МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Белгородский государственный технологический университет им. В.Г. Шухова

ТЕОРИЯ ИНФОРМАЦИИ

Методические указания к выполнению практических заданий и индивидуальных домашних заданийдля студентов специальности 10.05.03 (090303) - Информационная безопасность автоматизированных систем

Белгород

2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Белгородский государственный технологический университет им. В.Г. Шухова
Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Утверждено
научно-методическим советом
университета

ТЕОРИЯ ИНФОРМАЦИИ

Методические указания к выполнению практических заданий и индивидуальных домашних заданийдля студентов специальности 10.05.03 (090303) - Информационная безопасность автоматизированных систем

Белгород
2015

УДК 004.5
ББК 32.811.4
Т33

Составители: канд. физ.-мат. наук, доц. Е.Н. Сергиенко
Иванова К. С.
Вожакова Ю.В.

Рецензент д-р техн. наук, проф., В.Г. Рубанов

Т33 **Теория** информации: методические указания к выполнению практических заданий и индивидуальных домашних заданий для студентов специальности 10.05.03 (090303) — Информационная безопасность автоматизированных систем / сост.: Е.Н. Сергиенко, К.С. Иванова, Ю.В. Вожакова, – Белгород: Изд-во БГТУ, 2015. – 88с.

Методические указания содержат теоретические сведения по каждому разделу, образцы решения задач, задачи для самостоятельного решения. Приведены рекомендуемые темы ИДЗ и методические рекомендации к их выполнению.

Методические указания предназначены для студентов специальности 10.05.03 (090303) — Информационная безопасность автоматизированных систем.

Данное издание публикуется в авторской редакции.

УДК 004.5
ББК 32.811.4

Содержание

ВВЕДЕНИЕ

Рабочая программа курса теории информации предусматривает выполнение девятилабораторных работ и одного ИДЗ. Цельметодических указаний – организовать самостоятельную работу студента по выполнению практических заданий, помочь овладеть каждым этапом алгоритма реализуемого метода решения задач. Поэтому в данных методических указаниях задания подробно сформулированы и иногда прокомментированы.

Методические указания содержат теоретические сведения по каждому разделу, образцы решения задач, задачи для самостоятельного решения. В приложении приведены рекомендуемые темы ИДЗ, методические рекомендации по их выполнению, пример оформления отчета, а также полезные таблицы.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАДАНИЙ

Выполнение практических заданий предполагает предварительное изучение соответствующего раздела курса. Студенту необходимо повторить теоретические сведения по теме практических заданий.

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ №1-4

Цель практических занятий: познакомиться с такими понятиями, как количество информации и энтропия; изучить виды канальных матриц и марковские источники. Научиться вычислять количество информации, а также работать с различными видами канальных матриц и вычислять с их помощью характеристики источника и приемника информации.

Количественная оценка информации

Дискретным источником сообщений называют некоторое устройство, которое через определенные равные интервалы времени выдает (детерминированным или случайным образом) очередной символ (букву) сообщения, принадлежащий заданному конечному алфавиту.

Определение. Пусть*A – это некоторая случайная величина*. Количеством собственной информации (или собственной информацией), заключенной в событии$A = a_i$, называется число

$$I(a_i) = - \log_2 P(a_i) = -\log_2 p_i.$$

Равенство $I(a_i) = 0$, равносильно тому, что $P(a_i) = 1$.

Определение. Энтропией *сл. величины A* (или энтропией источника с данным алфавитом A и данным распределением вероятностей) называется математическое ожидание собственного количества информации:

$$H(A) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

Если $|A| = \infty$ и ряд (2) расходится, то $H(A) = +\infty$.

Замечания:

1) если $p_i = 0$, то полагаем, что $0 * \log_2 0 = 0$;
2) если $|A_i| = m$ и $p_i = 1/m$, то $H(A) = \log_2 m$ – мера Хартли;
3) так как энтропия зависит только от p_i, то

$H(A) = H(\vec{p}) = -\sum_{i=1}^{m} p_i \log_2 p_i$.

Свойства H(p)

1)$H(\vec{p})$ непрерывна, как функция p_i;

2$H(\vec{p})$ симметрична, то есть не зависит от перестановки переменных p_i;

3) $H(\vec{p}) = 0$ тогда и только тогда, когда одно $p_i = 1$, а остальные $p_j = 0$;

Примеры решения задач

Пример 1.1. Определить количество информации по Хартли системы, содержащей 123 символа.

Решение.

$I = \log_2 m;$

$I = \log_2 123 = 6{,}94 \sim 7.$

$123_{10} = 1111011_2.$

Таким образом, для передачи символа по каналу связи нужно 7 бит информации.

Пример 1.2. Указать наименьшее количество вопросов, позволяющих гарантированно угадать задуманное число *a* из множества, содержащего числа от 1 до 2^n,при ответах: «Да», «Нет».

Решение:

Пусть число n = 3. Тогда множество A чисел, среди которых находится число *a*, содержит $2^n = 2^3$ элементов:

A = {1, 2, 3, 4, 5, 6, 7, 8}.

В множестве A 8 элементов. Пусть задуманное число *a* = 5.

Зададим вопросы:

1. a меньше 5? Ответ: нет;

2. a меньше 7? Ответ: да;

3. a меньше 6? Ответ:да;

Ответ: a=5

Пусть мощность множества A равна 2^n.

В общем случае мы задаем вопросы так:

1. a меньше 2^{n-1}? Ответ: да или нет;

2. a меньше 2^{n-2}? Ответ: да или нет;

3. a меньше 2^{n-3}? Ответ: да или нет;

и т. д.

Чтобы осталось 2 числа, нужно сделать $n-1$ шаг. В итоге n вопросов гарантируют правильный ответ.

Пример 1.3. Известно, что одно из kвозможных сообщений, передаваемых равномерным двоичным кодом, несёт 5 бит информации. Чему равно k?

Решение:

$I = log_2 k$ = 5бит, следовательно, $k = 2^5 = 32$. Таким образом, возможно 32 сообщения, которые содержат 5 бит информации.

Пример 1.4. Чему равна энтропия системы, состоящей из двух элементов, каждый из которых может быть в двух состояниях?

Решение:

а) Пусть k= 2 (количество состояний),n = 2 (количество элементов системы): тогда $m = k^n = 4$

$H_{max} = log_2 m = log_2 k^n = log_2 2^2 = 2$ бит/символ

Пример 1.5. Генератор вырабатывает три частоты *f1, f2, f3*. В шифраторе частоты комбинируются по три частоты в кодовой комбинации. а) Чему равно максимальное количество комбинаций, составленных из этих частот? б) Чему равно количество информации на одну кодовую посылку этих комбинаций?

Решение:

а) Общее число Nнеповторяющихся сообщений, которое может быть составлено из алфавита m путем комбинирования по n символов в сообщении, находиться по формуле

$N = m^n$;

$m = 3$; $n=3$;

$N = 3^3 = 27$.

б) Неопределенность, приходящаяся на символ первичного алфавита, составленного из равновероятностных и взаимонезависимых символов находиться по формуле:

$I = log_2 N$

$I = log_2 27 = 3 log_2 3 = 4{,}7549$бит.

Пример 1.6. Сколькими способами можно передать положение фигур на шахматной доске? Чему равно количество информации в каждом случае?

Решение:

Можно пронумеровать все клетки шахматной доски и передавать номер клетки. Для этого потребуется 64 качественных признака. Для передачи номера клетки будем достаточно одного сообщения. При этом количество информации $I = log_2 64 = 6$ бит.

Указать на доске необходимую клетку можно и передавая ее координаты по горизонтали и вертикали. Для этого достаточно 8 качественных признаков (8 номеров по горизонтали и 8 по вертикали). В этом случае придется передавать 2 сообщения. Количество информации $I = 2log_2 8 = 2*3 = 6$ бит

Пример 1.7. Алфавит состоит из букв *A, B, C, D*. Вероятности появления букв равны соответственно *p(A)=p(B)=0,25; p(C)=0,34; p(D)=0,16*. Определить количество информации на символ сообщения, составленного из такого алфавита.

Решение:

По формуле $H = \sum_{i=1}^{m} p_i \, log_2 \frac{1}{p_i} = -\sum_{i=1}^{m} p_i log_2 \, p_i$ *бит/символ*, получаем

H= -(2*0,25log$_2$0,25 + 0,34log$_2$0,34 +0,16log$_2$0,16) = 1,95 бит/символ.

Пример 1.8. Определить количество и объем информации в тексте "Широка страна моя родная", если для передачи каждый символ кодируется 8 битами.

Решение:

Число принятых символов равно k = 24. Следовательно, объем передаваемой информации равен 24*8 = 192 бита. Количество информации (мощность русского алфавита m = 32):

а) для равновероятного алфавита $H = log_2 m = log_2 32 = 5$ бит/символ; $I = kH = 24*5 = 120$ бит.

б)для неравновероятного алфавита берется энтропия русского алфавита, которая равна 4,36 бит/символ. Тогда I = 24*4,36 = 104,64 бит.

Пример 1.9. В урне 5 красных, 8 чёрных и 10 белых шаров. Выбирают без возвращения 2шара и фиксируют их цвета (без учёта порядка). Случайная величина A есть пара цветов. Найти H(A).

Решение:

Найдем вероятности извлечения пар шаров различных цветов:

Количество возможных исходов равно число сочетаний из 23 по 2.

$$P(кк) = \frac{\frac{5!}{2!3!}}{\frac{23!}{2!21!}} = 10/253;$$

P(чч) = 28/253;
P(бб) = 45/253;
P(кч) = 40/253;
P(кб) = 50/253;
P(чб) = 80/253.

Теперь вычислим H(A):

$H(A) = -$ (10/253*log_2 10/253 + 28/253*log_2 28/253 +45/253*log_2 45/253 + 40/253*$log_2$40/253 +50/253*log_2 50/253 + 80/253*log_2 80/253) = 2,388 *бит/символ*.

Задачи

Задача 1.1. Указать наименьшее количество вопросов, позволяющих всегда угадать день рождения любого человека при ответах: «Да», «Нет».

Задача 1.2. Известно, что одно из *k*возможных сообщений, передаваемых равномерным двоичным кодом, несёт *3* бита информации. Чему равно *k*?

Задача 1.3. а)Чему равна энтропия системы, состоящей из двух элементов, каждый из которых может быть в трёх состояниях? б) Чему равна энтропия системы, состоящей из трёх элементов, каждый из которых может быть в четырёх состояниях? в) Чему равна энтропия системы, состоящей из четырёх элементов, каждый из которых может быть в трёх состояниях?

Задача 1.4. Генератор вырабатывает четыре частоты *f1, f2, f3, f4*. В шифраторе частоты комбинируются по три частоты в кодовой комбинации. а) Чему равно максимальное количество комбинаций, составленных из этих частот? б) Чему равно количество информации на одну кодовую посылку этих комбинаций?

Задача 1.5. Алфавит состоит из букв *A, B, C, D*. Вероятности появления букв равны соответственно *p(A)=p(B)=0,35; p(C)=0,14; p(D)=0,26*. Определить количество информации на символ сообщения, составленного из такого алфавита.

Задача 1.6.В алфавите три буквы *A, B, C*. а) Составить максимальное количество сообщений, комбинируя по три буквы в

сообщении. б) Какое количество информации приходится на символ первичного алфавита?

Задача 1.7. Какое количество информации приходится на букву алфавита, состоящего из *16*; *25*; *32* букв?

Задача 1.8. Число символов алфавита $m=5$. Определить количество информации на символ сообщения, составленного из этого алфавита:

а) если символы алфавита встречаются с равными вероятностями;

б) если символы алфавита встречаются в сообщении с вероятностями $p1=0,8;\ p2=0,15;\ p3=0,03;\ p4=0,015;\ p5=0,005$.

Насколько недогружены символы во втором случае?

Задача 1.9. Чему равна энтропия системы, состояние которой описывается дискретной величиной со следующим распределением вероятностей:

x_i	x_1	x_2	x_3	x_4
p_i	*0,1*	*0,2*	*0,3*	*0,4*

Задача 1.10. Вероятность появления события при данном количестве опытов равна p, вероятность непоявления события $q = 1 - p$. При каком значении qрезультат будет обладать максимальной неопределённостью?

Задача 1.11. Чему равно количество информации при получении *8* сообщений равномерного четырёхзначного троичного кода?

Задача 1.12. Чему равно количество информации о неисправности nтранзисторов после температурных испытаний партии транзисторов из Nштук, выпущенное в один и тот же день, одним и тем же заводом?

Задача 1.13. На ВЦ постоянная информация хранится в *32768* стандартных ячейках. Сколькими способами можно передать сведения о том, из какой ячейки можно извлечь данные о состоянии информации? Чему равно количество информации в каждом случае? Какое геометрическое построение хранилища позволит передавать эту информацию минимальным количеством качественных признаков?

Задача 1.14. В плановом отделе работает три экономиста: два опытных и один неопытный. Для неопытного появление любого типа документа – равновероятно. Опытные специалисты знают, что сводки типа Sсоставляют *10%*общего количества документов,

поступающих в отдел. Определить, какое количество информации получит каждый экономист отдела при получении сводки типа *S*?

Задача 1.15. Определить объём и количество информации в данной фразе:

«Feci quod potui, faciantmeliorapotentes».

Задача 1.16. Чему равна вероятность появления комбинации *10110* при передаче пятизначных двоичных кодов? Чему равно среднее количество информации, приходящейся на одну комбинацию?

Задача 1.17. Сообщения составлены из равновероятного алфавита, содержащего *m=128*качественных признаков. Чему равно количество символов в принятом сообщении, если известно, что оно содержит *42* бита информации? Чему равна энтропия этого сообщения?

Задача 1.18. Физическая система может находиться в одном из четырёх состояний. Состояния системы заданы через вероятности следующим образом:

A	a_1	a_2	a_3	a_4
	0,25	*0,25*	*0,3*	*0,2*

Определить энтропию такой системы.

Задача 1.19. Определить энтропию физической системы *B*, которая может находиться в одном из *10* состояний. Вероятности состояний системы *B*:

B	b_1	b_2	b_3	*b4*	b_5	b_6	b_7	b_8	b_9	b_{10}
	0,01	*0,07*	*0,035*	*0,035*	*0,35*	*0,07*	*0,14*	*0,07*	*0,15*	*0,07*

Задача 1.20. В сообщении, составленном из *5* качественных признаков, которые используются с разной частотой, вероятности их появления равны соответственно: *p1=0,7; p2=0,2; p3=0,08; p4=0,015; p5=0,005*. Всего в сообщении принято *20* знаков. Определить количество информации во всём сообщении. Каким будет количество информации в данном сообщении, если все признаки будут иметь равную вероятность?

Задача 1.21. Определить количество информации в произвольном тексте: а) если символы алфавита равновероятны и взаимонезависимы; б) если символы алфавита неравновероятны. В каком случае количество информации может совпасть с объёмом?

Задача 1.22. На вычислительный центр с периферийного объекта необходимо передать определённую экономическую информацию, содержащуюся в таблицах с различными

показателями. Определить максимально возможный объём информации, которым может быть загружен канал связи, если таблиц *100*, они имеют *64* клетки, цифры, содержащиеся в таблицах, не более чем трёхзначные, а код, в котором передаются сообщения, – пятизначный двоичный.

Задача 1.23. Определить энтропию системы, состоящей из двух независимых подсистем. Первая подсистема состоит из трёх элементов, каждый из которых может находиться в двух состояниях с вероятностями *p1=0,6; p2=0,4*. Вторая подсистема состоит из двух элементов, каждый из которых может находиться в трёх состояниях с вероятностями *p1=0,1; p2=0,4; p3=0,5*.

Условная энтропия и энтропия объединения

Определение.Пусть *{A; B} – случайный вектор*. Собственной информацией, заключенной в событии $\{A = a_i; B = b_j\}$, называется число

$$I(a_i, b_j) = -log_2 P_{AB}(a_i; b_j) = -log_2 p_{ij}.$$

Определение. Пусть *{A; B} – случайный вектор*. Условной информацией, заключенной в событии $\{A = a_i/B = b_j\}$называется число

$$I(a_i/b_j) = -log_2 p(A = a_i/B = b_j) = -log_2 p(i/j)$$

Понятие условной энтропии в теории информации используется при определении взаимозависимости между символами кодируемого алфавита; для определения потерь при передаче информации по каналам связи; при вычислении энтропии объединения.

Во всех случаях при вычислении условной энтропии в том или ином виде используются условные вероятности.

$$p(A/B) = \frac{p(AB)}{p(B)}$$

$$p(B/A) = \frac{p(AB)}{p(A)}$$

Свойства

1. $I(a_i/b_j) = I(a_i)$ тогда и только тогда, когда события $A = a_i$ и $B = b_j$ независимы.
2. $I(a_i, b_j) = I(a_i) + I(b_j/a_i) = I(b_j) + I(a_i/b_j)$.

Определение.Пусть *{A; B} – случайный вектор*. Взаимной информацией между событиями $A = a_i$ и $B = b_j$называется число

$$\tilde{I}(a_i; b_j) = I(a_i) - I(a_i/b_j) = log_2\frac{p_{ij}}{p_i q_j}.$$

Свойства

1. Симметричности: $\tilde{I}(b_j, a_i) = I(b_j) - I(b_j/a_i) = - log_2 q_j - (- log_2\frac{pij}{pi}) = log_2\frac{pij}{pi} - log_2 q_j = log_2\frac{pij}{pi*qj} = \tilde{I}(a_i, b_j)$

2. $\tilde{I}(a_i: b_j) = 0$ тогда и только тогда, когда события $A = a_i$и $B = b_j$ независимы.

3. Аддитивности: $\tilde{I}(a_i, b_j) = I(a_i) + I(b_j) - I(a_i; b_j)$.

Определение. Пусть $b_j \in B$ и $q_j > 0$. Частной условной энтропией *сл. величины A* при условии события $B = b_j$ называют число

$$H(A/B = b_j) = -\sum_j p(a_i/b_j) log\,(p(a_i/b_j));$$

$$H(B/A = a_j) = -\sum_j p(b_i/a_j) log\,(p(b_i/a_j));$$

где индекс *i*выбран для характеристики произвольного состояния источника сообщений *A*, индекс *j* выбран для характеристики произвольного состояния адресата *B*.

Определение.Пусть *{A; B} – случайный вектор*. Энтропией совместного распределения, или совместной энтропией *сл. величин A и B* называется число

$$H(A; B) = -\sum_j\sum_i p_{ij} log_2 p_{ij}$$

Определение.Общей условной энтропией *сл. величины* A при условии *сл. величины B* называется число

$$H(A/B) = -\sum_i\sum_j p_j p(i/j)\, log_2\ p(i/j).$$

$$H(A/B) = -\sum_i\sum_j p_{ij}\, log_2\ p(i/j).$$

Понятие общей и частной условной энтропии широко используется при вычислении информационных потерь в каналах связи с шумами.

Определение. Средней взаимной информацией *сл. величины* A и *сл. величины B* называется число, равное математическому ожиданию случайной величины *Ĩ (A; B).*

$$\tilde{I}(A;B) = \sum\sum p_{ij} \log_2 \frac{p_{ij}}{p_i q_j} = H(A) - H(A/B) = H(B) - H(B/A) = H(A) + H(B) - H(A;B).$$

Примеры решения задач

Пример 2.1. В результате статистических испытаний установлено, что при передаче каждых 100 сообщений длиной по 5 символов в сообщении символ *К*встречается 50 раз, а символ *Т*– 30 раз. Вместе с символом *К*символ Твстречается 10 раз. Определить условные энтропии *H(K/T)* и *H(T/K).*

Решение.

