Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Элементы теории информации. Параметры и характеристики дискретных информационный систем

Студент: Стрелковская В. А.

ФИТ 3 курс 2 группа

Преподаватель: Хартанович А. А.

Минск 2025

# Теоретические сведения

**Цель**: приобретение практических навыков расчета и анализа параметров и информативных характеристик дискретных ИС.

Простейшая информационная система состоит из трех элементов: **источника сообщения, канала передачи сообщения и получателя сообщения.**

**Дискретный сигнал** (устройство или канал передачи) характеризуется конечным числом значений информационного параметра.

**Дискретные сообщения** состоят из последовательности дискретных знаков. Часто этот параметр принимает всего два значения (0 или 1). Сообщение или канал его передачи на основе этих двух значений сигнала называют **двоичным** или **бинарным.**

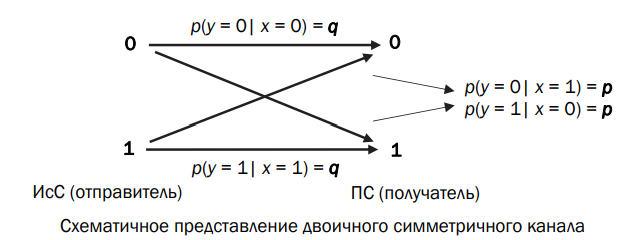
**Кодирование –** построение сигнала по определённым правилам, обеспечивающим соответствие между сообщением и сигналом. Или же **преобразование сообщения в сигнал**.

**Алфавит, А** – это общее число знаков или символов (N), используемых для генерации или передачи сообщений. Символы алфавита будем обозначать через {аi}, где 1 ≤ i ≤ N; N – мощность алфавита.

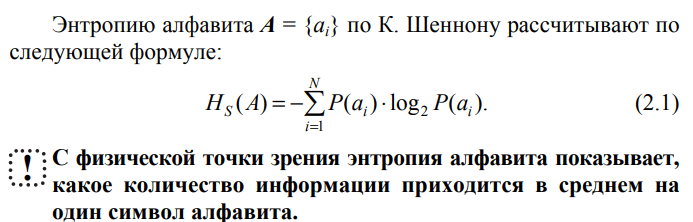
Минимальное число элементов алфавита Nmin = 2, А = {0, 1} – двоичный код. *Один дискретный знак* представляет собой **элементарное сообщение**, *последовательность знаков* – **сообщение**.

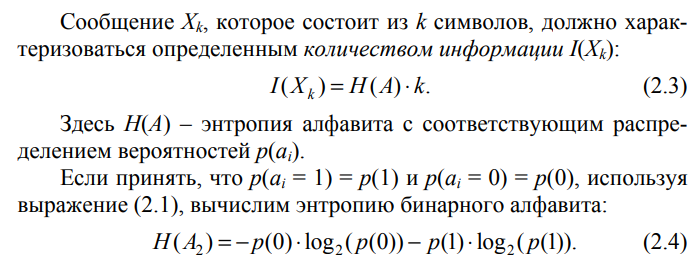
**Двоичный канал** передачи информации строится на основе двоичного алфавита: А = {0, 1}.

**Двоичный симметричный канал** (ДСК) – канал, в котором вероятности искажения переданного 0 (принята соответственно 1; этому событию соответствует условная вероятность р(1|0)) и переданной 1 (принят соответственно 0; этому событию соответствует условная вероятность р(0|1)) равны, как и равны вероятности передачи 0 (р(0)) и 1 (р(1)).

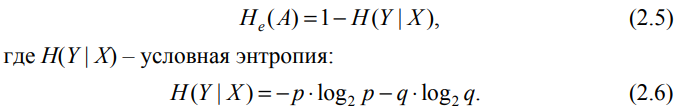


**Энтропия** – информационная характеристика алфавита.





Если вероятность ошибки в ДСК отлична от 0 (р > 0), переданное сообщение может содержать ошибки: Хk ≠ Yk. Количество информации в таком сообщении при его передаче по ДСК будет определяться не энтропией двоичного алфавита (в соответствии с выражением (2.3)), а эффективной энтропией Hе(A) алфавита или пропускной способностью канала:



# Практическое задание

# 2.1 Энтропия алфавита

Для вычисления энтропии вначале необходимо подсчитать количество появлений каждого символа в исходной строке. Для этого используется функция, представленная на рисунке 2.1.

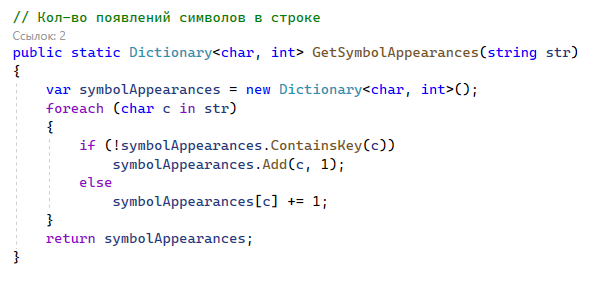


Рисунок 2.1 – Функция подсчёта количества появлений символов

Далее, для вычисления энтропии, необходимо перейти от количества появлений символов к вероятности их появления. Для этого необходимо кол-во появлений каждого символа разделить на кол-во символов в строке. Далее эту вероятность *P*(*ai*)можно подставлять в формулу (1.1). Функция, вычисляющая энтропию Шеннона, представлена на рисунке 2.2.

