n}\cdot H(\mathbf{B}_n)\\
H(\mathcal{R}) &= \lim_{n\to\infty}\frac{1}{n}\cdot H(\mathbf{A}_n)\\
H(\mathcal{D}) &= \lim_{n\to\infty}\frac{1}{n}\cdot H(\mathbf{D}_n)
\end{aligned}$$

The entropy of a source was defined as the limit of the mean value of information per character for very long words. Similarly we can define $I(\overline{\mathcal{S}},\mathcal{R})$ the amount of transferred information per character transferred through the channel $\mathcal{C}$ as

$$I(\overline{\mathcal{S}},\mathcal{R}) = \lim_{n\to\infty}\frac{1}{n}\cdot I(\mathbf{A}_n,\mathbf{B}_n) = H(\overline{\mathcal{S}}) + H(\mathcal{R}) - H(\mathcal{D}).$$

We can see that the mean value of transferred information per character depends not only on properties of the channel but also on properties of the input source.

5.6 Channel capacity

The following approach to the notion of channel capacity was taken from the book [5]. Another approach with analogical results can be found in the book [9].

The channel capacity can be defined in three ways:

- by means of the maximum amount of information transferable through the channel
- by means of the maximum entropy of the source whose messages the channel is capable to transfer with an arbitrary small risk of failure
- by means of the number of reliable transferred sequences

We will denote these three types of capacities by $C_1,\ C_2,\ C_3$.

Channel capacity C_1 of the first type

The channel capacity of the first type is defined as follows:

$$C_1(\mathcal{C}) = \sup_{\overline{\mathcal{S}}} I(\overline{\mathcal{S}}, \mathcal{R}(\mathcal{C}, \overline{\mathcal{S}})),$$

where the supremum is taken over the set of all sources with the alphabet Y.

Channel capacity C_2 of the second type

Before defining the capacity of the second type we need to define what does it mean that "the messages from the source $\mathcal{S}$ can be transmitted through the channel $\mathcal{C}$ with an arbitrary small risk of failure".
In the case that input and output alphabets of the channel $\mathcal{C}$ are the same, i. e., if $Y = Z$, we can define by several ways a real function $\mathbf{w}$ with domain $Y^n \times Z^n$ which returns a real number $\mathbf{w}(\mathbf{y}, \mathbf{z})$ expressing the difference of words $\mathbf{z}$ and $\mathbf{y}$ for every pair of words $\mathbf{y} = y_1 y_2 \dots y_n \in Y^n$, $\mathbf{z} = z_1 z_2 \dots z_n \in Z^n$. Such function is called **weight function**. We will use two weight functions $\mathbf{w_e}$ and $\mathbf{w_f}$ defined as follows:

$$\begin{aligned} \mathbf{w_e} &= \begin{cases} 0 & \text{if } \mathbf{y} = \mathbf{z} \\ 1 & \text{otherwise} \end{cases} \\ \mathbf{w_f} &= \frac{d(\mathbf{y}, \mathbf{z})}{n}, \quad \text{where } d \text{ is the Hamming distance (definition 4.6, page 84).} \end{aligned}$$

Suppose we have a channel $\mathcal{C} = (Y, Z, \nu)$ with a source $\mathcal{S} = (Y^*, \mu)$, let $\mathbf{w}$ be a weight function. Then we can evaluate the quality of the transmission of messages from the source $\mathcal{S}$ through the channel $\mathcal{C}$ by the mean value of the weight function $\mathbf{w}$ for input and output words of the length n:

$$\mathbf{r_n}(\mathcal{S}, \mathcal{C}, \mathbf{w}) = \sum_{\mathbf{y} \in Y^n} \sum_{\mathbf{z} \in Z^n} \mathbf{w}(\mathbf{y}, \mathbf{z}) \cdot \nu(\mathbf{z}|\mathbf{y}) \cdot \mu(\mathbf{y}).$$