Общее количество переданных символов

$$n = 100 \cdot 5 = 500.$$

Вероятность появления символа К:

$$p(K) = \frac{50}{500} = 0{,}1.$$

Вероятность появления символа Т:

$$p(T) = \frac{30}{500} = 0{,}06.$$

Вероятность совместного появления символа Ки Т:

$$p(KT) = \frac{10}{500} = 0{,}02.$$

Так как$p(KT) = p(T)p(K/T) = p(K)p(T/K)$, то условная вероятность появления символа К относительно символа Т

$$p(K/T) = \frac{p(KT)}{p(T)} = \frac{0{,}02}{0{,}06} = 0{,}33.$$

Условная вероятность появления символа Т относительно символа К

$$p(T/K) = \frac{p(KT)}{p(K)} = \frac{0{,}02}{0{,}1} = 0{,}2.$$

Условная энтропия символа Котносительно Т:

$$H(K/T) = -\{p(K/T)log_2 P(K/T) + [1 - p(K/T)]log_2[1 - p(K/T)]\}$$
$$== -(0{,}33log_2 0{,}33 + 0{,}67log_2 0{,}67)$$
$$= 0{,}9149\ бит/символ.$$

Условная энтропия появления символа Тотносительно К:

$$H(T/K) = -(0{,}2log_2 0{,}2 + 0{,}8log_2 0{,}8) = 0{,}7219 бит/символ.$$

Пример 2.2. Определить общую условную энтропию сообщений, составленных из алфавита А,В, если вероятности

появления символов в сообщении равны p(A)= 0,6; p(B) = 0,4. Условные вероятности переходов одного символа в другой равны p(B/A)= 0,15; p(A/B)= 0,1.

Решение.

$H(B/A) = -\sum_i \sum_j p(a_i)p(b_j/a_i)log_2 p(b_j/a_i) = -[0,6(0,85 \log_2 0,85 + 0,15\log_2 0,15) + 0,4(0,1 \log_2 0,1 + 0,9\log_2 0,9)] = 0,6 \cdot 0,6098 + 0,4 \cdot 0,4689 = 0,36588 + 0,18786 = 0,55374$ бит/символ.

Пример 2.3. При передаче 100 сигналов, соответствующих цифре 7, статистика принятых сигналов распределилась следующим образом: 7 – принята 90 раз, 5 – 4 раз, 9 – 3 раз, 10 – 2 раза, 4 – один раз. Чему равна энтропия источника?

Решение.

$H(A) = -\sum_i p(a_i) \log_2 p(a_i) = -(0,9 \log_2 0,9 + 0,04 \log_2 0,04 + 0,03 \log_2 0,03 + 0,02 \log_2 0,02 + 0,01 \log_2 0,01) = 0,6535$ бит/символ.

Пример 2.4. Задан закон распределения случайного вектора:

	b1	b2	b3
a1	7/24	1/24	0
a2	1/24	1/4	1/24
a3	0	1/24	7/24

Найти $H(A)$, $H(B)$, $H(A/B)$, $H(A;B)$.

Решение.

$p_i = \sum_j p_{ij}$;

$q_j = \sum_i q_{ij}$;

Получим распределение вероятностей для величин A и B:

A	a1	a2	a3
	1/3	1/3	1/3

B	b1	b2	b3
	1/3	1/3	1/3

$$H(\text{A}) = -\sum_j p(a_i) \log\big(p(a_i)\big).$$

H(A) = -1/3(3log$_2$1/3) = 1,58 бит.

H(B) = 1,58 бит.

$$H(A/B) = -\sum_j p(b_j) \sum_i \text{p}\big(a_i/b_j\big) \log\,(\text{p}(a_i/b_j)) =$$

$$= -\sum_i \sum_j \text{p}\big(a_i, b_j\big)\ log_2\left(p\big(a_i, b_j\big)\right) \text{ бит/два символа.}$$

$H(A/B) = -(2 * \frac{7}{24}\log_2\frac{7}{8} + 4 * \frac{1}{24}\log_2\frac{1}{8} + \frac{1}{4}\log_2\frac{3}{4}) = 0,71613$

$$H(A;\ B)\ = -\sum_j \sum_i p_{ij} log_2 p_{ij}$$

$H(A;B) = -(\frac{7}{24}\log_2\frac{7}{24} + \frac{1}{24}\log_2\frac{1}{24} + 0 + \frac{1}{24}\log_2\frac{1}{24} + \frac{1}{4}\log_2\frac{1}{4} +$
$124\log 2124 + 0 + 124\log 2124 + 724\log 2724) = 2{,}3011$

Канальная матрица

В общем случае, если мы передаём *m*сигналов *A*и ожидаем получить *m*сигналов *B*, влияние помех в канале связи полностью описывается со стороны источника *канальной матрицей*:

B / *A*	$\boldsymbol{b_1}$	$\boldsymbol{b_2}$	…	$\boldsymbol{b_j}$	…	$\boldsymbol{b_m}$
$\boldsymbol{a_1}$	$\mathrm{p}(b_1/a_1)$	$\mathrm{p}(b_2/a_1)$	…	$\mathrm{p}(b_j/a_1)$	…	$\mathrm{p}(b_m/a_1)$
$\boldsymbol{a_2}$	$\mathrm{p}(b_1/a_2)$	$\mathrm{p}(b_2/a_2)$	…	$\mathrm{p}(b_j/a_2)$	…	$\mathrm{p}(b_m/a_2)$
…	…	…	…	…	…	…
$\boldsymbol{a_i}$	$\mathrm{p}(b_1/a_i)$	$\mathrm{p}(b_2/a_i)$	…	$\mathrm{p}(b_j/a_i)$	…	$\mathrm{p}(b_m/a_i)$
…	…	…	…	…	…	…
$\boldsymbol{a_m}$	$\mathrm{p}(b_1/a_m)$	$\mathrm{p}(b_2/a_m)$	…	$\mathrm{p}(b_j/a_m)$	…	$\mathrm{p}(b_m/a_m)$

Вероятности, которые расположены по диагонали, определяют правильный приём, остальные – ложный. Значения цифр, заполняющих колонки канальной матрицы, обычно уменьшаются по мере удаления от главной диагонали и при полном отсутствии помех все, кроме цифр, расположенных на главной диагонали, равны нулю.

Если описывать канал связи со стороны источника сообщений, то прохождение данного вида сигнала в данном канале связи описывается распределением условных вероятностей вида $\mathrm{p}(b_j/a_i)$, так для сигнала a_1распределением вида

$$p(b_1/a_1); p(b_2/a_1); \ldots; p(b_i/a_1); p(b_m/a_1).$$

Сумма вероятностей распределения (3) всегда равна 1.

Потери информации, которые приходятся на долю сигнала a_i, описываются при помощи *частной условной энтропии* вида

$$H(b_j/a_i) = -\sum_j p(b_j/a_i)\log\,(p(b_j/a_i)).$$

Суммирование производится по j, так как i-е состояние остаётся постоянным.

Чтобы учесть потери при передаче всех сигналов по данному каналу связи, следует просуммировать все частные условные энтропии, т.е. произвести двойное суммирование по iи по j.

Общая условная энтропия

$$H(B/A) = -\sum_i p(a_i)\sum_j \mathrm{p}(b_j/a_i)\log\,(\mathrm{p}(b_j/a_i)).$$

Если мы исследуем канал связи со стороны приёмника сообщений, то с получением сигнала b_j предполагаем, что был послан какой-то из сигналов $a_1, a_2, \dots, a_i, \dots, a_m$. При этом канальная матрица будет иметь вид

B / A	$\boldsymbol{b_1}$	$\boldsymbol{b_2}$	…	$\boldsymbol{b_j}$	…	$\boldsymbol{b_m}$
$\boldsymbol{a_1}$	$\mathrm{p}(a_1/b_1)$	$\mathrm{p}(a_1/b_2)$	…	$\mathrm{p}(a_1/b_j)$	…	$\mathrm{p}(a_1/b_m)$
$\boldsymbol{a_2}$	$\mathrm{p}(a_2/b_1)$	$\mathrm{p}(a_2/b_2)$	…	$\mathrm{p}(a_2/b_j)$	…	$\mathrm{p}(a_2/b_m)$
…	…	…	…	…	…	…
$\boldsymbol{a_i}$	$\mathrm{p}(a_i/b_1)$	$\mathrm{p}(a_i/b_2)$	…	$\mathrm{p}(a_i/b_j)$	…	$\mathrm{p}(a_i/b_m)$
…	…	…	…	…	…	…
$\boldsymbol{a_m}$	$\mathrm{p}(a_m/b_1)$	$\mathrm{p}(a_m/b_2)$	…	$\mathrm{p}(a_m/b_j)$	…	$\mathrm{p}(a_m/b_m)$

В этом случае единице должны равняться суммы условных вероятностей по столбцам канальной матрицы$p(a_1/b_j) + p(a_2/b_j) + \cdots + p(a_i/b_j) + p(a_m/b_j) = 1$.

Частная условная энтропия:

$$H(A/b_j) = -\sum_{i=1}^{m} p(a_i/b_j)log\ (p(a_i/b_j)),$$

Общая условная энтропия:

$$H(A/B) = -\sum_j p(b_j) \sum_i p(a_i/b_j)\, log\ (p(a_i/b_j)).$$

Так как$p(a_i)p(b_j/a_i) = p(a_i, b_j)$,
то для вычисления общей условной энтропии наравне с выражением (6) может быть использовано следующее выражение:

$$H(A/B) = -\sum_i \sum_j p(a_i, b_j) \log (\mathrm{p}(a_i/b_j)).$$

Если заданы канальная матрица вида $\mathrm{p}(b_j/a_i)$и безусловные вероятности источника$p(a_i)$, то безусловные вероятности приёмника $p(b_j) = \sum_i p(a_i)\mathrm{p}(b_j/a_i)$, т.е. может быть вычислена энтропия приёмника

$$H(B) = -\sum_j p(b_j)\, log\left(p(b_j)\right).$$

Наоборот, если заданы вероятности вида $p(b_j)$ и канальная матрица, описывающая канал связи со стороны приёмника сообщений, то $p(a_i) = \sum_i p(b_j)\mathrm{p}(a_i/b_j)$, а значит может быть определена энтропия источника сообщений

$$H(A) = -\sum_j p(a_i)\, log\big(p(a_i)\big).$$

Энтропия объединения используется для вычисления энтропии совместного появления статистически зависимых сообщений. Например, передав сто раз цифру 5 по каналу связи с помехами, заметим, что цифра 5 была принята 90 раз, цифра 6 – 8 раз и цифра 4 – 2 раза. Неопределённость возникновения комбинаций вида 5 – 4, 5 – 5, 5 – 6, при передаче цифры 5 может быть описана при помощи энтропии объединения.$H(A, B)$ – неопределённость того, что будет послано *A*, а принято *B*. Для наборов переданных сообщений *A*и принятых сообщений *B*энтропия объединения представляет собой сумму вида

$$H(A, B) = -\sum_i \sum_j \mathrm{p}(a_i, b_j)\ log_2\left(p(a_i, b_j)\right) \text{ бит/два символа.}$$

Энтропия объединения и условная энтропия связаны между собой следующими соотношениями:

$$H(A, B) = H(A) + H(B/A) = H(B) + H(A/B),$$
$$H(B/A) = H(A, B) - H(A);\ H(A/B) = H(A, B) - H(B).$$

Энтропия объединения может быть подсчитана при помощи матрицы вида

(A, B)	$\boldsymbol{b_1}$	$\boldsymbol{b_2}$	…	$\boldsymbol{b_m}$
$\boldsymbol{a_1}$	$\mathrm{p}(a_1, b_1)$	$\mathrm{p}(a_1, b_2)$	…	$\mathrm{p}(a_1, b_m)$
$\boldsymbol{a_2}$	$\mathrm{p}(a_2, b_1)$	$\mathrm{p}(a_2, b_2)$	…	$\mathrm{p}(a_2, b_m)$
…	…	…	…	…
$\boldsymbol{a_m}$	$\mathrm{p}(a_m, b_1)$	$\mathrm{p}(a_m, b_2)$	…	$\mathrm{p}(a_m, b_m)$

Такая матрица обладает замечательным свойством:

$\sum_i \mathrm{p}(a_i, b_j) = \mathrm{p}(b_j)$; $\sum_j \mathrm{p}(a_i, b_j) = \mathrm{p}(a_i)$, при этом $\sum_i \mathrm{p}(a_i) = \sum_j \mathrm{p}(b_j) = 1$. Это свойство, в свою очередь, позволяет вычислять энтропию как источника, так и приёмника сообщений непосредственно по канальной матрице

$$H(A) = -\sum_i \sum_j \mathrm{p}(a_i, b_j) log \sum_j \mathrm{p}(a_i, b_j),$$

$$H(B) = -\sum_i \sum_j \mathrm{p}(b_j, a_i) log \sum_i \mathrm{p}(b_j, a_i).$$

Суммирование производим по *i* и *j*, так как для того, чтобы найти безусловные вероятности, необходимо суммировать их по одному индексу (имея в виду матричное представление вероятностей), а для нахождения *H* суммирование производится по другому индексу.

Условные вероятности вида $\mathrm{p}(a_i/b_j)$ и $\mathrm{p}(b_j/a_i)$вычисляются как

$$\mathrm{p}(a_i/b_j) = \frac{\mathrm{p}(a_i, b_j)}{\sum_i \mathrm{p}(a_i, b_j)};\ \mathrm{p}(b_j/a_i) = \frac{\mathrm{p}(a_i, b_j)}{\sum_j \mathrm{p}(a_i, b_j)}.$$

Количество информации на символ сообщения, переданного по каналу связи, в котором влияние помех описывается при помощи энтропии объединения, подсчитывается следующим образом:

$$I(A, B) = H(A) + H(B) - H(B, A).$$

Примеры решения задач

Пример 3.1.

Канал связи описывается следующей канальной матрицей со стороны источника:

$$\begin{matrix} 0{,}98 & 0{,}01 & 0{,}01 \\ 0{,}1 & 0{,}75 & 0{,}15 \\ 0{,}2 & 0{,}3 & 0{,}5 \end{matrix}$$

Также известны вероятности появления символов источника:

$p(a_1) = 0{,}2$; $p(a_2) = 0{,}1$; $p(a_3) = 0{,}7$.

Выполнить переход к канальной матрице со стороны приемника.

Решение:

Перейдем к матрице (A, B), умножив строки данной канальной матрицы наверояtности появления символов источника; получим матрицу

$$\begin{vmatrix} 0{,}196 & 0{,}002 & 0{,}002 \\ 0{,}01 & 0{,}075 & 0{,}015 \\ 0{,}14 & 0{,}21 & 0{,}35 \end{vmatrix}$$

Найдем вероятности появления символов приемника:

$p(b_1) = 0{,}196 + 0{,}01 + 0{,}14 = 0{,}346$;

$p(b_2) = 0{,}287$;

$p(b_3) = 0{,}367$.

Теперь осуществим переход к канальной матрице со стороны приемника:

Разделив каждую строку матрицы $\begin{vmatrix} 0{,}196 & 0{,}002 & 0{,}002 \\ 0{,}01 & 0{,}075 & 0{,}015 \\ 0{,}14 & 0{,}21 & 0{,}35 \end{vmatrix}$ на числа (0,346 0,287 0,376)

Соответственно получим матрицу A/B:

$$\begin{vmatrix} 0{,}566 & 0{,}007 & 0{,}007 \\ 0{,}029 & 0{,}261 & 0{,}062 \\ 0{,}405 & 0{,}732 & 0{,}931 \end{vmatrix}$$

Задачи(по темам 2-3)

Задача 2.1. В результате статистических испытаний установлено, что при передаче каждых 100 сообщений длиной по 8 символов в сообщении символ *К*встречается 70 раз, а символ Т– 45 раз. Вместе с символом *К*символ *Т*встречается 17 раз. Определить условные энтропии *H(K/T)* и *H(T/K).*

Задача 2.2. При передаче текстовых сообщений статистические наблюдения показали, что для слов со средней длиной в 6 букв на каждые 100 слов буква *А*встречается 80 раз, буква*В* встречается 50 раз, буквы Аи Ввместе встречаются 10 раз. Определить условную энтропию появления А, если в слове

присутствует B, и условную энтропию *B*, если в слове присутствует A.

Задача 2.3. Определить общую условную энтропию сообщений, составленных из алфавита A, B, если вероятности появления символов в сообщении равны $p(A)$=0,7; $p(B)$=0,3. Условные вероятности переходов одного символа в другой равны $p(B/A)$=0,25; $p(A/B)$=0,2.

Задача 2.4. Сообщения передаются двоичным кодом. В первом случае вероятности появления 0 и 1 равны соответственно p_0=0,8 и p_1=0,2. Помехи в канале связи отсутствуют, т.е. условные вероятности переходов 0 в 1 и 1 в 0 равны нулю. Во втором случае символы передаются с равными вероятностями $p_0=p_1$=0,5, однако в результате действия помех условные вероятности переходов равны $p(1/1)$=0,8; $p(1/0)$=0,2; $p(0/0)$=0,8; $p(0/1)$=0,2. Чему равна энтропия сообщений в первом и во втором случаях?

Задача 2.5. Чему равна условная энтропия сообщений, передаваемых по каналу связи, если канальная матрица имеет вид

$p(B/A)$			
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Задача 2.6. Влияние помех в канале связи описывается следующим распределением условных вероятностей:

$p(B/A)$		
0,98	0,01	0,01
0,15	0,75	0,1
0,3	0,2	0,5

Вычислить полную условную энтропию сообщений, передаваемых по данному каналу связи:

а) при равновероятном появлении символов в сообщении:

б) при вероятностях $p(a_1)$=0,7; $p(a_2)$=0,2; $p(a_3)$=0,1.

Задача 2.7.

а) Определить частные условные энтропии для каждого символа алфавита, если канал связи для передачи сообщений описывается следующей канальной матрицей:

$p(B/A)$			
0,9	0,1	0	0

0,05	0,94	0,01	0
0	0,01	0,98	0,01
0	0	0,1	0,9

б) Чему равна общая условная энтропия для сообщений, передаваемых по каналу связи, описанному приведённой выше канальной матрицей, если символы источника сообщений равновероятны?

в) Чему равна общая условная энтропия для сообщений, передаваемых по данному каналу связи, если распределение вероятностей появления символов на выходе источника имеет вид

$p(a_1)$	$p(a_2)$	$p(a_3)$	$p(a_4)$
0,15	0,32	0,25	0,28

Задача 2.8. Определить общую условную энтропию сообщений, передаваемых по каналу связи, который описывается следующей канальной матрицей:

p(A/B)			
0,9	0,1	0	0
0,05	0,84	0,01	0
0,03	0,06	0,98	0,1
0,02	0	0,01	0,9

Символы алфавита, из которого составлены сообщения, – равновероятны.

Задача 2.9. Составить произвольную канальную матрицу, описывающую канал связи: а) со стороны источника сообщений; б) со стороны приёмника. Показать процесс определения частных и общей условной энтропии для обоих случаев.