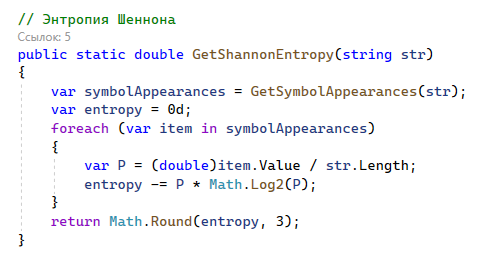


Рисунок 2.2 – Функция вычисления энтропии Шеннона

# Кириллица

В качестве кириллического алфавита был выбран казахский. Электронный текстовый документ содержится в файле kazakh.txt, представленном на рисунке 2.3.

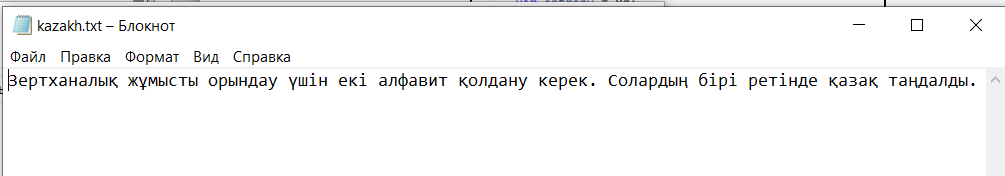


Рисунок 2.3 – Текст на казахском языке

# Латиница

В качестве алфавита на латинице был выбран испанский алфавит. Электронный документ содержится в файле spanish.txt, представленном на рисунке 2.4.

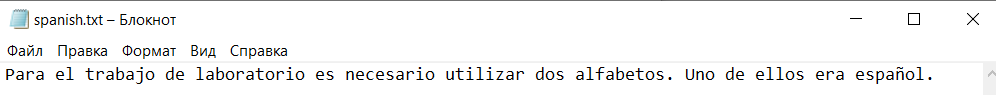


Рисунок 2.4 – Текст на испанском языке

# Вычисление энтропии алфавита

Графики, отражающие частоты появления символов в тексте на выбранных языках, представлены на рисунке 2.5.

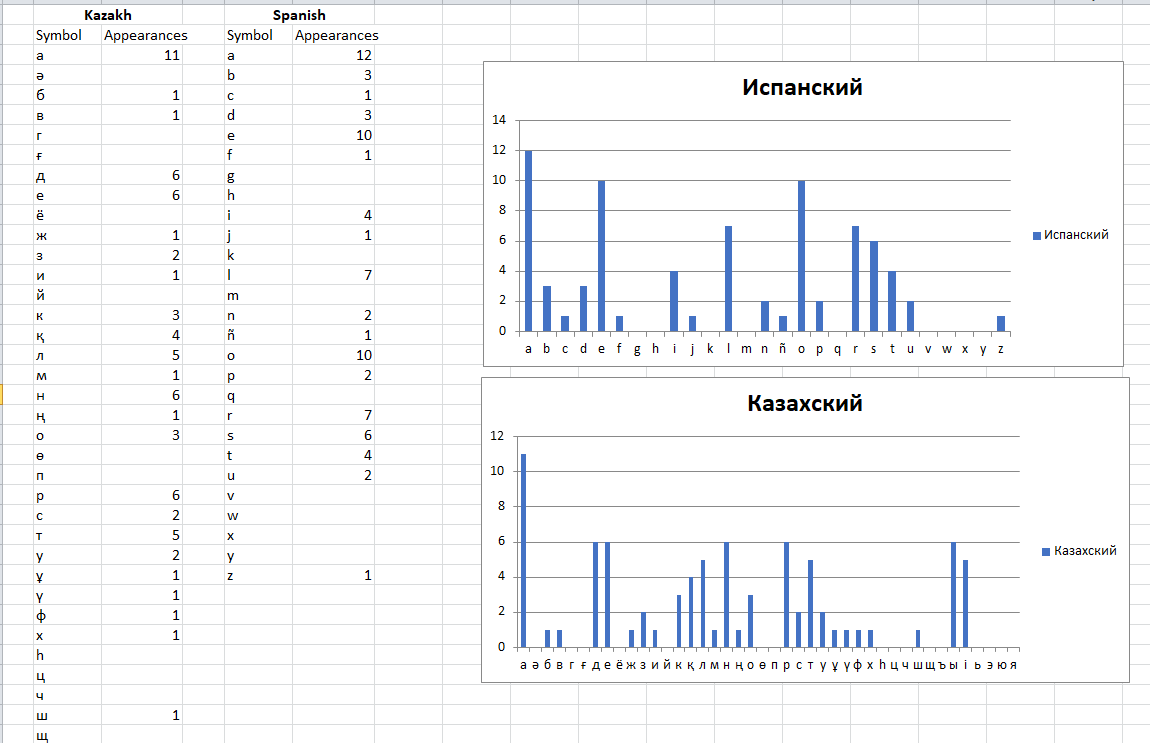


Рисунок 2.5 – Частоты появления символов в алфавитах

Из этих данных можно получить вероятности появления каждого символа *P*(*ai*)*.* При подстановке их в формулу получаем следующие значения энтропии для казахского, испанского и бинарного алфавитов, представленные на рисунке 2.6:

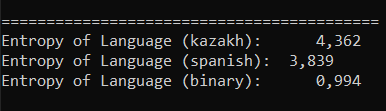


Рисунок 2.6 – Вывод функции подсчёта энтропии

# Количество информации

При известной энтропии алфавита количество информации вычисляется по следующей формуле:

(1.2)

То есть, необходимо умножить полученную энтропию на количество символов в сообщении. Программный код вычисления количества информации сообщения представлен на рисунке 2.7.

// Количество информации

public static double GetInformationAmount(string alphabet, string str)

{

if (IsBinaryAlphabet(alphabet))

return str.Length;

var informationAmount = GetShannonEntropy(alphabet) \* str.Length;

return Math.Round(informationAmount, 3);

}

Рисунок 2.7 – Функция количества информации

Если алфавит является бинарным, то за количество информации принимается длина строки, так как энтропия равняется 1. В противном случае, энтропия выбранного алфавита умножается на количество символов в сообщении, количество информации в котором мы желаем найти.

Вывод функции представлен на рисунке 2.8.

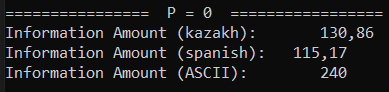
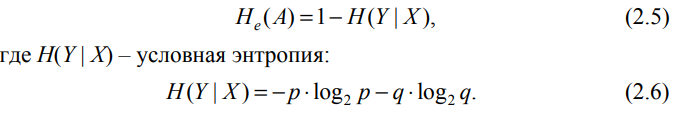


Рисунок 2.8 – Вывод количества информации

Как видно из данного рисунка, количество информации в сообщении, состоящем из ФИО в кодировке ASCII, равняется 240 битам, то есть равняется количеству символов в исходном сообщении, записанном в бинарном виде.

# Эффективная энтропия

Если сообщение может передаться ошибочно, то для вычисления количества информации используется формула эффективной энтропии:

Программный код функции для вычисления эффективной энтропии представлен на рисунке 2.9.

// Эффективная энтропия

public static double GetEffectiveEntropy(string alphabet, double p)

{

var q = 1 - p;

if (IsBinaryAlphabet(alphabet) && (p == 0 || q == 0))

return 1;

if (!IsBinaryAlphabet(alphabet) && p == 1)

return 0;

return 1 - (-p \* Math.Log2(p) - q \* Math.Log2(q));

}

Рисунок 2.9 – Функция эффективной энтропии

В бинарном алфавите эффективная энтропия и, соответственно, количество информации, будет равняться длине строки в случае, если вероятность ошибки равняется 0 или 1. В небинарных же алфавитах, при вероятности ошибки, равной 1, эффективная энтропия и количество информации будет равняться 0.

В таком случае, функция вычисления количества информации станет выглядеть следующим образом:

// Количество информации при наличии вероятности ошибки

public static double GetInformationAmount(string alphabet, string str, double p)

{

var informationAmount = GetShannonEntropy(alphabet) \* str.Length \* GetEffectiveEntropy(alphabet, p);

return Math.Round(informationAmount, 3);

}

Рисунок 2.10 – Функция кол-ва информации при наличии ошибок

Вывод функции вычисления количества информации при вероятности ошибки, отличной от нулевой, представлен на рисунке 2.11.

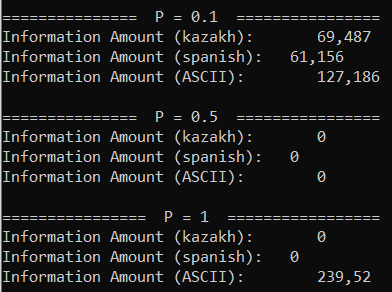


Рисунок 2.11 – Вывод кол-ва информации при *p>0*

**Вывод:** При вероятности ошибки *p=*1 количество информации в бинарном алфавите равняется количеству информации при *p=*0. Это связано с тем, что все биты исходного сообщения заменяются на обратные, поэтому из такого сообщения можно получить исходную информацию. В небинарном алфавите при *p=*1 определить исходное сообщение не возможно из-за того, что мощность алфавита больше 2, и каждый символ заменяется на некоторый неизвестный символ из алфавита, поэтому количество информации в таком случае равняется 0.