In the case of complete transmission chain we have a source $\mathcal{S}_X = (X^*, \phi)$ whose words in alphabet X are encoded by the mapping $h : X^* \to Y^*$ into words in alphabet Y. We get the source (Y^*, μ) where $\mu(\mathbf{y}) = 0$ if there is no word $\mathbf{x} \in X^*$ such that $\mathbf{y} = h(\mathbf{x})$, otherwise $\mu(\mathbf{y}) = \phi(h^{-1}(\mathbf{x}))$. The words from the source (Y^*, μ) appear after transmission through the channel $\mathcal{C}$ on its output as words in alphabet Z and these words are finally decoded by mapping $g : Z^* \to X^*$ into the words in the original alphabet X.
The transmission of the word $\mathbf{x} \in X^n$ will be as follows:

$$\begin{aligned}\mathbf{x} \in X^n \to \mathbf{y} = h(\mathbf{x}) \in Y^n \to\ & \text{input of channel } \mathcal{C} \to \\ \to\ & \text{output of channel } \mathcal{C} \to \mathbf{z} \in Z^n \to g(\mathbf{z}) \in X^n\end{aligned}$$

After transmitting the word $\mathbf{x} \in X^n$ we receive the word $g(\mathbf{z})$ and we assess the eventual difference of transmitted and received word as $\mathbf{w}(\mathbf{x}, g(\mathbf{z}))$. The total quality of transmission can be calculated:

$$\begin{aligned}\mathbf{r_n}(\mathcal{S}_X, h, \mathcal{C}, g, \mathbf{w}) &= \sum_{\mathbf{x}\in X^n} \sum_{\mathbf{z}\in Z^n} \mathbf{w}(\mathbf{x}, g(\mathbf{z})) \cdot \nu(\mathbf{z}|h(\mathbf{x})) \cdot \mu(h(\mathbf{x})) \\ &= \sum_{\mathbf{x}\in X^n} \sum_{\mathbf{z}\in Z^n} \mathbf{w}(\mathbf{x}, g(\mathbf{z})) \cdot \nu(\mathbf{z}|h(\mathbf{x})) \cdot \phi(\mathbf{x}) \ .\end{aligned}$$

The value $\mathbf{r}_n$ is called **risk of failure**. If the risk of failure is small the transmission of words of the length n is without a large number of errors. On the contrary, if the risk of failure is large many errors occur during the transmission of the words of the length n.

Definition 5.2. We say that the messages from the source $\mathcal{S}_X = (X, \phi)$ can be transmitted through the channel $\mathcal{C} = (Y, z, \nu)$ **with an arbitrary small risk of failure** with respect to given weight function $\mathbf{w}$ if for arbitrary $\varepsilon > 0$ there exists n, and encoding and decoding functions h and g, such that

$$\mathbf{r_n}(\mathcal{S}_X, h, \mathcal{C}, g, \mathbf{w}) < \varepsilon \ .$$

Definition 5.3. Define

$$C_2^e(\mathcal{C}) = \sup_{\mathcal{S}} H(\mathcal{S}), \qquad C_2^f(\mathcal{C}) = \sup_{\mathcal{S}} H(\mathcal{S}),$$

where supremum is taken over the set of all sources which can be transmitted through the channel $\mathcal{C} = (Y, z, \nu)$ with an arbitrary small risk of failure with respect to the weight function $\mathbf{w} = \mathbf{w}_e$ for C_2^e, and $\mathbf{w} = \mathbf{w}_f$ for C_2^f.

Channel capacity of the third type

The definition of the channel capacity of the third type makes use of the following notion of ε-distinguishable set of words.

Definition 5.4. The set $U \subseteq Y^n$ of input words is **ε-distinguishable**, if there exists a partition $\{Z(\mathbf{u}) : \mathbf{u} \in U\}$ of the set Z^n such that:

$$\nu(Z(\mathbf{u})|\mathbf{u}) \geq 1 - \varepsilon.$$

Remember that the partition $\{Z(\mathbf{u}) : \mathbf{u} \in U\}$ is a system of subsets of the set Z^n such that it holds:

1. If $\mathbf{u},\ \mathbf{v} \in U$, $\mathbf{u} \neq \mathbf{w}$ then $Z(\mathbf{u}) \cap Z(\mathbf{v}) = \emptyset$
2. $\bigcup_{\mathbf{u} \in U} Z(\mathbf{u}) = Z^n$.