Задача 2.10. Определить энтропию приёмника сообщений, если канальная матрица

p(B/A)		
0,97	0,03	0
0,01	0,98	0,01
0	0,04	0,96

Вероятности появления символов на выходе источника сообщений равны

$p(a_1)$	$p(a_2)$	$p(a_3)$

0,5	0,3	0,2

Задача 2.11. Определить энтропию источника сообщений, если вероятность появления сигналов на входе приёмника

$p(b_1)$	$p(b_2)$	$p(b_3)$	$p(b_4)$
0,1	0,3	0,4	0,2

Канальная матрица имеет вид

$p(A/B)$			
0,99	0,02	0	0
0,01	0,98	0,01	0,01
0	0	0,98	0,02
0	0	0,01	0,97

Задача 2.12.При передаче сообщений по каналу связи с шумами была получена следующая статистика: частота f_1из 100 раз была принята 97 раз, 2 раза была принята частота f_2и 1 раз – частота f_3; при передаче f_2 98 раз принята f_2, два раза – f_1; При передаче f_3 96 раз принята f_3, два раза – f_2и два раза –f_4; при передаче f_4 99 раз принятаf_4и один раз – f_3.

а) Составить канальную матрицу, описывающую данный канал связи с точки зрения условий прохождения частот.

б) Определить общую условную энтропию сообщений, алфавитом которых являются частоты, если вероятности появления этих частот в передаваемых сообщениях соответственно равны:

$p(f_1)$	$p(f_2)$	$p(f_3)$	$p(f_4)$
0,37	0,3	0,23	0,1

в) Определить энтропию принятых сообщений.

Задача 2.13. Задана матрица вероятностей системы, объединённой в одну систему из двух взаимозависимых систем Ви А:

$p(A,B)$		
0,3	0	0
0,2	0,3	0,1
0	0,1	0

Определить полные условные энтропии $H(B/A)$ и $H(A/B)$.

Задача 2.14. Взаимодействие двух систем Аи Вописывается следующей матрицей:

$p(A,B)$		
0,4	0,1	0

0	0,2	0,1
0	0	0,2

Определить безусловную энтропию системы Аи системы В.

Задача 2.15. Сообщения создаются двумя источниками 1 и 2 и передаются по одному каналу связи. Известно, что на выходе источника *1*сигналы появляются с вероятностями *p(A)*=0,5; *p(B)*=0,333; *p(C)*=0,167. Условные вероятности появления сигналов *D, E, F* и Gисточника 2 при условии, что были переданы сигналы *A, B, C*источника *1*, соответственно равны:

p(D/A)=p(E/A)=p(F/A)=p(G/A)=0,25; p(D/B)=0,3;

p(E/B)=0,2; P(F/B)=0,2; p(G/B)=0,3; p(D/C)=0,166;

p(E/C)=0,5; p(F/C)=0,167; p(G/C)=0,167.

Определить совместную энтропию источников *1 и 2*, условную энтропию *H(2/1)*, энтропию второго источника, а также максимальное значение энтропии *H(1, 2).*

Задача 2.16. Источник сообщений *X*вырабатывает сообщения при помощи двух частот f_1 и f_2с вероятностями p_1=0,3; p_2=0,7. Второй источник *Y*создаёт сообщения при помощи частот f_3, f_4и f_5. Вероятности появления этих частот в сообщениях не известны, но известны условные вероятности их появления относительно частот первого источника

$$p(3/1) = 0,35;\ p(4/1) = 0,25;\ p(5/1) = 0,4;$$

$$p(3/2) = 0,4;\ p(4/2) = 0,3;\ p(5/2) = 0,3;$$

Определить: $H(Y/X)$, $H(X, Y)$, $H(X, Y)_{макс}$.

Задача 2.17 .При передаче 100 сигналов, соответствующих цифре 7, статистика принятых сигналов распределилась следующим образом: 7 – принята 90 раз, 5 – 4 раза, 9 – 3 раза, 10 – 2 раза, 4 – один раз. Чему равна неопределённость того, что при передаче 7 будет принята цифра 7?

Задача 3.1. Определить полные условные энтропии двух систем Аи В, если известна матрица вероятностей некоторой системы, полученной в результате объединения систем Аи В:

p(A,B)		
0,2	0	0
0,1	0,2	0
0	0,1	0,4

Найти также энтропию объединения H(A, B) и H(B, A).

Задача 3. 2. Чему равна энтропия источника сообщений H(A), энтропия приёмника H(B), энтропия объединения H(A, B), если канал связи описан следующей канальной матрицей:

p(A,B)		
0,1	0,1	0
0	0,2	0,1
0	0,2	0,3

Безусловные вероятности системы B:

$p(b_1)$	$p(b_2)$	$p(b_3)$
0,2	0,4	0,4

Задача3.3.УсловныевероятностисостоянийсистемыАотносительносистемыB: $p(b_1/a_1)=0,5$; $p(b_2/a_1)=0,5$; $p(b_2/a_2)=0,666$; $p(b_3/a_2)=0,334$; $p(b_2/a_3)=0,4$; $p(b_3/a_3)=0,6$; $p(b_3/a_1)=p(b_1/a_2)=p(b_1/a_3)=0$. Какойвидимеетматрицавероятностейобъединённойсистемы AB, еслибезусловныевероятностисистемыA: $p(a_1)=0,2$; $p(a_2)=0,3$; $p(a_3)=0,5$?

Задача 3.4. Определить все возможные информационные характеристики канала связи, в котором взаимосвязь источника с приёмником может быть описана матрицей вида

p(A,B)		
0,2	0,1	0
0,2	0	0,2
0,1	0,1	0,1

Марковские источники

Дискретным источником сообщений называют некоторое устройство, которое через определенные промежутки времени выдает (детерминированным или случайным образом) очередной символ (букву) сообщения, принадлежащий заданному конечному алфавиту.

*Модель источника*задана, если дан алфавит источника - конечное множество $A = (a_1, ..., a_m)$, и для любой конечной последовательности $(a_{i_1}, a_{i_2}, ..., a_{i_n})$ букв из алфавита Аопределена вероятность $P(a_n)$ появления этой последовательности на выходе источника.

Дискретный источник без памяти задается двумя параметрами $(A,\vec{p})$, где$A = (a_1, ..., a_m)$ - алфавит источника и $\vec{p} = (p_1, ..., p_m)$ - вероятностный вектор (т. е. $p_k = P(a_k)$, $k = 1, ..., m$).

Вероятность$P(a_n) = P(ai_1, ai_2, ..., ai_n) = P(ai_1)\ P(ai_2)\ ...\ P(ai_n) = pi_1pi_2...pi_n$. Таким образом, название *"источник без памяти"* означает, что очередная случайная буква A_kпорождается источником независимо от предшествующих букв A_1, A_2, A_{k-1}.

Простой марковский источник задается тремя параметрами $(A, \vec{p}, Q)$, где$A = (a_1, ..., a_m)$ - алфавит источника и $\vec{p} = (p_1, ..., p_m)$ - вероятностный вектор и $Q = (Q_{ij})_{mxm}$ - стохастическая матрица. Число $q_{ij} = P(a_j/a_i)$ равно вероятности появления символа a_j при условии, что предшествующим символом был символ a_i. Матрица Qимеет вид:

$$Q_m = \begin{pmatrix} q_{11} & \cdots & q_{1m} \\ \vdots & \ddots & \vdots \\ q_{m1} & \cdots & q_{mm} \end{pmatrix}$$

В первой строке стоят вероятности появления символов $a_1, ..., a_m$ при условии, что предыдущим символом был символ a_1. Поэтому $\sum_{j=1}^{m} q_{ij} = 1$ для $i = 1, ..., m$. Закон распределения вероятностей символов алфавита в первый (начальный) момент времени имеет вид $P_1(a_j) = p_j,\ j = 1, 2, ...\ , m$, т. е. совпадает с вероятностным вектором. Для второго момента $P_2(a_j) = \sum_{i=1}^{m} P(a_j/a_i)*P_1(a_i) = \sum_{i=1}^{m} q_{ij} * p_i = \sum_{i=1}^{m} p_i * q_{ij}$.

Например, при $j = 3$ имеем

$$P_2(a_3) = \sum_{i=1}^{m} p_i q_{i3} = p_1q_{13} + p_2q_{23} + p_3q_{33}.$$

Поэтому вероятностный вектор $\vec{p}^{(2)} = (p_2(a_1), ..., p_2(a_m))$ можно вычислять по формуле

$$\vec{p}^{(2)} = \vec{p}^{(1)}Q = \vec{p}Q.$$

Очевидно, что $\vec{p}^{(k)} = \vec{p}^{(1)}Q^{k-1}$.

Важным подклассом дискретных источников сообщений являются стационарные источники. Для стационарного источника $\forall\, \vec{p}^{(k)} = \vec{p}^{(1)}$, или $\vec{p} = \vec{p} * Q$, и наоборот. Дискретный источник без памяти всегда стационарен.

Не вдаваясь в обсуждение понятия «предельная энтропия», примем как факт, что

1) для дискретного источника без памяти предельная энтропия существует и равна

$$H = H(\vec{p}) = -\sum_{i=1}^{m} p_i log_2 p_i$$

2) для стационарного простого марковского источника предельная энтропия существует и равна

$$H = H^{(2)} = -\sum_{i=1}^{m} (p_i\ (\sum_{j=1}^{m} q_{ij} log_2 p_{ij}))$$

3)Для стационарного простого марковского источника$H^{(1)} = H^{(2)} = H^{(3)} = ... = H\infty$.

4) Энтропией на одну букву k-буквенного сообщения является величина

$H_k = (H^{(1)} + H^{(2)} + ... + H^{(k)})/k$,причем$H^{(1)} = H_1 = -\sum_{i=1}^{m} p_i log_2 p_i$.

Примеры решения задач

Пример 4.1. Для простого марковского источника*(A, $\vec{p}$, Q),* с заданной матрицей

$$Q = \begin{pmatrix} 0 & 3/4 & 1/4 \\ 1/3 & 1/6 & 1/2 \\ 0 & 2/5 & 3/5 \end{pmatrix}$$

найти вектор $\vec{p}$, для которого источник будет стационарным, и вычислить H_k, $H^{(k)}uH\infty$.

Решение.

Пусть вектор $\vec{p} = (x, y, z)$. Для стационарного источника требуется выполнение равенства $\vec{p}$ *Q = $\vec{p}$, или

$$\{x, y, z\} * \begin{pmatrix} 0 & 3/4 & 1/4 \\ 1/3 & 1/6 & 1/2 \\ 0 & 2/5 & 3/5 \end{pmatrix} = \{x, y, z\}.$$

Имеем:

(1/3*y*; 3/4*x* + 1/6*y* + 2/5*z*; 1/4*x* + 1/2*y* +3/5*z*) = *(x, y, z).*

Получаем систему и решаем ее:

$$\begin{cases} 1/3y = x; \\ 3/4x + 1/6y + 2/5z = y; \\ 1/4x + 1/2y + 3/5z = z; \end{cases}$$

$$\begin{cases} y = 3x; \\ 45x + 10y + 24z = 60y; \\ 5x + 10y + 12z; \end{cases}$$

$$\begin{cases} 45x + 30x + 24z = 180x \\ 5x + 30x + 12z = 20z \\ 8z = 35x \end{cases} \qquad \begin{cases} 24z = 105x \\ 35x = 8z \end{cases}$$

или$z = \frac{35}{8}x$; $y = 3x$. Но$x + y + z = 1$, поэтому

$x + 3x + \frac{35}{8}x = 1; 67x = 8$. Отсюда $x = \frac{8}{67}$; $y = \frac{24}{67}$; $z = \frac{35}{67}$, т.е. $\vec{p} = \left\{\frac{8}{67}; \frac{24}{67}; \frac{35}{67}\right\}$.

Далее получаем

$H^{(1)} = H_1 = -\sum_{i=1}^{3} p_i log_2 p_i = -\left(\frac{8}{67} log_2 \frac{8}{67} + \frac{24}{67} log_2 \frac{24}{67} + \frac{35}{67} log_2 \frac{35}{67}\right) \approx 1{,}393$

$H^{(2)} = -\sum_{i=1}^{3} p_i \left(\sum_{j=1}^{3} q_{ij}\, log_2 q_{ij}\right) = -(p_1 (\sum_{j=1}^{3} q_{1j} log_2 q_{1j}) + p_2 (\sum_{j=1}^{3} q_{2j} log_2 q_{2j}) + p_3 (\sum_{j=1}^{3} q_{3j} log_2 q_{3j})) = -\left(\frac{8}{67} (\frac{3}{4} log_2 \frac{3}{4} + 14log214 + 246713log213 + 16log216 + 12log212 + \frac{35}{67}(\frac{2}{5} log_2 \frac{2}{5} + \frac{3}{5} log_2 \frac{3}{5})\right) \approx 1{,}241.$

$H^{(3)} = H^{(4)} = \ldots = H^{(2)}$.

$H_\infty = H^{(2)} = 1{,}241$.

$H_k = \frac{1}{k}\,(H^{(1)} + (k-1)H^{(2)})$.

Задачи

Задача 4.1. Для простого марковского источника $(A, \vec{p}, Q)$ с заданной матрицей Q найти вектор $\vec{p}$, для которого источник будет стационарным, и вычислить H_k, $H^{(k)}$ и H_∞, если матрица Q равна

а) $\begin{pmatrix} p & 1-p \\ q & 1-q \end{pmatrix}$; *б)* $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} & 0 \end{pmatrix}$; *в)* $\begin{pmatrix} 0 & \frac{1}{5} & \frac{4}{5} \\ \frac{1}{2} & \frac{1}{10} & \frac{2}{5} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$;

г); $\begin{pmatrix} 1-p & \frac{p}{2} & \frac{p}{2} \\ \frac{p}{2} & 1-p & \frac{p}{2} \\ \frac{p}{2} & \frac{p}{2} & 1-p \end{pmatrix}$

$$д)\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \quad е)\begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{2} \end{pmatrix};$$

$$ж)\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}; \; з)\begin{pmatrix} \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & 0 & 0 \end{pmatrix};$$

$$и)\begin{pmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{2}{3} & 0 & \frac{1}{3} \end{pmatrix}; \quad к)\begin{pmatrix} \frac{7}{8} & \frac{1}{8} & 0 \\ \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix}.$$

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ №5-7

Цель практических занятий: познакомиться с понятием избыточности сообщений, изучить такие виды оптимального кодирования, как алгоритм Шеннона-Фано, алгоритм Хаффмана; научиться сжимать сообщения с их помощью.

Определение избыточности сообщений. Оптимальное кодирование

Пусть даны два конечных множества , $A = \{a_1, \ldots, a_m\}$ – алфавит источника сообщений, и $B = \{b_1, \ldots, b_D\}$ – кодовый алфавит. Обозначим через $B^* = \bigcup_{n\geq 1} B^n$ множество всех конечных последовательностей в алфавите B.

Определение. Алфавитным D-ичным кодированием будем называть произвольное отображение $\varphi: A \to B^*$. При этом образ $\varphi(a_i) \in B^*$ символа $a_i \in A$ назовем кодовым словом, или результатом кодирования символа a_i, а длину этого кодового слова обозначим $l_i = len(\varphi(a_i))$.

Набор кодовых слов $\varphi(A) = ((\varphi(a_1), \ldots, \varphi(a_m))$ будем называть D-ичным кодом для алфавита A. Если при этом все

кодовые слова имеют одинаковую длину, кодирование φ назовем равномерным, в противном случае – неравномерным.

Определение.Продолжением φ^* алфавитного кодирования φ на множество$A^* = \bigcup_{n\geq 1} A^n$ всех конечных последовательностей в алфавите A назовем отображение $\varphi^*: A^* \to B^*$, которое получается по правилу сцепления (приписывания) кодовых слов: для любого $a^n = (a_{i_1}, \ldots, a_{i_n})$ положим $\varphi^*(a_{i_1}, \ldots, a_{i_n}) = \varphi(a_{i_1})\ldots \varphi(a_{i_n})$.

Определение. Алфавитное кодирование φи набор кодовых слов $\varphi(A)$ называем

1) префиксным(суффиксным), если никакое кодовое слово $\varphi(a_i)$ не является началом (окончанием) какого-либо другого кодового слова $\varphi(a_j)$, $i \neq j$;

2) однозначно декодируемым, если отображение φ^* инъективно.

Замечание: если кодирование φ однозначно декодируемое, то для любой последовательности $a^n = (a_{i_1}, \ldots, a_{i_n})$ соответствующая кодовая последовательность $\varphi^*(a_{i_1}, \ldots, a_{i_n})$ единственным образом разбивается на кодовые слова $\varphi(a_{i_1})\ldots \varphi(a_{i_n})$ из кода $\varphi(A)$.

Пример.Пусть $m = D = 2$, $A = B = \{0, 1\}$. В следующей таблице приведены пять вариантов алфавитного кодирования и указано, являются они или нет префиксными, суффиксными и однозначно декодируемыми.

	φ_1	φ_2	φ_3	φ_4	φ_5
0	0	0	0	00	0
1	01	10	010	11	00
префиксное	-	+	-	+	-
суффиксное	+	-	-	+	-
однозначно декодируемое	+	+	+	+	-

Утверждение: если алфавитное кодирование φ является префиксным или суффиксным, то оно – однозначно декодируемо. Обратное неверно.

Корневое дерево Gназывается D-ичным, если любая его вершина имеет не более D потомков. Для такого дерева индуктивно определим D-ичную разметку, то есть сопоставим каждой вершине u ее метку $\mu(u)$, представляющую собой некоторое слово в алфавите $B = \{b_1, \ldots, b_D\}$. Вначале для каждой неконцевой вершины u произвольным образом пометим все исходящие из u ребра

различными символами алфавита B(это можно сделать, поскольку число потомков не превосходит D). По определению, метка $\mu(u_0)$ корня u_0 есть пустое слово. Если вершина u является предком вершины v, метка $\mu(u)$уже определена, а ребро (u, v) помечено символом $b_j \in B$, то метка $\mu(v)$ вершины v получается из метки $\mu(u)$ приписыванием справа символа b_j, $\mu(v) = \mu(u)b_j$. Очевидно, для вершин k-го уровня метка имеет длину k. Полученное в результате дерево с метками будем назвать размеченным.

Теорема.

1. Если G — D-ичное размеченное корневое дерево, то множество меток всех его листьев образует префиксный D-ичный код.

2. Если $\varphi: A \to B^*$— D-ичное префиксное кодирование, то существует такое D-ичное размеченное корневое дерево G, для которого множество меток всех листьев совпадает с кодом $\varphi(A)$.

Предположим, что дано алфавитное кодирование $\varphi: A \to B^*$ с длинами кодовых слов $len(\varphi(a_i)) = l_i$, $1 \le i \le m$, и правило кодирования φ^* применяется к последовательностям, порождаемым дискретным источником без памяти $(A, \vec{p})$, $\vec{p} = (p_1 \ldots, p_m)$. Далее в этом пункте считаем, что распределение $\vec{p} = (p_1 \ldots, p_m)$ задано и фиксировано.

Определение. Пусть задан алфавит источника $A = \{a_1, \ldots, a_m\}$, кодовый алфавит $B = \{b_1, \ldots, b_D\}$
и распределение $\vec{p} = (p_1 \ldots, p_m)$. Тогда средней длиной кодового слова при алфавитном кодировании $\varphi: A \to B^*$ (или, коротко, средней длиной кода $\varphi(A)$) называется величина

$$l^{\varphi} = \sum_{i=1}^{m} p_i l_i.$$

Определение. Алфавитное кодирование $\varphi: A \to B^*$ (и код $\varphi(A)$) называется оптимальным, если φ однозначно декодируемо и при этом средняя длина l^{φ} минимальна.

Утверждение. Оптимальное кодирование φ существует.

Теорема. Если алфавитное кодирование φ однозначно декодируемо, то справедливо неравенство

$$l^{\varphi} \ge \frac{H(\vec{p})}{log_2 D},$$

причём равенство имеет место тогда и только тогда, когда все положительные координаты вектора $\vec{p}$ имеют вид $p_i = D^{-l_i}$.

Теорема. Если распределение невырожденное, то существует такое префиксное кодирование φ, для которого справедливо неравенство

_____

Следствие. Средняя длина оптимального алфавитного кодирования φ удовлетворяет неравенствам

_____ _____

Всюду далее предполагаем, что на алфавите источника сообщений $A = \{a_1, \ldots, a_m\} (m \geq 2)$ задано распределение вероятностей $= (p_1 \ldots, p_m)$, причём вероятности символов упорядочены в невозрастающем порядке, т. е. $p_1 \geq p_2 \geq \ldots \geq p_m$.

Алгоритм Фано (D = 2, B ={0,1})

Выберем число k, $1 \leq k < m$, так, чтобы величина

была минимальной. Разобьем множество $A = \{ , \ldots, \}$ на подмножества: A_0 , где $A_0 = \{a_1, \ldots, a_k\}$, $A_1 = \{a_{k+1}, \ldots, a_m\}$.

Следующие шаги алгоритма определим индуктивно. Предположим, что уже задано подмножество $\subseteq A$,

Если подмножество состоит из единственного символа a_j, то для этого символа определяем кодирование $\varphi(a_j) = i_1 \ldots i_t$. Если же в подмножестве $= \{a_j, \ldots, a_s\}$ не менее двух символов, то выберем число k, j , так, чтобы минимизировать величину , и разобьем множество на подмножества $= \{a_j, \ldots, a_k\}$ и $= \{a_{k+1}, \ldots, a_s\}$.

Процесс разбиения на подмножества продолжается, пока не получим все одноэлементные подмножества и тем самым не определим кодирование φ всюду на алфавите A.

Замечание. В описанном алгоритме имеется неоднозначность в выборе значения k, поскольку указанное выше минимальное значение модуля разности сумм вероятностей может

достигаться при различных k. Можно принять соглашение о том, что в случае такой неоднозначности выбирается, например, наименьшее возможное значение k.

Замечание. Очевидно, что в результате применения алгоритма Фано получается префиксное кодирование, так как фактически происходит построение кодового дерева от корня A к листьям — одноэлементным подмножествам $A_{i_1,\ldots,i_t}$ (Рис. 1).

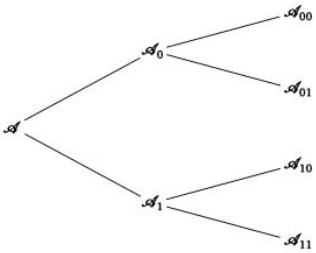


Рис. 1

Пример.Рассмотрим алфавитA = {a, b, c, d, e} и распределение вероятностей $\vec{p} = (0{,}3;\ 0{,}2;\ 0{,}2;\ 0{,}2;\ 0{,}1;)$. Последовательные разбиения подмножеств в алгоритме Фано можно наглядно представить диаграммой (Рис. 2).

a	0,3	0	0	
b	0,2	0	1	
c	0,2	1	0	
d	0,2	1	1	0
e	0,1	1	1	1

Рис.2

В результате получается кодирование $\varphi(a)$ = 00, $\varphi(b)$ = 01, $\varphi(c)$ = 10, $\varphi(d)$ = 110, $\varphi(e)$ = 111 со средней длиной $l^{\varphi} = (0{,}3 + 0{,}2 + 0{,}2) * 2 + (0{,}2 + 0{,}1) * 3 =$ 2,3 бита.

Алгоритм Хаффмана (D = 2, B = {0,1})

1-й этап — построение двоичного дерева. Будем строить двоичное дерево с mлистьями, начиная с листьев и продвигаясь к

корню. Возьмём в качестве листьев дерева символы $a_1,\ldots, a_m$, которым приписаны вероятности $p_1,\ldots, p_m$.

Основная операция алгоритма — *редукция* — состоит в следующем. Возьмём две вершины a_{m-1} и a_m с наименьшими вероятностями p_{m-1} и p_m и добавим в дерево новую вершину $a_{m-1} \cup a_m$, которой припишем вероятность $p_{m-1} + p_m$. Вершину $a_{m-1} \cup a_m$ соединим рёбрами с вершинами a_{m-1} *и* a_m и объявим общим предком для этих вершин. Ребро от $a_{m-1} \cup a_m$ к a_{m-1}пометим символом 0, а ребро от $a_{m-1} \cup a_m$ к a_m — символом 1. Получаем новый (редуцированный) алфавит $A^{(1)} = \{a_1, \ldots, a_{m-2}, a_{m-1} \cup a_m\}$ с набором вероятностей $\vec{p}^{(1)} = (p_1, \ldots, p_{m-2}, p_{m-1} + p_m)$.

Если в полученном алфавите $A^{(1)}$ не менее двух символов, то упорядочим эти символы в порядке невозрастания вероятностей и снова применим операцию редукции.

Если же алфавит $A^{(1)}$ состоит из одного символа с приписанной ему вероятностью 1, то этот один символ объявим корнем и завершим этап построения дерева.

2-й этап — кодирование. Чтобы получить кодовое слово $\varphi(a_j)$ для символа a_j, последовательно считываем метки рёбер на пути от корня дерева к листу a_j.

Замечание. В описанном алгоритме имеется неоднозначность в операции редукции — как выбирать две вершины с наименьшими вероятностями, если имеется больше двух вершин с одинаковыми наименьшими вероятностями? Примем соглашение о том, что в случае такой неоднозначности выбираются две вершины с наибольшими номерами в списке вершин. Можно было бы принять и другое соглашение о правиле выбора, и тогда, вообще говоря, получалось бы другое дерево и другое кодирование. Можно доказать, что при любом таком соглашении о выборе пары вершин для редукции будет получаться оптимальное кодирование, т. е. с точки зрения средней длины кода несущественно, какое именно соглашение будет принято.

Замечание. Очевидно, что алгоритм Хаффмана приводит к префиксному кодированию, так как фактически происходит построение размеченного кодового дерева.

Пример.Применим алгоритм Хаффмана к данным, рассмотренным ранее. Шаги алгоритма представлены на следующем рис. 3 .

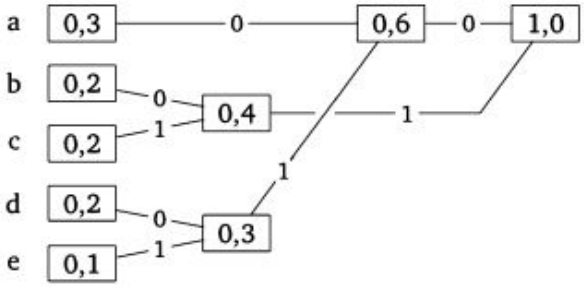


Рис. 3

В результате получается кодирование $\varphi(a)$ = 00, $\varphi(b)$ = 10, $\varphi(c)$ = 11, $\varphi(d)$ = 010, $\varphi(e)$ = 011 со средней длиной $l^{\varphi} = (0{,}3 + 0{,}2 + 0{,}2) * 2 + (0{,}2 + 0{,}1) * 3 = 2{,}3$ бита.

Арифметическое кодирование

Пусть дан алфавит $a_1, \ldots, a_m$, причем символы появляются с вероятностями $p_1, \ldots, p_m$. Кодируется последовательность $a_{i_1}, \ldots, a_{i_n}$ длины n. Алгоритм арифметического кодирования заключается в следующем.

Вход: объем алфавита m, вероятности букв p_i, $i = 1, \ldots, m$, длина последовательности n, последовательность на выходе источника $(x_1, \ldots, x_n)$.
Вывод: кодовое слово арифметического кода.
Кумулятивные вероятности:
$q_1 = 0$;
fori = 2 tomdo
$q_i = q_{i-1} + p_{i-1}$;
end
Кодирование:
F=0, G=1;
fori = 1 to n do
$F \leftarrow F + q(x_i)G$;
$G \leftarrow p(x_i)G$;
end
Формирование кодового слова:
$c \leftarrow$ первые $-logG+ 1$ разрядов после запятой в двоичной записи числа $F + G/2$.

Пример. Рассмотрим источник $X = \{a, b, c\}$, распределение вероятностей $p_a = 0{,}1$, $p_b = 0{,}6$, $p_c = 0{,}3$. Вычисления, выполняемые арифметическим кодером при кодировании последовательности $x = (bcbab)$ длиной $n = 5$, приведены в таблице. В этой таблице через $\hat{F}$ обозначено число $(F + G/2)$, округленное вниз с точностью до $-logG+ 1 = 9$ двоичных разрядов.

Шаг i	x_i	$p(x_i)$	$q(x_i)$	F	G
0	-	-	-	0,0000	1,0000
1	B	0,6	0,1	0,1000	0,6000
2	C	0,3	0,7	0,5200	0,1800
3	B	0,6	0,1	0,5380	0,1080
4	A	0,1	0,0	0,5380	0,0108
5	B	0,6	0,1	0,5391	0,0065
6	Длина кодового слова $-logG+ 1 = 9$. Кодовое слово $F + G/2 = 0{,}5423\ldots \rightarrow \hat{F} = 0{,}541 \rightarrow$ 100010101.				

Процесс декодирования арифметического кода заключается в следующем.

Вход: объем алфавита m, вероятности букв p_i, $i = 1, \ldots, m$, длина последовательности n, кодовое слово в виде числа $\hat{F}$.
Выход: декодированная последовательность букв $(x_1, \ldots, x_n)$.
Инициализация: $q_{n+1} = 1$; $S = 0$; $G = 1$.
Декодирование:

```
fori = 1 to n do
        j = 1;
        while S + q_i G < F̂ do
                j ← j + 1;
end
        S ← S + q_i G;
        G ← p_j G;
x_i = j;
end
```

Результат: последовательность $(x_1, \ldots, x_n)$.

Итак, декодеру арифметического кода известны алфавит $X = \{1, \ldots, m\}$, вероятности $\{p_1, \ldots, p_m\}$, кумулятивные вероятности $\{q_1, \ldots, q_m\}$, длина n последовательности сообщений и полученное из канала значение $\hat{F}$. Задача состоит в вычислении последовательности сообщений $\{x_i\}$. Приведенный выше алгоритм решает эту задачу. ***Пример.*** Рассмотрим источник $X = \{a, b, c\}$ с

распределением вероятностей $p_a = 0,1$, $p_b = 0,6$, $p_c = 0,3$. Пусть на вход декодера поступила двоичная последовательность 010001001. По этой последовательности нужно восстановить последовательность закодированных сообщений. Процесс декодирования отражен в таблице.

Шаг i	*S*	*G*	*Гипотеза x_i*	*q(x)*	*S + qG*	*Решение*	p(x)
0	*100010101 → $\hat{F}$ = 0,541*						
1	0,0000	1,0000	*A*	0,0	*0,0000 <$\hat{F}$*	*B*	0,6
			B	0,1	***0,1000<$\hat{F}$***		
			C	0,7	*0,7000 <$\hat{F}$*		
2	0,1000	6,0000	*A*	0,0	*0,1000 <$\hat{F}$*	*C*	0,3
			B	0,1	*0,1600 <$\hat{F}$*		
			C	0,7	***0,5200 <$\hat{F}$***		
3	0,5200	0,1800	*A*	0,0	*0,5200 <$\hat{F}$*	*B*	0,6
			B	0,1	***0,5380 <$\hat{F}$***		
			C	0,7	*0,6460 <$\hat{F}$*		
4	0,5380	0,1080	*A*	0,0	***0,5380 <$\hat{F}$***	*A*	0,1
			B	0,1	*0,5488 <$\hat{F}$*		
			C	-			
5	0,5380	0,0108	*A*	0,0	*0,5380 <$\hat{F}$*	*B*	0,6
			B	0,1	***0,5391 <$\hat{F}$***		
			C	0,7	*0,5456 <$\hat{F}$*		

Примеры решения задач

Пример 5.1. Сообщения составляются из алфавита *a, b, c, d*. Вероятность появления букв алфавита в текстах равна соответственно: *p(a)=0,1; p(b)=0,2; p(c)=0,6; p(d)=0,1*. Найти избыточность сообщений, составленных из данного алфавита.

Решение.

Для алфавита из четырех букв максимальная энтропия составит $H_{max} = log_2 m = log_2 4 = 2$ бит/символ. Средняя энтропия на символ сообщения

$$H = \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} = -\sum_{i=1}^{m} p_i \log_2 p_i =$$

$$-(0.1\log_2(0.1)+0.2\log_2(0.2)+0.6\log_2(0.6)+0.1\log_2(0.1)) = 1.23$$

бит/символ.

Тогда избыточность $R = 1 - H/H_{max} = 1-1,23/2 = 0,385$.

Пример 5.2. Построить оптимальный код сообщения, в котором вероятности появления букв первичного алфавита, состоящего из 8 символов, являются целой отрицательной степенью двух, а $\sum_i p_i = 1$.

Решение.

Построение оптимального кода ведется по методике Шеннона-Фано. Результаты построения отражены в таблице:

Буква	Вероятность появления буквы	Кодовое слово	Число знаков в кодовом слове	$p_i l_i$
A_1	1/4	00	2	0,5
A_2	1/4	01	2	0,5
A_3	1/8	100	3	0,375
A_4	1/8	101	3	0,375
A_5	1/16	1100	4	0,25
A_6	1/16	1101	4	0,25
A_7	1/16	1110	4	0,25
A_8	1/16	1111	4	0,25

Проверка:

$$L = \sum_{i=1}^{N} p_i l_i = 2*0{,}5 + 2*0{,}375 + 4*0{,}25 = 2{,}75;$$

$$H = -\sum p_i \log_2 p_i = 2*0{,}25\log_2 0{,}25 + 2*0{,}125\log_2 0{,}125 + 4*0{,}0625\log_2 0{,}0$$

$$= 1 + 2*0{,}75 + 4*0{,}25 = 2{,}75 \quad \textit{бит / символ.}$$

Примечание.*Кодовое слова одинаковой вероятности появления имеют равную длину.*

Пример 5.3. Чему равна минимальная длина кодовых слов для передачи 16, 128, 57, 10, 432 сообщений в двоичном и восьмеричном коде при равномерном кодировании.

Решение.

$L = log_2 N/log_2 m$

N – общее число передаваемых сообщений

m – качественный признак алфавита; *m=2;*

L1 = log_2 16/$log_2$2 = 4;
L2 = log_2 128/log_2 2 = 7;
L3 = log_2 57/log_2 2 = 5,83, округляем до 6;
L4 = log_2 10/log_2 2 = 3,32, L4 = 4;
L5 = log_2 432/log2 2 = log2 6+ log2 72 = 2,58 + 6,61 = 8,74, L5 = 9.
2) m=8; L1 = log_2 16/log_2 8 ≈ 1.3, L1 = 2;
L2 = log_2 128/log_2 8 = 2,33, L2 = 3;
L3 = log_2 57/log_2 8 = 1,94, округляем до 2;
L4 = log_2 10/log_2 8 = 1,1, L4 = 2;
L5 = log_2 432/log_2 8 = 8,74/3, L5 = 3.

Пример 5.4. Какое минимальное число вопросов необходимо задать собеседнику, чтобы угадать любое число из 240, если собеседник отвечает только "Да" и "Нет"?

Решение.

*N = log_2 240/log_2 2 = log_2 24+ log_2 10 = 4,58 + 3,32 = 7,9.*Так как число 7,9 вопросов нельзя задать, то число вопросов будет 8.

Пример 5.5. Дан источник А, вырабатывающий символы (*a, b, c, d, e*)с вероятностями *p(a)* = 0,3, *p(b)* = 0,2, *p(c)* = 0,2, *p(d)* = 0,2, *p(e)* = 0,1. Построить оптимальный код по методу Шеннона-Фано. Выполнить блочное кодирование по 2 символа по алгоритму Шеннона-Фано. Какой метод эффективнее?

Решение.

Построим оптимальный код по алгоритму Шеннона-Фано:

a	0,3	0	0	
b	0,2	0	1	
c	0,2	1	0	
d	0,2	1	1	0
e	0,1	1	1	1

Результат:

a->00*; b*->0*; c*->10*; d*->110*; e*->111.

Осуществим поблочное кодирование:

aa	0,36	0	0			
ab	0,18	0	1			
ba	0,18	1	0	0		
bb	0,09	1	0	1		
ac	0,06	1	1	0	0	
ca	0,06	1	1	0	1	
bc	0,03	1	1	1	0	
cb	0,03	1	1	1	1	0
cc	0,01	1	1	1	1	1

Найдём среднюю длину кодового слова:

$$\boldsymbol{l_{\varphi}}=\sum_{j=1}^{n}\boldsymbol{p_j l_j}$$

Средняя длина кодового слова для блочного кодирования по 1 символу:

$$\boldsymbol{l_{\varphi}}=(2*0,3+2*0,2+2*0,2+3*0,2+3*0,1)=2,3$$

Средняя длина кодового слова для блочного кодирования по 2 символа:

$\boldsymbol{l_{\varphi}}$=(2*0,36+2*0,18+3*0,18+3*0,09+4*0,06+4*0,06+4*0,03+5*0,03+5*0,01) =2,69

$$\frac{\boldsymbol{l_{\varphi}}}{\mathbf{2}} = 2,69/2 = 1,345$$

При кодировании по 1 символу средняя длина кодового слова равна 2,3; при кодировании по 2 символа средняя длина одного символа кодового слова равна 1,345. Следовательно, кодирование по 2 символам является более эффективным.

Задачи

Задача 5.1. Сообщения составляются из алфавита *a, b, c, d*. Вероятность появления букв алфавита в текстах равна соответственно: *p(a)*=0,2; *p(b)*=0,3; *p(c)*=0,4; *p(d)*=0,1. Найти избыточность сообщений, составленных из данного алфавита.

Задача 5.2. Вероятности появления символов на выходе источника сообщений следующие: $p_1=p_2=p_3=0{,}25$; $p_4=p_5=0{,}1$; $p_6=0{,}05$. Какой вид будет иметь оптимальный код для передачи?

Задача 5.3. Может ли средняя длина кодового слова оптимального кода быть меньше энтропии кодируемого алфавита?

Задача 5.4. Построить ОНК для передачи сообщений, алфавит которых состоит из двух букв А и В с вероятностями $p(A)=0{,}89$ и $p(B)=0{,}11$, при кодировании по одному, по два и по три символа в блоке. Оценить эффективность полученных кодов.

Задача 5.5. Закодировать оптимальным кодом последовательность из трёх символов A, B, C с вероятностями соответственно $p(A)=0{,}7$; $p(B)=0{,}2$; $p(C)=0{,}1$. Сравнить относительную эффективность при посимвольном кодировании, при кодировании по два и по три символа.

Задача 5.6. Построить методом Хаффмана оптимальный код для алфавита со следующим распределением вероятностей появления букв в тексте: $A=0{,}5$; $B=0{,}15$; $C=0{,}12$; $D=0{,}1$; $E=0{,}04$; $F=0{,}04$; $G=0{,}03$; $H=0{,}02$.