The number $\nu(Z(\mathbf{u})|\mathbf{u})$ is the conditional probability of the event that the received word is an element of the set $Z(\mathbf{u})$ given the word $\mathbf{u}$ was transmitted. If the set $U \subseteq Y^n$ is ε-distinguishable and the received word is an element of the set $Z(\mathbf{u})$, we know that the probability of transmitting the word $\mathbf{u}$ is $1 - \varepsilon$ provided that only words from the set U can be transmitted.

Denote by $d_n(\mathcal{C}, \varepsilon)$ the maximum number of ε-distinguishable words from Y^n, where $\mathcal{C}$ is a channel, n a natural number and $\varepsilon > 0$.

The the third type of channel capacity $C_3(\mathcal{C})$ is defined

$$C_3(\mathcal{C}) = \inf_{\varepsilon} \limsup_{n \to \infty} \frac{1}{n} \log_2 d_n(\mathcal{C}, \varepsilon).$$

It can be shown that for most types of channels it holds:

$$C_1(\mathcal{C}) = C_2^e(\mathcal{C}) = C_2^f(\mathcal{C}) = C_3(\mathcal{C}),$$

what implies that all channel capacities were defined purposefully and reasonably.

5.7 Shannon's theorems

In this section we will suppose that we have a source $\mathcal{S}$ with entropy $H(\mathcal{S})$ and a communication channel $\mathcal{C}$ with capacity $C(\mathcal{C})$.

Theorem 5.1 (Direct Shannon theorem)**.** *If for a stationary independent source $\mathcal{S}$ and for a stationary independent channel $\mathcal{C}$ it holds:*

$$H(\mathcal{S}) < C(\mathcal{C}),$$

then the messages from the source $\mathcal{S}$ can be transmitted through the channel $\mathcal{C}$ with an arbitrary small risk of failure.

Theorem 5.2 (Reverse Shannon theorem)**.** *If for a stationary independent source $\mathcal{S}$ and for a stationary independent channel $\mathcal{C}$ it holds:*

$$H(\mathcal{S}) > C(\mathcal{C}),$$

then the messages from the source $\mathcal{S}$ cannot be transmitted through the channel $\mathcal{C}$ with an arbitrary small risk of failure.

Shannon's theorems hold for much more general types of channels and sources – namely for ergodic sources and ergodic channels. Shannon's theorems show that the notions of information, entropy of source and channel capacity were defined reasonably and these notions hang together closely.

The proofs of Shannon theorems can be found in the book [9] or, some of them, in the book [3].

Index

Bibliography

[1] Adámek, J.: *Kódování*, SNTL Praha, 1989

[2] Berlehamp, R., R.: *Algebraic Coding Theory*, McGraw-Hill, New York, 1968 (*Russian translation: Algebrajicheskaja teorija kodirovanija, Mir, Moskva, 1971*)

[3] Billingsley, P.: *Ergodic Theory and Information*, J. Willey and Sons, Inc., New York, London, Sydney, 1965 (*Russian translation: Ergodicheskaja teorija i informacija, Mir, Moskva, 1969*)

[4] Černý, J., Brunovský, P.: *A Note on Information Without Probability*, Information and Control, pp. 134 - 144, Vol. 25, No. 2, June, 1974

[5] Černý, J.: *Entropia a informácia v kybernetike*, Alfa – vydavateľstvo technickej a ekonomickej literatúry, Bratislava, 1981

[6] Halmos, P., R.: *Measure Theory (Graduate Texts in Mathematics)*, Springer Verlag,

[7] Hankerson, D., Harris, G.,A., Johnson, O.,D., Jr.: *Introduction to Information Theory and Data Compression*, CRC Press LLC, 1998, ISBN 0-8493-3985-5

[8] Jaglom, A., M., Jaglom, I., M.: *Pravděpodobnost a informace*, SAV, Praha, 1964

[9] Kolesnik, V., D., Poltyrev, G., S.,: *Kurs teorii informacii*, Nauka, Moskva, 1982

[10] Neubrunn, T., Riečan, B.,: *Miera a integrál, Veda, Bratislava, 1981*

[11] Schulz, R., H.: *Codierungstheorie, Eine Einfuhrung*, Vieweg, Wiesbaden 1991, ISBN 3-528-06419-6