Задача 5.7. Построить ОНК по методу Шеннона-Фано и по методу Хаффмана, если символы источника сообщений появляются с вероятностями: $A_1= A_2= A_3= A_4=0{,}19$; $A_5=A_6=A_7=0{,}08$. Оценить эффективность полученных кодов.

Задача 5.8. Дан алфавит со следующим распределением вероятностей: $p_1=0{,}4$; $p_2=0{,}18$; $p_3=0{,}1$; $p_4=0{,}1$; $p_5=0{,}07$; $p_6=0{,}06$; $p_7=0{,}05$; $p_8=0{,}04$. Построить коды методом Шеннона-Фано и методом Хаффмана. Определить, какой код требует меньшую среднюю длину.

Задача 5.9. Какой вид имеют комбинации оптимального неравномерного кода при поблочном кодировании сообщений, составленных из алфавита *A, B, C, D*, если вероятности появления букв алфавита $p(A)=p(B)=p(C)=p(D)=0{,}25$?

Задача 5.10. Выяснить, являются ли однозначно декодируемыми следующие коды:

а) 010, 0100, 0010; б) 0, 01, 011, 111;

в) 0, 01, 001, 0010, 0011;

г) 110, 11, 100, 00, 10;

д) 100, 001, 101, 1101, 11011;

е) 010, 0001, 0110, 1100, 00011, 00110, 11110, 10101;

ж) 00, 012, 0110, 0112, 100, 201, 212, 22;

з) 123, 1234, 5, 421, 2135, 3513, 3512, 5124.

Задача 5.11. Построить двоичный код из четырёх кодовых слов, не префиксный, не суффиксный, но однозначно декодируемый.

Задача 5.12. Построить двоичный префиксный код с заданной последовательностью длин кодовых слов:

а) 1, 2, 3, 3; б) 1, 2, 4, 4, 4; в) 2, 2, 3, 4, 4;
г) 2, 2, 2, 4, 4, 4; д) 2, 3, 3, 3, 4, 4;
е) 1, 2, 3, 3, 4, 4, 4, 4; ж) 2, 2, 3, 3,4, 4, 4, 4.

Задача 5.13. Построить двоичные коды по алгоритмам Фано и Хаффмана для указанных ниже распределений: сравнить средние длины кодовых слов:

а) (0,4, 0,2, 0,2, 0,2); б) (0,7, 0,1, 0,1, 0,1);
в) $\left(\frac{27}{40},\frac{9}{40},\frac{3}{40},\frac{1}{40}\right)$; г) $\left(\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6},\frac{1}{20}\right)$;
д) $\left(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{16}\right)$;
е) (0,2,0,2, 0,2, 0,2,0,2);
ж) $\left(\frac{28}{72},\frac{15}{72},\frac{12}{72},\frac{11}{72},\frac{6}{72}\right)$; *з) (0,5,0,2, 0,1,0,09,0,08, 0,03);*
и) (0,5;0,2; 0,1; 0,09;0,08; 0,03);
к) (0,4;0,2;0,1;0,1;0,1;0,1);
л) (0,4; 0,3; 0,1; 0,07; 0,06; 0,04; 0,03);
м) (0,4; 0,3; 0,08; 0,06; 0,04; 0,04; 0,04; 0,04);
н) (0,32; 0,24; 0,20; 0,09; 0,05; 0,04; 0,04;0,02).

Задача 5.14. Построить коды по алгоритму Хаффмана для указанных ниже распределений при $D = 2$, вычислить средние длины кодовых слов:

а) $\left(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{16}\right)$; *б) (0,3, 0,2, 0,2, 0,2, 0,1);*
в) $\left(\frac{3}{8},\frac{1}{6},\frac{1}{8},\frac{1}{8},\frac{1}{8},\frac{1}{12}\right)$; *г)* $\left(\frac{1}{21},\frac{2}{21},\frac{3}{21},\frac{4}{21},\frac{5}{21},\frac{6}{21}\right)$;
д) (0,2; 0,15; 0,05; 0,2; 0,25; 0,15);
е) (0,4; 0,2; 0,1; 0,1; 0,1; 0,1);
ж) (0,3; 0,3; 0,1; 0,1; 0,1; 0,1);
з) (0,3; 0,2; 0,15; 0,15; 0,1; 0,1);
и) (0,24; 0,24; 0,16; 0,16; 0,12; 0,8);
к) (0,3; 0,2; 0,1; 0,1; 0,1; 0,1; 0,1);
л) (0,2; 0,12; 0,08; 0,15; 0,25; 0,1; 0,1);
м) (0,3; 0,25; 0,15; 0,1; 0,1; 0,05; 0,05);
н) (0,49; 0,26; 0,12; 0,04; 0,04; 0,03; 0,02);
о) (0,25;0,05; 0,1; 0,13; 0,2; 0,12; 0,08; 0,07);
п) (0,21; 0,2; 0,17; 0,16; 0,12; 0,08; 0,04; 0,02);
р) (0,2; 0,15; 0,15; 0,13; 0,12; 0,11; 0,11; 0,03);
с) (0,3; 0,2; 0,15; 0,1; 0,1; 0,08; 0,05; 0,02).

Задача 5.15. Для распределения $\left(\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{12}\right)$ построить два различных оптимальных двоичных кода с наборами длин кодовых слов (1, 2, 3, 3) и (2, 2, 2, 2).

Задача 5.16. Для распределения (0,3; 0,2;0,2; 0,1;0,1; 0,05; 0,05) построить три оптимальных двоичных кода с наборами длин кодовых слов (2, 2, 2, 3, 4, 5, 5), (2, 2, 3, 3, 3,4, 4) и (2, 2, 2,4, 4, 4, 4).

Задача 5.17. Построить двоичный код Хаффмана для распределения $\left(\frac{1}{3},\frac{1}{5},\frac{1}{5},\frac{2}{15},\frac{2}{15}\right)$. Доказать, что этот код является оптимальным и для распределения $\left(\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5}\right)$.

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ №8 - 12

Цель практических занятий: познакомиться с помехоустойчивыми кодами, изучить коды Хемминга. Научиться строить порождающую и проверочную матрицы для кода Хемминга. Научиться строить код Хемминга по проверочной матрице. Научиться вычислять синдром. Изучить линейные групповые коды и способы их построения. Научиться исправлять ошибки в процессе декодирования линейных кодов.

Помехоустойчивое кодирование

Пусть $A = \{a_1; a_2; \ldots; a_m\}$ – конечное множество символов. Тогда A^n есть множество всех слов длины n, состоящих из символов алфавитаA, т. е. $A^n = \{(a_{i_1}; a_{i_2}; \ldots; a_{i_n}),$ где $\forall a_{i_k} \epsilon A\}$.

Определение. Пусть$B = \{b_1, b_2, \ldots, b_D\}$ – входной алфавит канала связи; $X = \{x_1, x_2, \ldots, x_q\}$ – выходной алфавит канала связи. Кодер канала преобразовывает каждое информационное слово $\{b_{i_1}; b_{i_2}; \ldots; b_{i_k}\}$ длины k в кодовое слово $\{x_{i_1}; x_{i_2}; \ldots; x_{i_n}\}$ длины n. Пусть натуральные числа n и k таковы, что $D^k \leq q^n$.

Блоковым кодированием с входным алфавитом $B = \{b_1, \ldots, b_D\}$, выходным алфавитом $X = \{x_1 \ldots, x_q\}$, блоком информационных символов длины k и блоком кодовых символов длины n называется произвольное инъективное отображение $f : : B^k \to X^n$.

Множество $G = f(B^n) \subseteq X^n$, называется блоковым кодом, а его элементы — кодовыми словами.

Замечание. Предполагается, что введённое выше блоковое кодирование распространяется на множество B^* всех

конечных последовательностей в алфавите B следующим образом. Если входная последовательность имеет длину, кратную k, то она разбивается на непересекающиеся блоки длины k и каждый такой блок заменяется на блок длины n в алфавите X в соответствии с кодированием (независимо от других блоков). Если же длина входной последовательности не кратна k, то последний неполный блок дополняется до длины k некоторым заранее оговорённым способом, и
возвращаемся к предыдущему случаю.

Кодовые слова кода $G \subseteq X_n$ предназначены для передачи по дискретному каналу связи без памяти (далее коротко — каналу связи), при этом n символов кодового слова передаются по каналу связи последовательно, один за другим. Если на вход блокового кодера поступает информационный блок $\overrightarrow{b_k} = b_{j_1} b_{j_2} \ldots b_{j_k} \in B^k$, то полученное кодовое слово $\overrightarrow{x_n} = f(\overrightarrow{b_k}) \in G$ поступает в канал связи, и на выходе канала появляется некоторый вектор $\overrightarrow{y_n} \in Y^n$.

Далее вектор $\overrightarrow{y_n}$ поступает на вход некоторого декодера общего вида Q. Если декодер Q принимает решение, что на входе канала связи было кодовое слово $\tilde{x}^n \in G$, то к этому кодовому слову применяется обратное преобразование f^1, и в итоге получается информационный блок $\tilde{b}_k = f^1(\tilde{x}_n)$, который может и не совпадать с исходным блоком $\overrightarrow{b_k}$. Если же декодер Q выдаёт сообщение об ошибке типа «декодер не может принять решение», то делаем вывод о безуспешной передаче; в реальных каналах связи в этом случае может последовать запрос на повторную передачу информационного блока.

Для принятия решения часто используется алгоритм декодирования, основанный на расстоянии Хэмминга.

Определение. Пусть даны два кодовых слова $\vec{a}$ и $\vec{b}$, а $x(a_i, b_i) = \begin{cases} 1, a_i \neq b_i \\ 0, a_i = b_i \end{cases}$, *при i=1,2…,n.* Расстоянием Хэмминга на множестве X_n называется число $\rho(\vec{a}, \vec{b}) = \sum_{i=1}^{n} x(a_i, b_i)$ – число координат, в которых символы a_i и b_i, не совпадают.

Принцип построения декодера – принцип минимального расстояния, т. е. принятое слово декодируется в ближайшее кодовое слово.

Пример. Кодирование происходит по правилу

$i_1 = 00 \rightarrow 001 = c_1$

$i_2 = 01 \rightarrow 101 = c_2$

$i_3 = 10 \to 110 = c_3$

$i_4 = 11 \to 010 = c_4$

Такчто k=2, n=3.

Для этого кода $\rho(c_1, c_2) = 1$, $\rho(c_1, c_3) = 3$, $\rho(c_1, c_4) = 2$, $\rho(c_2, c_3) = 2$, $\rho(c_2, c_4) = 3$, $\rho(c_3, c_4) = 1$.

Пусть принято слово $y = 100$; для него кодовое $\rho(c_1, y) = 2$, $\rho(c_2, y) = 1$, $\rho(c_3, y) = 1$, $\rho(c_4, y) = 2$.

Следовательно, yможно декодировать и в c_2, и в c_3. Если декодирование неоднозначно, выбор осуществляется по заранее оговоренному правилу: например, выбирается наименьшее в лексикографическом смысле ($0 < 1$). Тогда $y \to c_2 = 101$.

Определение. Минимальным кодовым расстоянием блокового кода G называется число $d(G) = min\ \rho(\vec{a}, \vec{b})$, *для всех* $\vec{a}$, $\vec{b} \in G$.

В рассмотренном выше примере минимальное кодовое расстояние равно 1.

Теорема. Декодер Q обнаруживает любые комбинации из t или меньшего числа ошибок тогда и только тогда, когда $d(G) \geq t+1$.

Теорема. Декодер Q правильно исправляет любые комбинации из t или меньшего числа ошибок тогда и только тогда, когда $d(G) \geq 2t+1$.

В нашем примере $d = 1$, поэтому $t = 0$.

В дальнейшем считаем, что $B = \{0, 1\}$, $X = \{0, 1\}$.

Примеры решения задач

Пример 6.1.Чему равно общее количествоN комбинаций пятизначного двоичного кода с постоянным весом 2?

Решение.

Для двоичных кодов число кодовых комбинаций в кодах с постоянным весомl длиной в nсимволов равно

$$N = C_n^l = \frac{n!}{l!(n-l)!},$$

где l - число единиц в кодовом слове.

$$N = \frac{5!}{2!(5-2)!} = \frac{4*5}{2} = 10.$$

Пример 6.2.Определить минимальное кодовое расстояние, необходимое при построении кода, исправляющего двойную ошибку.

Решение.

Согласно теореме, код с минимальным расстоянием d исправляет tошибок, если $d \geq 2t+1$.

Таким образом, для того, чтобы код исправлял 2 ошибки, минимальное расстояние d = 2*2+1 = 5.

Пример 6.3. Даны кодовые слова:

c_1 = 00000;c_2 = 11100;c_3 = 10011;c_4 = 01111.

Найти минимальное кодовое расстояние по Хэммингу.

Решение.

$d(1,2)$ = 3;$d(1,3)$ = 3;$d(1,4)$ = 4;$d(2,3)$ = 4;$d(2,4)$ = 3;$d(3,4)$ = 3;d_{min} = 3.

Задачи

Задача 6.1.Чему равно общее количество комбинаций пятизначного двоичного кода с постоянным весом $l=2$? Построить все комбинации такого кода.

Задача 6.2.Определить минимальное кодовое расстояние, необходимое для обнаружения в коде тройной ошибки.

Задача 6.3.Какое количество ошибок может исправить код, в котором минимальное кодовое расстояние $d(G)=7$.

Задача 6.4.Даны кодовые слова: 00001, 11100, 10110, 01110, Можно ли в них обнаружить одиночную ошибку?

Задача 6.5.Определить минимальное кодовое расстояние, необходимое при построении кода, исправляющего двойную ошибку.

Задача 6.6.Построить четырёхзначный двоичный код, обнаруживающий одиночную ошибку.

Задача 6.7.Определить минимальное кодовое расстояние для кодов: а) обнаруживающего 3 и исправляющего 2 ошибки; б) обнаруживающего 5 и исправляющего 3 ошибки.

Задача 6.8.Какое максимальное кодовое расстояние может быть между двумя пятизначными комбинациями?

Задача 6.9.Чему равно кодовое расстояние между комбинациями 0001 и0010, 11000111001 и 10000011101?

Линейные коды

Линейные блоковые коды являются важным классом кодов, для которых описание, оценка параметров, алгоритмы кодирования и декодирования основаны на линейной алгебре.

Информационные слова, $C \in B^k$ образуют линейное пространство размерности *k*, то есть линейная комбинация информационных слов есть информационное слово.

Определение. Линейным блоковым кодом длины n называется произвольное линейное подпространство C размерности k линейного пространства $V^n(2)$.Код *C* есть линейный(*n; k)*- код.

Для линейного кода сумма двух кодовых слов есть кодовое слово; нулевое слово длины n есть кодовое слово.

Определение.Вес Хэмминга *w(r)* кодового слова $\vec{c}$равен числу его ненулевых компонентов.

Для линейного кода минимальное расстояние $d^* = \boldsymbol{min}_{\forall \vec{c} \neq \mathbf{0}} \boldsymbol{w}(\vec{c}) = d(C)$.

Матричное описание линейных блоковых кодов

Линейный код*C* образует подпространство в $GF(2^n)$.Любое множество базисных векторов этого подпространства может быть использовано в качестве строк для построения*kxn* матрицы*G*, которая является порождающей матрицей кода. Пространство строк матрицы *G* порождает линейный код *C*.Любое кодовое слово есть линейная комбинация строк из *G*.

Наиболее естественный способ кодирования использует отображение $\vec{c} = \vec{\boldsymbol{\imath}} * G$, где $\vec{\boldsymbol{\imath}}$ - информационное слово (строка) *k* последовательно кодируемых информационных символов, $\vec{c}$ - кодовое слово. Это равенство определят кодер и зависит от выбора базисных векторов.

Рассмотрим порождающую матрицу $G_{3x5} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$

Пусть $\vec{\boldsymbol{\imath}} = (0; 1; 1)$, тогда кодовое слово

$\vec{c} = (0; 1; 1) * \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} = (0; 1; 1; 1; 0).$

Порождающая матрица является сжатым описанием линейного кода.

Так как *C* является подпространством, то оно имеет ортогональное дополнение $\boldsymbol{C}$, которое состоит из всех векторов, ортогональных к *C*. Ортогональное дополнение также является подпространством и тоже может рассматриваться как код. Этот код называется дуальным (двойственным) к *C*. Ортогональное дополнение C имеет размерность*(n-k)xn*и любой его базис состоит из *n-k* векторов. Пусть *H* – матрица, строки которой являются этими базисными векторами, тогда последовательность $\vec{c}$ является кодовым словом тогда и только тогда, когда она ортогональна каждой строке матрицы *H*, т. е. если $\vec{c} * H^{tr} = 0$.

Это равенство позволяет проверить, является ли данное слово кодовым. Матрица *H* называется проверочной матрицей кода. Она является матрицей размерности *(n-k) xn* и поскольку равенство $\vec{c} * H^{tr} = 0$ выполняется для любой строки матрицы *G*, то $\boldsymbol{G} * H = \emptyset$.

Пусть для *G* в примере $H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$, тогда $\boldsymbol{G} * H^{tr}$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Порождающая и проверочная матрицы не единственны.

Систематические коды представляют собой такие коды, в которых информационные и корректирующие разряды расположены по строго определенной системе и всегда занимают строго определенные места в кодовых комбинациях. Систематические коды являются равномерными, т. е. все комбинации кода с заданными корректирующими способностями имеют одинаковую длину. Порождающая матрица имеет левую каноническую форму, если она имеет вид $G_{kxn} = (Ek \mid \boldsymbol{G}^{(1)}_{\boldsymbol{k\,x\,(n-k)}})$.Тогда $H = (-\boldsymbol{G}^{(1)}_{\boldsymbol{T}} \mid \boldsymbol{E_{n-k}})$. Для правой канонической формы $G_{kxn} = (\boldsymbol{G}^{(2)}_{\boldsymbol{k\,x\,(n-k)}} \mid \boldsymbol{E_k})$, а $H = (En\text{-}k \mid -\boldsymbol{G}^{(2)\boldsymbol{tr}}_{\boldsymbol{T}})$.

Если порождающая матрица имеет каноническую форму, то код систематический.

Теорема 1.Если любые $l \le d - 1$ столбцов проверочной матрицы линейного (n, k)-кода линейно независимы, то $d(G) = d$.

Теорема 2.Если $d(G) = d$, то любые $l \le d - 1$ столбцов проверочной матрицы H линейно независимы и найдутся d линейно зависимых столбцов.

Пусть H – проверочная матрица линейного (n, k)-кода. Синдромом слова $\vec{y} \in X^n$, $X = GF(q)$, называется вектор $\vec{S}(\vec{y})$, определяемый соотношением $\vec{S}(\vec{y}) = \vec{y} * H^{tr}$. Очевидно, что $\vec{S}(\vec{c}) = \vec{0}$ для любого кодового слова $\vec{c} \in C$.Поэтому ненулевой синдром$\vec{S}(\vec{y})$, указывает на наличие ошибок в принятом слове $\vec{y}$. Предположим, что передавалось кодовое слово $\vec{c}$, а в декодер поступает искаженное слово $\vec{y} = \vec{e} + \vec{c}$, где $\vec{e}$– вектор ошибок.. Хотя вектор $\vec{y}$ зависит как от переданного слова, так и от ошибок, синдром $\vec{S}(\vec{y})$ зависит только от вектора ошибок:

$$\vec{S}(\vec{y}) = \vec{y} * H^{tr} = (\vec{c} + \vec{e}) * H^{tr} = \vec{c} * H^{tr} + \vec{e} * H^{tr} = \vec{e} * H^{tr} = \vec{S}(\vec{e}).$$

Пусть $\vec{y} = (y_1, \ldots, y_n)$ – принятая последовательность.

На **первом шаге** по последовательности $\vec{y}$, поступившей на вход декодера, вычисляется синдром $\vec{S}(\vec{y}) = \vec{y} * H^{tr} = (s_1, \ldots, s_r)$, *где* $r = n - k$.

На **втором шаге** по вычисленному синдрому $\vec{S}(\vec{y})$ отыскивается вектор ошибок e, вес которого не превосходит t_0, а синдром совпадает с$\vec{S}(\vec{y})$.

На **третьем шаге** для найденного вектора ошибки определяется переданное кодовое слово $\vec{c} = \vec{y} - \vec{e}$или выдается специальный сигнал в случае обнаружения ошибок.

Теперь покажем, что синдромное декодирование позволяет обнаруживать ошибки, которые не могут быть исправлены.

Примеры решения задач

Пример 7.1.Рассмотрим расширенный двоичный (8, 4)-код Хэмминга с расстоянием d=4 и проверочной матрицей H = $\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$.

Этот код позволяет исправлять однократные и обнаруживать двукратные ошибки. Векторы исправляемых ошибок и их синдромы представлены в таблице:

Векторы ошибок $\vec{e}$	Синдромы ($\tilde{o}_1$,$\tilde{o}_2$,$\tilde{o}_3$,$\tilde{o}_4$)
00000000	0000
10000000	1001
01000000	0101
00100000	1101
00010000	0011
00001000	1011
00000100	0111
00000010	1111
00000001	0001

Имеется 2^4 = 16 различных синдромов , но только 9 из них представлены в таблице. Приведенные синдромы соответствуют ошибкам кратности 1. Все ошибки кратности 2 и также некоторые другие ошибки имеют другие синдромы, не указанные в таблице. Например, если передавалось слово $\vec{c}$ = (11100001) и произошли две ошибки на позициях 1 и 5, то получено слово $\vec{y}$ = (01101001), то $\vec{S}(\vec{y})$ = (0010). Попытка найти вектор ошибки, имеющий такой же синдром, приведет к неудаче, поскольку такого синдрома в таблице нет. В этом случае декодер должен сигнализировать о том, что имеет место ошибка, которую невозможно исправить.

Пример 7.2. Задан двоичный код, состоящий из 8 слов:
000000; 110100; 101010; 011110; 011001; 101101; 110011; 000111
Определить, является ли данный код линейным.

Решение.

Определим, образуют ли данные векторы линейное подпространство. Найдём все попарные суммы ненулевых слов:

110100⊕101010 = 011110; 110100 ⊕011110 = 101010; 110100 ⊕011001= 101101; 110100 ⊕101101 = 011001; 110100 ⊕110011 = 000111; 110100 ⊕000111 = 110011; 101010 ⊕011110 = 110100; 101010 ⊕011001 = 110011; 101010 ⊕101101 = 000111; 101010 ⊕110011 = 011001; 101010 ⊕000111 = 101101; 011110 ⊕011001=000111; 011110 ⊕101101 = 110011; 011110 ⊕101101=110011; 011110 ⊕110011=101101; 011110 ⊕000111 = 011001; 011001 ⊕101101 = 110100;011001 ⊕110011=101010; 011001 ⊕000111 = 011110; 101101 ⊕110011 = 011110; 101101 ⊕000111 = 101010; 110011 ⊕000111 = 110100; 110100 ⊕110100 = 000000.

Это линейное подпространство. Данный код является линейным.

Пример 7.3. Дана порождающая матрица G_{3x6} = $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$

Множество информационных слов имеет размерность k = 3 и содержит 8 слов:

$\overrightarrow{i_1}$ = 000;$\overrightarrow{i_2}$ = 001;$\overrightarrow{i_3}$ = 010;$\overrightarrow{i_4}$= 011;$\overrightarrow{i_5}$= 100;$\overrightarrow{i_6}$ = 101;$\overrightarrow{i_7}$ = 110;$\overrightarrow{i_8}$ = 111;

Составить таблицу кодирования.

Решение.

1. (000) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (000000);

2. (001) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (011001);

3. (010) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (101010);

4. (011) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (110011);

5. (100) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (110100);

6. (101) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (101101);

7. (110) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (011110);

8. (111) x $\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ = (000111);

Пример 7.4. Для приведенного выше примера найти проверочную матрицу H_{3x6}.

Решение.

Матрица $G_{3x6} = \begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$ имеет правую каноническую форму.

Найдем H_{3x6} как $\begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$

Проверим, выполняется ли условие $GxH^{tr} = \emptyset$.

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix} \text{x} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} = \emptyset.$$

Пример 7.5. Для предыдущего примера рассмотреть векторы ошибок:
e_1 = 100000; e_2 = 010000; e_3= 001000; e_4 = 000100; e_5= 000010; e_6 = 000001
и найти все синдромы s_i.
Решение.

$$s_1 = (100000) \text{ x} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} = (100);$$

$$s_2 = (010000) \text{ x} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} = (010);$$

s_3 = (001); s_4 = (110); s_5 = (101); s_6 = (011).

Пример 7.6. Используя данные, полученные в предыдущих примерах, найти кодовое слово для информационного слова $\vec{\imath}$ = 011 и осуществить передачу кодового слова по каналу связи. Предположить, что слово получено с ошибкой в 3 разряде. Исправить эту ошибку.

Решение.

Кодовое слово для слова 011 -> 110011. Осуществим его передачу по каналу связи.

Предположим, что полученное слово $\vec{y}$ = 111011. Найдем его синдром $\vec{S}(\vec{y}) = \vec{y} \times H^{tr}$.

$$(111011) \times \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} = (001).$$

Данный вектор соответствует третьей ошибке. Значит, ошибка в третьем разряде. Исправим ее: $\vec{c}$ = 110011.

Задачи

Задача 7.1.Сложить по модулю 2 слова: 10110110, 110 и 11101; 100110 и 100110,

Задача 7.2.Определить вес кодового вектора, полученного в результате сложения по модулю 2 кодовых векторов: 1010001 и 0110101.

Задача 7.3.Чему равен вес следующих двоичных кодовых слов: 11011010; 10110; 0000000; 10; 01; 11111111?

Задача 7.4.Определить минимальное кодовое расстояние между двоичными векторами: 1100011; 1001111; 1010101.

Задача 7.5.Определить корректирующие способности следующего кода: 001; 010; 111.

Задача 7.6. Способен ли код исправлять ошибки, если его комбинации имеют вид: 1001010; 0101110; 1101001; 0011011; 1011100?

Задача 7.7.Построить линейный код по заданной порождающей матрице:

G_{4x7}						
1	0	0	0	1	1	1
0	1	0	0	1	1	0
0	0	1	0	1	0	1
0	0	0	1	0	1	1

Задача 7. 8. Линейный код построен по матрице

G_{4x7}						
1	0	0	0	0	1	1
0	1	0	0	1	0	1
0	0	1	0	1	1	0
0	0	0	1	1	1	1

Показать процесс исправления ошибки в произвольном разряде корректирующего кода, информационная часть которого представляет собой четырёхразрядные комбинации натурального двоичного кода.

Задача 7.9. Определить, какие из приведённых ниже комбинаций групповых кодов содержат ошибку: 1100111; 0110101; 0011010; 0010110, если известно, что код построен по матрице

G_{4x7}						
1	0	0	0	1	1	1
0	1	0	0	0	1	1
0	0	1	0	1	1	0
0	0	0	1	1	0	1

Задача 7.10. Построить линейное пространство над полем GF(3), проходящее через векторы $\overrightarrow{x_1}$ = (1010), $\overrightarrow{x_2}$ = (0111). Поскольку имеется несколько таких линейных пространств, следует выбирать пространство наименьшей размерности. Содержит ли это пространство вектор (2101)? Определить, каким числом способов можно выбрать различные системы базисных векторов в этом пространстве.

Задача 7.11. Рассмотреть двоичный код C, состоящий из следующих восьми слов:

{000000; 001011; 010101; 011110; 100110; 101101; 110011; 111000}.

Является ли код C линейным? Если да, то найти для этого кода порождающую и проверочную матрицы. Каковы основные параметры этого кода? Являются ли линейными кодами подмножества слов 1-4 и 5-8? Если да, то найти для этих кодов порождающие и проверочные матрицы.

Простейшие систематические коды. Код Хэмминга

Примитивный код Хэмминга является типичным примером систематического кода.

Его параметры определяются по формулам: r- число проверочных разрядов; $n=2^r-1$- длина кодового слова; $k=n-r$- длина информационного слова.

В отличие от канонического систематического кода, в коде Хэмминга информационные и проверочные биты не разнесены в отдельные подпоследовательности, а чередуются. Если биты кодовой комбинации пронумеровать, начиная с 1, слева направо, то контрольными (проверочными) оказываются номера 1, 2, 4, 8 и т. д.; все остальные биты являются информационными.

Цель этих перестановок – сделать так, чтобы синдром ошибки непосредственно указывал на локализацию ошибки, минуя промежуточную таблицу соответствия синдромов и ошибок. Код Хэмминга начинают строить с проверочной матрицы *H*, так как ее вид очевиден: столбцы проверочной матрицы представляют собой набор синдромов, соответствующих двоичному представлению номера столбца. Порождающая матрица стоится так. Выделим в матрице Н подматрицу Н1, состоящую из столбцов с номерами, не равными степеням двойки. Повернём матрицу Н1 и получим проверочную матрицу Р порождающей матрицы. Поставим столбцы матрицы Р на 1,2,4 и т.д. места (номера столбцов равны степеням двойки), а остальные столбцы будут столбцами единичной матрицы.

Примеры решения задач

Пример 8.1. Известно, что число проверочных символов r = 3. Записать код Хэмминга.

Решение.

При *r = 3 параметры кода:* $n=2^r-1=2^3-1=7$; $k=n-r=4$. Конструируется проверочная матрица $H_{rxn}= H_{3x7}$. Среди столбцов проверочной матрицы отсутствует нулевой и нет равных столбцов. Очевидно, имеется 7 различных ненулевых трехбитовых символов. Запишем каждый столбец его номером в двоичной системе:

$$H_{3x7}=\begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Выделим столбцы, являющиеся степенями двойки. Запишем подматрицу из остальных столбцов. Тогда:

$$H1_{3x4} = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{vmatrix}$$

Повернем матрицуН1 на 90^0 по часовой стрелке. Поставим столбцы этой матрицы на первое, второе и четвертое места в порождающей матрице. Остальные столбцы будут столбцами единичной матрицы.

$$G_{4x7} = \begin{vmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

Если информационное слово$\vec{i} = \{i_1 i_2 i_3 i_4\}$

$\vec{c} = \{i_1 + i_2 + i_4;\ i_1 + i_3 + i_4;\ i_1;\ i_2 + i_3 + i_4;\ i_2;\ i_3;\ i_4\}$.

Как видим, на 3, 5, 6 и 7 местах стоят информационные символы. Отсюда следует процедура декодирования. Получив слово и убедившись, что оно кодовое, извлекаем из него 3, 5, 6 и 7 биты и получаем информационное слово.

Пример 8.2. Известно, что число проверочных символов r = 4. Записать код Хэмминга.

Решение:

При r = 4 получим n = 15, k = 15-4 = 11.

$$H_{4x15} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Выделим столбцы, являющиеся степенями двойки. Запишем подматрицу из остальных столбцов.

$$R_{4x11} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$ и повернем эту матрицу на 90^o по часовой стрелке.

$$R_{11x4} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Вычислим теперь порождающую матрицу *G11x15*.

$$G_{11x15} = \left(\begin{array}{c:ccc|ccccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Пример 3. Построить код Хемминга для передачи сообщений длиной 4 бита. Показать процесс кодирования, декодирования и исправления одиночной ошибки на примере информационного слова 0101.

Решение:

Код имеет параметры k=4, n=7, d=3. Кодирование происходит по формуле $\vec{c} = \vec{\imath} * G$, так что:

$$\vec{c} = (0\ 1\ 0\ 1) * \begin{pmatrix} 1\ 1\ 1\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 1\ 1\ 0\ 0 \\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \\ 1\ 1\ 0\ 1\ 0\ 0\ 1 \end{pmatrix} = (0\ 1\ 0\ 0\ 1\ 0\ 1)$$

Пусть принятое слово $\vec{y} = (0\ 1\ 0\ 0\ 1\ 0\ 1)$. Для проверки безошибочности приёма вычислим синдром:

$$\vec{s} = \vec{y} * H^{tr} = (0\ 1\ 0\ 0\ 1\ 0\ 1) * \begin{pmatrix} 0\ 0\ 1 \\ 0\ 1\ 0 \\ 0\ 1\ 1 \\ 1\ 0\ 0 \\ 1\ 0\ 1 \\ 1\ 1\ 0 \\ 1\ 1\ 1 \end{pmatrix} = (0\ 0\ 0)$$

Нулевой синдром означает, что $\vec{y}$ есть кодовое слово.
Для декодирования выбираем 3, 5, 6 и 7 биты; получаем информационное слово $\vec{c} = (0\ 1\ 0\ 1)$.
Пусть в 3 бите произошла ошибка и получено слово $\vec{y}$=(0 1 1 0 1 0 1). Вычислим синдром:

$$\vec{s} = \vec{y} * H^{tr} = (0\ 1\ 1\ 0\ 1\ 0\ 1) * \begin{pmatrix} 0\ 0\ 1 \\ 0\ 1\ 0 \\ 0\ 1\ 1 \\ 1\ 0\ 0 \\ 1\ 0\ 1 \\ 1\ 1\ 0 \\ 1\ 1\ 1 \end{pmatrix} = (0\ 1\ 1)$$

Это означает, что синдром $\vec{s} = (0\ 1\ 1)$ равен 3 в двоичной записи. Исправляем (инвертируем) 3-й бит и декодируем кодовое слово, как было сделано ранее.

Задачи

Задача 8.1.Чему равно количество ненулевых комбинаций систематического кода, содержащего 10 информационных разрядов, если этот код предназначен для передачи сообщений, в которых должна быть исправлена одиночная ошибка?

Задача 8.2.Какое вид имеют комбинации шестиразрядного систематического кода, если порождающая матрица имеет вид:

G_{3x6}					
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

Задача 8.3.Построить систематический код, исправляющий одиночную ошибку. Общее количество сообщений, передаваемых комбинациями полученного кода, должно быть не менее 30,

Задача 8.4.Построить код Хемминга для информационной комбинации 0101. Показать процесс обнаружения ошибки.

Задача 8.5.Какой вид имеют комбинации корректирующего кода Хемминга для передачи сообщений 1101 и 1011? Показать процесс построения корректирующего кода.

Задача 8.6.Переданы следующие комбинации в (7,4)-коде Хемминга: 1101001, 0001111, 0111100, Получены – 1001001, 0011111, 0110100, Показать процесс обнаружения ошибки.

Задача 8.7.Построить код Хемминга для передачи одиннадцатиразрядной информационной комбинации 10110110110, Показать процесс обнаружения ошибки, которая произошла в пятом разряде соответствующей комбинации корректирующего кода.

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ №13 - 18

Цель практических занятий: изучить циклические коды и коды БЧХ, научиться строить порождающий многочлен и проверочный многочлен циклического кода. Научиться находить кодовое слово и локализовывать ошибку. Научиться строить порождающий многочлен для кода БЧХ с заданной корректирующей способностью. Рассмотреть кодирование непрерывных сообщений на примере сверточных кодов. Изучить основные параметры сверточных кодов и способы их построения.

Циклические коды

Определение.Линейный двоичный *(n, k)* – код называется циклическим, если каждое кодовое слово $\vec{c} \in C$, будучи циклически сдвинуто (влево или вправо) на любое число позиций, также является кодовым словом.

Каждому кодовому слову $\vec{c} = \{c_0, c_1, \ldots, c_{n-1}\}$ однозначно сопоставляется полином $c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$ от формальной

переменной x. Рассмотрим кодовое слово $\vec{c}' = \{c_{n-1}; c_0; c_1; \ldots; c_{n-2}\}$ и полином $c'(x) = c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1}$. Очевидно, $c'(x) = x(c_0 + c_1x + \ldots + c_{n-2}x^{n-2} + c_{n-1}x^{n-1}) - c_{n-1}(x^n - 1)$, поэтому $c'(x) = xc(x)(mod(x^n - 1))$ или $c'(x) = x*c(x)(mod(x^n + 1))$. Следовательно, все полиномы вида $x^jc(x)(mod(x^n + 1))$ являются кодовыми.

Определение. Порождающим полиномом циклического (n, k) – кода называется кодовый полином $g(x)$наименьшей степени.

Степень $g(x)$равна n-$k = r$ и каждое кодовое слово $c(x)$ из С делится без остатка на $g(x)$.

Пусть С - циклический (n, k)-код. Выберем в Сненулевой кодовый многочлен наименьшей степени n-k, где $k \geq 1$. Фактически технология такая: рассматриваются все кодовые слова длины n. Обозначим кодовые слова многочленами и выберем среди них кодовый многочлен наименьшей степени r. Обозначим $k = n - r$–это длина информационного слова.

Порождающий полином $g(x)$ циклического (n, k) – кода G является делителем двучлена $x^n \pm 1$.

Циклический двоичный линейный код называется примитивным, если $n = 2^r - 1$. Тогда $k = n - r = 2^r - r - 1$. В этом случае многочлен$x^n \pm 1$ раскладывается в произведение всех неприводимых над полем Z_2 полиномов, степени которых делят r. В качестве порождающего полинома $g(x)$ можно взять любой делитель $x^n - 1$ при условии, чтоnесть наименьшее положительное целое число, для которого $x^n - 1 = 0(modg(x))$. Каждый такой полином g(x) будет порождать циклический двоичной код с длиной n, размерностью $k = n - r$, где r – степень $g(x)$, и со своим минимальным расстоянием d.

Так как циклический код линеен, то для него можно построить порождающую и проверочную матрицы. Пусть g(x) – порождающий многочлен. Тогда

$$G_{k\times n} = \begin{pmatrix} g_0\, g_1 \ \ldots\ g_{r-1} g_r\ 0 \ldots 0 \\ 0\ \ g_0\ \ldots g_{r-2} g_{r-1}\, g_r \ldots 0 \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ 0\ \ 0 \ldots 0\, g_0 g_1\ \ldots\ g_r \end{pmatrix} = \begin{pmatrix} g(x) \\ xg(x) \\ \ldots \\ x^{k-1}g(x) \end{pmatrix}$$

$H_{(n-k)\times n} = (1\ \ x\ \ x^2\ \ \ldots\ \ x^{n-2} x^{n-1})(mod\ g(x))$, где столбцы проверочной матрицы являются остатками от деления x^j на g(x), $j=0, \ldots, n-1$.

Пример. Рассмотрим двоичный код {000; 110; 011; 101}. Это циклический линейный код над полем Z_2 с d=2 (*проверьте!*). Порождающий многочлен кода $g(x)=1+x$ (почему?). Параметры кода n=3, k=2. Порождающая и проверочная матрицы кода имеют вид:

$$G_{2\times3} = \begin{pmatrix} 1+x \\ x+x^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix};\ H_{1\times3} = (1 \quad x \quad x^2)\big(mod(x+1)\big) = (1 \quad 1 \quad 1);$$

Пример.Пусть длина циклического кода $n = 7$. Разложить на множители многочлен $x^7 + 1$.

Решение:

Так как $7 = 2^3-1$, то r = 3. Неприводимые многочлены должны иметь степени 1 и 3. Неприводимый многочлен первой степени - это $x+1$. Неприводимые многочлены третьей степени - это $x^3 + x^2 +1$ и $x^3 + x + 1$.

Итак, $x^7 + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x +1)$.

Пусть многочлен $i(x)$ является информационным словом:

$i_0 + i_1x + ... + i_{k-1}x^{k-1} = \{i_0, i_1, \ldots, i_{k-1}\}$.

Все кодовые многочлены находятся по формуле

$$c(x) = i_{k-1}(x)*g_{n-k}(x).$$

Пример.Пусть $r = 3$, $n= 2^3 – 1 = 7$. Тогда $k = 4$, а $x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$. Выберем $g(x) = x^3 + x + 1$. Пусть$i(x) = i_0 + i_1x + i_2x^2 + i_3x^3$, тогда $c(x) = (i_0 + i_1x + i_2x^2 + i_3x^3)(1 + x + x^3) = i_0 + (i_0 + i_1)x + (i_1 + i_2)x^2 + (i_0 + i_2 + i_3)x^3 + (i_1 + i_3)x^4 + i_2x^5 + i_3x^6$. Пусть $c(x)$ принят безошибочно и$c(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6$. Тогда информационные символы восстанавливаются и системы уравнений:

$$\begin{cases} i_3 = c_6 \\ i_2 = c_5 \\ i_0 = c_0 \\ i_0 + i_1 = c_1 \end{cases}$$

Следовательно, кодирование несистематическое.

Пример.Рассмотрим другой способ кодирования: $c(x) = x^r * i(x) + R_{g(x)}(x^r * i(x))$. Применивего, получим, что $c(x) = x^3(i_0 + i_1x + i_2x^2 + i_3x^3) + R_{g(x)}(i_3x^6 + i_2x^5 + i_1x^4 + i_0x^3)$.Найдемпоследнееслагаемое$S(x)$иполучим, что$c(x) = (i_3x^6 + i_2x^5 + i_1x^4 + i_0x^3) + (s_2x^2 + s_1x + s_0) = s_0 + s_1x + s_2x^2 + i_0x^3 + i_1x^4 + i_2x^5 + i_3x^6$. Информационные символы занимают строго определенные позиции, т.е. кодирование систематическое.

Проверочным многочленом называется многочлен $h(x) = (x^n + 1)/g(x)$. Если принятое слово $y(x)$ удовлетворяет равенству $h(x) * y(x) = 0 \ (mod \ (x^n + 1))$, *то* $y(x) = c(x)$. В противном случае при передаче произошла ошибка.

Алгебраическое декодирование двоичного циклического кода с минимальным расстоянием 3.

Рассмотрим двоичный циклический код Хэмминга с минимальным расстоянием 3. Порождающий полином g(x) такого кода является неприводимым над $GF(2)$ примитивным делителем двучлена x^n-1, $n = 2^m-1$. Обозначим через □ корень этого полинома и воспользуемся тем, что каждое кодовое слово $c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$ делится нацело на g(x) и, следовательно, $c(□) = 0$.

Принятое слово можно представить в виде $y(x) = c(x) + e(x)$. В случае одиночной ошибки на позиции i полином ошибок равен $e(x) = x^i$. Синдром слова $y(x)$ равен:

$$S(y)=y(□) = c(□) +e(□)= = e(□) = □^i.$$

Для того, чтобы найти позицию ошибки по синдрому S, нужно решить уравнение:

$$□^i = S$$

Соответствующее значение i называется логарифмом S и записывается как

$$i = log_2 S.$$

Поэтому декодирование двоичного циклического кода состоит из следующих шагов: вычисление синдрома $S = y(□)$; определение позиции ошибки $i = log_□ S$; инвертирование символа в слове $y(x)$ на позиции i.

Алгебраическое декодирование двоичного циклического кода с минимальным расстоянием 5

Пусть $f_1(x)$ – примитивный полином, порождающий поле $GF(q)$, $q = 2^m$, и его корень □ - это примитивный элемент этого поля. Рассмотрим декодирование двоичного циклического кода С*с* длиной $n = 2^m - 1$ и минимальный расстоянием 5. Проверочная матрица для такого кода может быть взята в виде

$$H = \begin{pmatrix} 1 & 2 & \ldots & n-1 \\ 3 & 6 & 9 & \ldots & 3(n-1) \end{pmatrix}$$

Из условия $c*H^T = 0$ вытекает, что для любого кодового слова $c(x) \in C$ выполняются равенства:

$$c(\square) = 0,\ c(\square^3) = 0,$$

т. е. величины $\square$ и $\square^3$ являются корнями каждого кодового слова и, следовательно, корнями порождающего полинома.

Теперь опишем алгебраический метод определения позиций ошибок. Для этого введем в рассмотрение локаторы позиций, сопоставляя n позициям кодового слова различные ненулевые величины поля $GF(2^m)$,например сопоставляя $i \leftrightarrow \square^i$ для всех $i = 0, 1, \ldots, n-1$.

Пусть при передаче произошли две ошибки на позициях с номерами $X_1 = \square^i$, $X_2 = \square^j$. Локаторы однозначно определяют позиции ошибок, $i = log_{\square} X_1$ *и* $j = log_{\square} X^2$. Поэтому задачу декодирования двоичного кода можно представить как задачу отыскания локаторов ошибок.

В случае двойной ошибки слово на входе декодера имеет следующий вид:

$$y(x) = c(x) + x^i + x^j.$$

Декодеру при этом известно, что $c(\square) = 0,\ c(\square^3) = 0$. Введем в рассмотрение две величины $S_1 = y(\square) = \square^i + \square^j$, $S_3 = y(\square^3) = \square^{3i} + \square^{3j}$ и назовем их компонентами синдрома. Их можно вычислить по полиному $y(x)$, поэтому они считаются известными к началу декодирования. Нетрудно видеть, что локаторы ошибок и компоненты синдрома связаны между собой следующими соотношениями:

$$S_1 = X_1 + X_2;\ S_3 = X_1^3 + X_2^3.$$

Предположим, что $\tilde{o}(x) = x^2 + \tilde{o}_1 x + \tilde{o}_2$ – полином, обладающий, по определению, тем свойством, что локаторы X_1, X_2 являются корнями $\tilde{o}(x)$. Назовем его полиномом локаторов ошибок. По теореме Безу полином локаторов ошибок можно представить следующим образом:

$$\tilde{o}(x) = (x - X_1)(x - X_2) = x^2 - (X_1 + X_2)x + X_1 X_2.$$

Поскольку поле $GF(2^m)$ имеет характеристику 2 , то

$$S_1^3 = (X_1 + X_2)^3 = X_1^3 + X_1^2 X_2 + X_1 X_2^2 + X_2^3$$

и $\tilde{o}_2 = X_1 X_2 = \frac{(S_1^3 - S_3)}{S1}$; $\tilde{o}_1 = -S_1$. Учитывая эти соотношения, получим:

$$\tilde{o}(x) = x^2 - S_1 x + \frac{S_1^3 - S_3}{S_1}.$$

В двоичном случае $-S_1 = S_1$ для всех элементов поля и знак минус можно заменить на плюс, но сохраним знак минус, поскольку приведенные формулы с небольшими изменениями имеют место и в недвоичном случае.

Зная компоненты синдрома S_1 и S_3, можно вычислить коэффициенты полинома локаторов ошибок. Декодирование происходит следующим образом:

1-й случай. Пусть $S_1 = 0$ и $S_3 = 0$. В этом случае предполагается, что ошибок не было. В качестве результата принимается принятое слово.

2-й случай. Допустим, $S_1 \neq 0$, $S_3 = S_1^3$. Тогда $\tilde{o}(x) = x^2 - S_1 x = x(x - S_1)$. Значит, $x_2 = 0$, $x_1 = S_1$, значит, $\square^i = S_1$, следовательно, $i = log_{\square} S_1$.

Результатом декодирования является слово $c(x)$, которое получается из $y(x)$ инвертированием символа на позиции i.

3-й случай. Пусть $S_1 \neq 0$, $S_3 \neq S_1^3$. Вычисляем многочлен ошибок $\tilde{o}(x)$. Для отыскания позиций ошибок нужно найти 2 ненулевых корня этого многочлена. Корни можно найти, например, перебирая все ненулевые элементы поля $GF(2^m)$. При двух ошибках будут найдены два локатора X_1 и X_2. Тогда $i = log_{\square} X_1$, $j = log_{\square} X_2$.

Примеры решения задач

Пример 9.1. Найти все неприводимые многочлены над полем Z_2 при $n = 4$.

Решение.

Заметим, что у неприводимого многочлена свободный член не равен 0 и количество слагаемых нечётно. Выпишем кандидатов (их 4):

$x^4 + x^2 + 1;$
$x^4 + x^3 + 1;$
$x^4 + x + 1;$
$x^4 + x^3 + x^2 + x + 1.$

Все эти многочлены корней в Z_2 не имеют. Но это не означает неприводимость многочлена. Неприводимость означает невозможность разложения.

1. Разложим многочлен $g(x) = x^4 + x^2 + 1 = (x^2 + ax + n)(x^2 + cx + d) = x^4 + (c + a)x^3 + (d + ac + b)x^2 + (ad + bc)x + bd$. Приравнивая коэффициенты при одинаковых степенях x, получим систему уравнений:

$$\begin{cases} a + c = 0, \\ d + ac + b = 1, \\ ad + bc = 0, \\ bd = 1 => b = 1, d = 1. \end{cases}$$

$$\begin{cases} a + c = 0, \\ 1 + ac + 1 = 1, \\ a + c = 0. \end{cases}$$

$$\begin{cases} a + c = 0, \\ 1 + ac + 1 = 1, \\ a + c = 0. \end{cases}$$

$$\begin{cases} a + c = 0 \\ ac = 1 \end{cases}$$

$$\begin{cases} a + c = 0 \\ ac = 1 => a = 1, c = 1. \end{cases}$$

Следовательно, многочлен приводимый и раскладывается на произведение неприводимых многочленов: $x^4 + x^2 + 1 = (x^2 + x + 1)^2$

2. Аналогично рассмотрим многочлен $g(x) = x^4 + x^3 + 1 = (x^2 + ax + b)(x^2 + cx + d)$.

$$\begin{cases} a + c = 1, \\ d + ac + b = 0, \\ ad + bc = 0, \\ bd = 1 => b = 1, d = 1. \end{cases}$$

$$\begin{cases} a + c = 1, \\ 1 + ac + 1 = 0, \\ a + c = 0. \end{cases}$$

$$\begin{cases} a + c = 1, \\ ac = 0, \\ a + c = 0. \end{cases}$$

Система не имеет решения.

Многочлен неприводимый, так как не раскладывается на простые множители.

3. Рассмотрим многочлен $g(x) = x^4 + x + 1$.Аналогично получим систему:

$$\begin{cases} a + c = 0, \\ d + ac + b = 0, \\ ad + bc = 1, \\ bd = 1 \Rightarrow b = 1, d = 1. \end{cases}$$

$$\begin{cases} a + c = 0, \\ ac = 0, \\ a + c = 1. \end{cases}$$

Система не имеет решения.

Многочлен неприводимый, так как не раскладывается на простые множители.

4.Рассмотрим многочлен $g(x) = x^4 + x^3 + x^2 + x + 1$.

$$\begin{cases} a + c = 1, \\ d + ac + b = 1, \\ ad + bc = 1, \\ bd = 1 \Rightarrow b = 1, d = 1. \end{cases}$$

$$\begin{cases} a + c = 1, \\ ac = 1, \\ a + c = 1. \end{cases}$$

Система не имеет решения.

Многочлен неприводимый, так как не раскладывается на простые множители.

Итак, неприводимые многочлены над полем Z_2 при n = 4:

$x^4 + x^3 + 1;$

$x^4 + x + 1;$

$x^4 + x^3 + x^2 + x + 1.$

Пример 9.2. Определить, является ли данный неприводимый многочлен $x^2 + x + 2$ примитивным в поле $GF(5^2)$.

Решение.

$a^2 = -a - 2 = 4a + 3; a^3 = 4a + 2; a^4 = 3a + 2; a^5 = 4a + 4; a^6 = 2; a^7 = 2a; a^8 = 3a + 1; a^9 = 3a + 4; a^{10} = a + 4; a^{11} = 3a + 3; a^{12} = 4; a^{13} = 4a; a^{14} = a + 2; a^{15} = a + 3; a^{16} = 2a + 3; a^{17} = a + 1; a^{18} = 3; a^{19} = 3a; a^{20} = 2a + 4; a^{21} = 2a + 1; a^{22} = 4a + 1; a^{23} = 2a + 2; a^{24} = 1.$

Этот многочлен является примитивным.

Задачи

Задача 9.1. Циклический код порождается многочленом $g(x) = x^3 + x + 1$.

а) Закодируйте число 78.

б) Закодируйте число 1010.

в) Найдите и исправьте ошибку принятой кодовой комбинации 0111000.

г) Найдите и исправьте ошибку в принятой кодовой комбинации 0111001.

Задача 9.2. Представьте порождающий многочлен 11001 в полиномиальном виде.

Задача 9.3.Представить в виде многочленов следующие комбинации циклического кода: 0111110, 0011111, 1001111.

Задача 9.4.Определить количество информационных разрядов в коде длиной 15 символов, обнаруживающем две и исправляющем одну ошибку.

Задача 9.5.Какой вид имеет комбинация циклического кода, информационная часть которого $i(x)=1011$, а порождающий многочлен $g(x)=1101$?

Задача 9.6.Построить несколько комбинаций циклического кода с числом информационных разрядов k=11 и порождающим многочленом x^4+x^3+1.

Задача 9.7.Обнаружить, какая из трёх принятых комбинаций 0011010, 0110100, 1010011 циклического кода, порождающий многочлен которого $g(x)=x^3+x^2+1$, – ошибочная. Исправить обнаруженную ошибку.

Задача 9.8.Построить циклический код длиной в 15 символов, исправляющий одну.

Коды БЧХ

Они представляют собой подкласс циклических кодов, способных исправлять заданное число ошибок.

1. Выберем число проверочных символов r. Определим длину кодового слова $n = 2^r - 1$. Коды такой длины называются примитивными кодами БЧХ.

Зададим число t ошибок, которые необходимо исправить.

1)Выберем примитивный многочлен степени r и построим поле $GF(2^r)$. Пусть ε - корень этого многочлена. Он является примитивным элементом поля.

Тогда

$x^n+1 = (x + 1)*m_1(x)*...*m_k(x)$.

Известно, что если элемент$\alpha \in GF(2^r)$, то $\alpha^{n-1} = 1$, т. е. α - корень x^n+1, $\alpha = \varepsilon^j$.

Рассмотрим степени $\varepsilon, \varepsilon^2, \varepsilon^3, ...,\varepsilon^{2t}$ и найдем в разложении $x^n+1= (x + 1)*m_1(x)*...*m_k(x)$все многочлены, для которых эти числа будут

корнями, в качестве порождающего многочлена возьмем $НОК(m_1(x), ..., m_{2t}(x))$ и построим код.

Далее выполняется кодирование тем или иным способом.

Примеры решения задач

Пример10.1.Закодировать информационное слово i= 1011010 с помощью кода БЧХ(15, 7, 2). Допустить, что при передаче произошло 2 ошибки. Исправить эти ошибки.

Решение.

Пусть поле $GF(2^r)$ строится по многочлену$x^4 + x + 1$ *(модулю поля)*.

Так как $15=2^4-1$, то r=4, k=11, $x^{15}+1=(x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)$.

В качестве порождающего многочлена используем $g(x) = x^8 + x^7 + x^6 + x^4 + 1 = (x^4+x+1)(x^4+x^3+x^2+x+1)$. Корнями первого множителя являются $\varepsilon, \varepsilon^2, \varepsilon^4$, а второго - ε^3. При t=2 исправляются 2 ошибки.

Найдем кодовое слово по формуле c(x) = i(x) * g(x).

$c(x) = 1 + x^2 + x^3 + x^4 + x^5 + x^9 + x^{12} + x^{13}$.

$c(x)$ = 101111000100110.

Внесем в данное слово 2 ошибки в 10 и в 4 разряды и получим $y(x)$ = 101101000110110 (счёт начинается с нуля).

Проанализируем полученное слово. Многочлен, по которому строится поле: $x^4 + x + 1$. Введем в рассмотрение 2 величины: $s_1 = y(\varepsilon)$; $y(x) = 1 + x^2 + x^3 + x^5 + x^9 + x^{10} + x^{12} + x^{13}$.

Вычислим$y(\varepsilon) = 1 + \varepsilon^2 + \varepsilon^3 + \varepsilon^5 + \varepsilon^9 + \varepsilon^{10} + \varepsilon^{12} + \varepsilon^{13}$.

Составим таблицу элементов поля:

0	0
1	1
ε	ε
ε^2	ε^2
ε^3	ε^3
ε^4	$\varepsilon + 1$
ε^5	$\varepsilon^2 + \varepsilon$
ε^6	$\varepsilon^3 + \varepsilon^2$
ε^7	$\varepsilon^3 + \varepsilon + 1$
ε^8	$\varepsilon^2 + 1$
ε^9	$\varepsilon^3 + \varepsilon$
ε^{10}	$\varepsilon^2 + \varepsilon + 1$

ε^{11}	$\varepsilon^3 + \varepsilon^2 + \varepsilon$
ε^{12}	$\varepsilon^3 + \varepsilon^2 + \varepsilon + 1$
ε^{13}	$\varepsilon^3 + \varepsilon^2 + 1$
ε^{14}	$\varepsilon^3 + 1$
ε^{15}	1

$y(\varepsilon) = 1 + \varepsilon^2 + \varepsilon^3 + \varepsilon^2 + \varepsilon + \varepsilon^3 + \varepsilon + \varepsilon + 1 + \varepsilon^2 + \varepsilon^2 + \varepsilon^3 + \varepsilon + 1 + 1 + \varepsilon^2 + \varepsilon^3 = \varepsilon^2$.

Это компонент синдрома $S_1 = \varepsilon^2$.

$S_3 = y(\varepsilon^3) = \varepsilon^3 + \varepsilon^2 + \varepsilon$.

Найдем локаторы ошибок x_1, x_2как корни уравнения $x^2 - S_1 x + \frac{S_1^3 - S_3}{S_1} = 0$. Получим, что $x_1 = \varepsilon^4, x_2 = \varepsilon^{10}$.

Инвертируем биты номер 4 и номер 10 и получим исходное кодовое слово 101111000100110.

Задачи

Задача 10.1.Построить код БЧХ с n=31 и t=2.

Задача 10.2.Показать процесс исправления двойной ошибки в БЧХ коде (15; 7).

Задача 10.3.Построить циклический код, способный исправлять шестикратную ошибку при общей длине кода n=63. Показать процесс исправления шестикратной ошибки.

Задача 10.4.Построить код БЧХ, способный исправить 14-кратную ошибку при общей длине кода n=63. Проверить корректирующие способности кода.

Сверточные коды

В современной информационной технике сверточные коды играют такую же важную роль как и блоковые. Сегодня сверточные коды играют ведущую роль в современных системах связи. Это в первую очередь относится к цифровому радиовещанию и к мобильной связи сети GSM.

Важнейшими отличиями сверточных кодов от блоковых являются следующие:

1. Сверточные коды позволяют производить кодирование и декодирование потоков данных непрерывно во времени.

2. Сверточные коды не нуждаются в блоковой синхронизации.

3. Применение сверточных кодов позволяет достичь очень высокой надежности передаваемой информации. «Хорошие» сверточные коды могут быть найдены путем моделирования.

Мы ограничимся изложением только самых необходимых теоретических основ и приведем типичные примеры применения сверточных кодов.

Термин «сверточные коды» возник из теории инвариантных линейных систем LTI (LinearTimeInvariant – англ.). В теории систем LTI сверткой называют характерный признак некоторой линейной операции. С точки зрения этой теории, кодирование является отображением информационной последовательности символов в кодовую
последовательность с помощью линейной схемы с параметрами, не меняющимися во времени. Такое отображение наглядно показано на рис. 1. Последовательность информационных символов поступает в демультиплексор, который разлагает входной поток на *k* самостоятельных подпоследовательностей. Схему на рис. 1 можно также интерпретировать как совместное кодирование *k* независимых информационных последовательностей. Кодирование производится с помощью дискретной во времени схемы *LTI* с *k* входами и *n* выходами. Эта схема характеризуется тремя параметрами (*n*, *k*, *m*), причем, параметр *m* определяется внутренней конструкцией кодера.

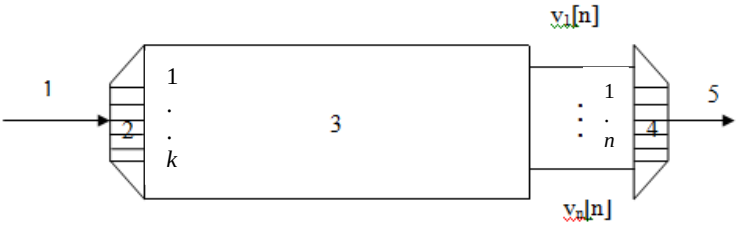


Рис. 1

Составляющие схемы на рис. 4:1 - поток данных (информационные символы);2 - демультиплексор;3 - LTI (n, k, m);4 - мультиплексор;5 - кодовое слово v.

На практике, как правило, используются двоичные сверточные коды, поэтому мы будем говорить о последовательностях битов. В этом случае, под линейностью схемы

мы подразумеваем выполнение этой схемой всех операций но правилам арифметики
по модулю 2. (т.е. в поле *GF(2)*). Так как декодер представляет собой дискретную схему, для его описания будем использовать методы теории обработки дискретных сигналов. Придерживаясь терминологии этой теории, будем говорить об информационной последовательности $u[n]$, входных последовательностях $u_1[n]$, $u_2[n]$, … $u_k[n]$, выходных последовательностях $v_i[n]$, $v_2[n]$,..., $v_k[n]$ и кодовой последовательности $u[n^2]$.

Важнейшим параметром схемы является ***импульсный отклик***. В мультиплексоре все выходные последовательности объединяются в одну кодовую последовательность путем поочередного считывания элементов из последовательностей $v_1[n]$, $v_2[n]$, ..., $v_n[n]$ при каждом фиксированном значении n. Для того, чтобы записать связь между параметрами кодера рис. 1 в компактной форме и при этом избежать путаницы, введем единые обозначения. Фигурными скобками будем обозначать любою последовательность символов, причем, нижние индексы используются для нумерации элементов последовательностей. Для импульсного отклика системы первый нижний индекс определяет номер выходной последовательности, а второй - номер входной.

1) ***Входные последовательности*** $\{u_j[n]\} = \{uj,0, uj,1, \ldots\}$ $j = 1, \ldots, k;$
2) ***Выходные последовательности*** $\{v_j[n]\} = \{v_{j,0}, v_{j,1}, \ldots\}$ $j = 1, \ldots, n;$
3) ***Информационная последовательность*** $\{u[n]\} = \{u_{1,0}, u_{2,1}, \ldots, u_{k,0}, u_{1,1}, u_{2,1}, \ldots, u_{k,1}, \ldots\};$
4) ***Кодовая последовательность*** $\{v[n]\} = \{v_{1,0}, v_{2,1}, \ldots, v_{n,0}, v_{1,1}, v_{2,1}, \ldots, v_{n,1}, \ldots\}.$
5) ***Импульсный отклик*** $g_{ji}[n] = \{g_{ji,0}, g_{ji,1}, \ldots, g_{ji,mi}\}.$

С учетом введенных обозначений операцию свертки, выполняемую схемой рис. 1, можно записать в виде

$$v_j[n] = \sum_{i=1}^{k} g_{ji}[n] \text{❇} u_i[n] = \sum_{i=1}^{k} \left(\sum_{m=0}^{mi} g_{ji}[m] * u_i[n-m]\right).$$

Замечание. Здесь символ ❇ обозначает операцию линейной свертки. В дальнейшем мы будем рассматривать также циклическую свертку «*».

Эта операция выполняется для всех выходов j. Здесь индексы, стоящие в квадратных скобках, определяют нормированные переменные времени, то есть они указывают номера элементов соответствующих последовательностей. Нумерация элементов, как

правило, начинается с нуля, поэтому $u_2[5]$, например, обозначает 6-ой элемент второй входной последовательности. Элементы с отрицательной временной переменной полагаются равными нулю.

Замечание.В дальнейшем, если это не будет приводить к путанице, лишние индексы будем опускать.

Аналогично блоковым кодам, сверточные коды могут быть описаны с помощью многочленов. При этом становятся очевидными не только сходство этих кодов, но и их различия.

Будем рассматривать импульсные отклики кодеров сверточных кодов как порождающие многочлены степени m_i:

$g_{ji}(X) = g_{ji,0} + g_{ji,1}X + \ldots + g_{ji,mi} + X^{mi}$

Переменная X здесь играет роль ≪указателя сдвига≫ и никакой смысловой нагрузки больше не несет, X^n означает n-кратный сдвиг относительно некоторой точки отсчета (например, начала входной последовательности).

Замечание. В литературе вместо переменной X очень часто используют
переменную D (от слова delay задержка - англ.).

Так как мы по прежнему рассматриваем только двоичные коды, все коэффициенты многочленов принадлежат $GF(2)$ и все операции над многочленами выполняются по правилам арифметики по модулю 2.

Аналогично блоковым кодам, процесс кодирования сверточных кодов может быть описан с помощью порождающих многочленов. Если входная последовательность имеет конечную длину, то мы фактически имеем дело с блоковым кодированием, поэтому, j-ю
выходную последовательность можно представить в виде

$$v_j(X) = \sum_{i=1}^{k} g_{ij}(X)u_i(X).$$

Так как имеется n выходов и кодовый многочлен образуется их перемешиванием, удобно записать

$$v(X) = \sum_{j=1}^{n} X^{j-1} v_j(X^n)$$

В этом случае $v(X) = \sum_{j=1}^{n} X^{j-1} \sum_{i=1}^{k} g_{ij}(X^n)u_i(X^n)$

Пример.Представление сверточного (2,1,3)-кода в виде многочлена.

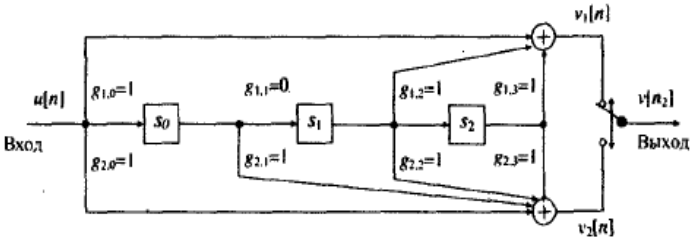


Рис. 2

В соответствии с рис. 2, имеем
$g_1 = 1 + X^2 + X^3$ и $g_2 = 1 + X + X^2 + X^3$.

В качестве информационной последовательности выберем $\{u[n]\} = \{1,0,1,1,1\}$.

Этой последовательности соответствует многочлен $u(X) = 1 + X^2 + X^3 + X^4$, поэтому

$v1(X) = u(X)*g1(X) = (1 + X^2 + X^3 + X^4)(1 + X^2 + X^3) = 1 + X^7$;

$v2(X) = u(X)*g2 = 1 + X + X^3 + X^4 + X^5 + X^7$.

Кодовый многочлен определяется как $v(X) = v_1(X^2) + Xv_2(X^2) = 1+X + X^3 + X^7 + X^9 + X^{11} + X^{14} + X^{15}$, что соответствует кодовому слову $\{u[n^2]\} = \{1,1,0,1,0,0,0,1,0,1,0,1,0,0,1,1\}$.

Граф состояний

Регистры кодеров содержат ограниченное число двоичных разрядов, следовательно, число состояний, в котором может находиться кодер, всегда конечно. Именно поэтому процесс кодирования можно описать как последовательность смены состояний кодера. Такой подход является ключевым и ведет к более глубокому пониманию свойств сверточных кодов. Более того, он способствует разработке эффективных алгоритмов кодирования и декодирования. Мы разовьем эту идею на нескольких примерах.

Пример. Описание состояний сверточного (2,1,3)-кода.

Скопируем схему кодера рис. 2 с единственной разницей: вместо разрядов регистра сдвига поставим переменные X_1, X_2 и X_3. Такая модифицированная схема представлена на рис. 6.

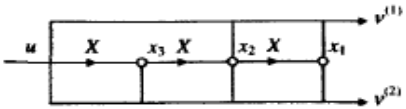


Рис. 3

Так как X_1, X_2, X_3 - двоичные переменные, то кодер, изображенный на рис. 3, может находиться в одном из $2^3 = 8$ состояний. Эти состояния пронумерованы в лексикографическом порядке и показаны на рис. 4. Нумерация состояний может быть произвольной, однако, лексикографический порядок наиболее удобен для программной реализации.

Смена состояний регистра кодера происходит следующим образом: переменные X_1и X_2 заменяются на X_2 и X_3 соответственно, а в разряд X_3 загружается новый информационный бит, поэтому, каждое состояние может переходить только в два последующих в зависимости от того, ≪0≫ или ≪1≫ были поданы на вход регистра сдвига.

Все возможные смены состояний, их зависимость от загружаемого бита, а также кодовые символы для этих переходов показаны на рис.5. Процесс кодирования начинается с S_0, т.е. состояния, в котором $X_1 = X_2 = X_3 = 0$.

1. Если первый информационный бит равен ≪0≫, кодер остается в состоянии S_0. На выход выдаются кодовые символы ≪00≫. Такой исход на рис. 5 обозначен через ≪0/00≫.

Состояние регистра			Состояние S_i
x_3	x_2	x_1	$i = x_3 + 2x_2 + 2^2x_1$
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

Рис. 4

2. Если первый информационный бит равен ≪1≫, кодер переходит в состояние S_1 (см рис. 4). Выход в этом случае равен ≪11≫. Этот переход обозначен ≪1/11≫. Продолжая этот процесс

вправо, можно построить всю диаграмму состояний (рис.5). Диаграмма рис. 5 содержит всю информацию о сверточном коде.

Кодирование информационной последовательности эквивалентно движению по некоторому неразрывному пути по диаграмме состояний. При программной реализации, например, кодирование может быть наиболее эффективно осуществлено исключительно с помощью заранее записанных в памяти таблиц переходов между состояниями.

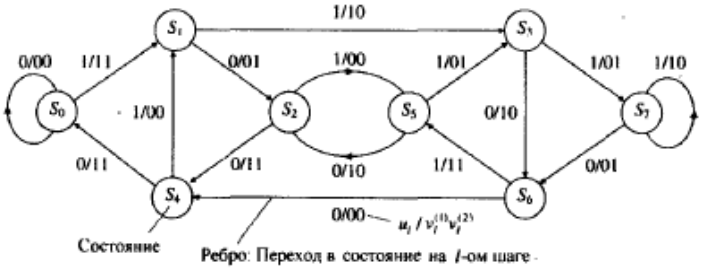


Рис. 5

Рассмотрим такое альтернативное кодирование на примере.

Пример.Кодирование сверточного (2,1,3)-кода с помощью таблиц переходов.

Выпишем для каждого состояния два его последующих, в зависимости от значения очередного бита (≪0≫ или ≪1≫). Для этих путей выпишем также соответствующие пары кодовых бит. Полученные результаты отражены на рис. 6.

Рассмотрим кодирование информационной последовательности *u[n] = 1,0,1,1,1*. Процесс начнем из состояния S_0 и в нем же и закончим, добавив к *u[n]*≪хвост≫ из трех нулей. В результате получим последовательность состояний
$\{S[n]\} = \{S_0,S_1,S_2,S_5,S_3,S_7,S_6,S_4,S_0\}$ и кодовое слово $\{v[n]\} = \{1,1,0,1,0,0,0,1,0,1,0,1,0,0,1,1\}$.

Состояние	Новое состояние $S[n+1]$		Кодовые биты	
	0	1	0	1
0	0	1	00	11
1	2	3	01	10
2	4	5	11	00
3	6	7	10	01
4	0	1	11	00
5	2	3	10	01
6	4	5	00	11
7	6	7	01	10

Рис. 6

Примеры решения задач

Пример 11.1. Используя сверточный код, закодировать последовательность 110110.

$$g_1(z) = 1 + x^2;\ g_2(z) = x^2.$$

Решение.

Построим кодер (рис. 7)

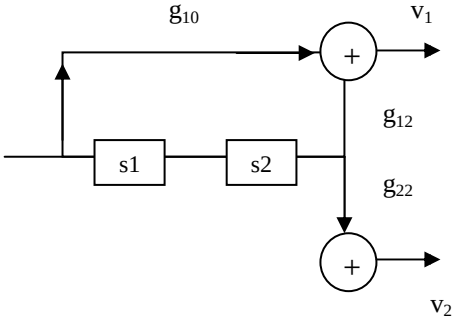


Рис. 7

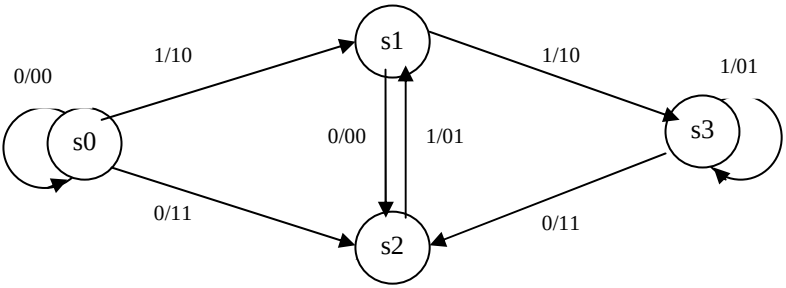
Состояния кодера:

Состояние кодера	Состояние s_i
00	s_0
10	s_1
01	s_2
11	s_3

Построим граф состояний, опираясь на следующую таблицу:

из s0 (00)		
0	00	00
1	10	10
из s1 (10)		
0	01	00
1	11	10
из s2 (01)		
0	00	11
1	10	01
из s3 (11)		
0	01	11
1	11	01

Граф состояний (рис. 8):



Составим таблицу переходов (табл. 1.):

Состояние	Новое состояние		Кодовые биты	
	0	**1**	**0**	**1**
0	0	1	00	10
1	2	3	00	10
2	0	1	11	01
3	2	3	11	01

Таблица 1.

Закодируем теперь последовательность 110110. Дополним ее нулями до нужного количества бит: 11011000. Получим последовательность переходов из состояния в состояние.

$s_0 \to s_1 \to s_3 \to s_2 \to s_1 \to s_3 \to s_2 \to s_0 \to s_0$

Получимпоследовательность: 1010110110111100.

Пример 11.2. Закодировать последовательность 110110 с помощью многочленов.

$u(x) = 1 + x + x^3 + x^4$;

$g_1(x) = 1 + x^2$;

$g_2(x) = x^2$.

Решение.

$v_1(x) = (1 + x + x^3 + x^4)(1 + x^2) = 1 + x^2 + x + x^3 + x^5 + x^4 + x^6 = 1 + x^2 + x + x^5 + x^4 + x^6$.

$v_2(x) = (1 + x + x^3 + x^4)x^2 = x^2 + x^3 + x^5 + x^6$.

$v(x) = v_1(x^2) + xv_2(x^2) = 1 + x^4 + x^2 + x^{10} + x^8 + x^{12} + x^5 + x^7 + x^{11} + x^{13} = 1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{13}$.

Получим последовательность: 101011011011110.

.**Пример 11.3.**Пусть g_1= 1011, g_2 = 1111. Изобразить кодер и закодировать последовательность 101

Решение.

Изобразим кодер (рис. 9)

Изображенный на рис. кодер имеет n = 2 выходов, $к$ = 1 входов и m= 3 (так как регистр кодера содержит три разряда *s0, s1, s2*).

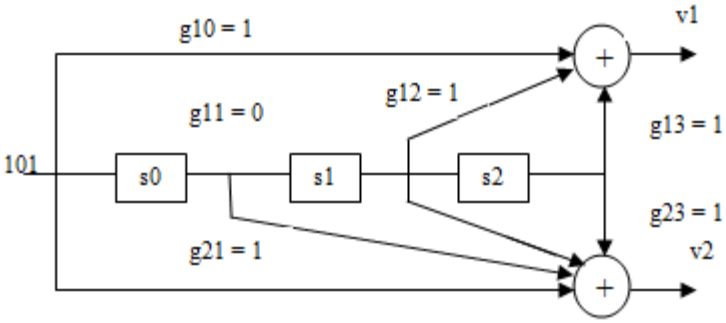


Рис. 9

Две выходные последовательности $v_1[n]$ и $v_2[n]$ можно генерировать с помощью двух фильтров с конечными импульсными откликами,$g_i[n]$ и $g_2[n]$ из входной

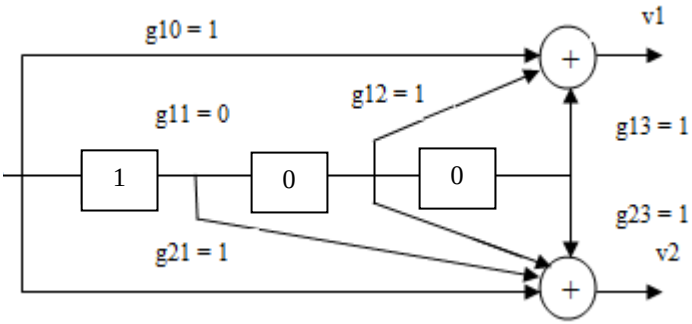
последовательности $u[n]$. Импульсныеотклики $g_1[n]$ и $g2[n]$можно получить непосредственно из рис. 12.

Рассмотрим работу этой схемы.

Такт 0. Состояние регистров и выход.

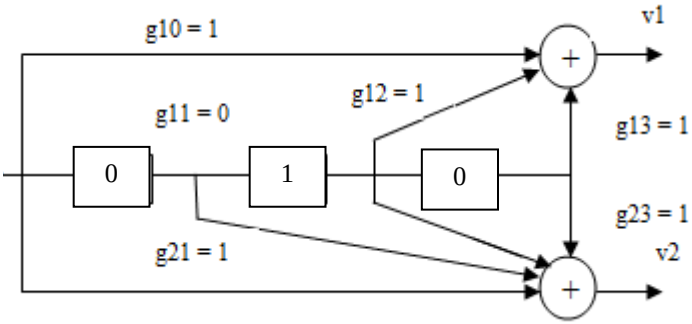
Состояние регистров	v_1	v_2
000	**0**	**0**

Такт 1. Поступила 1.



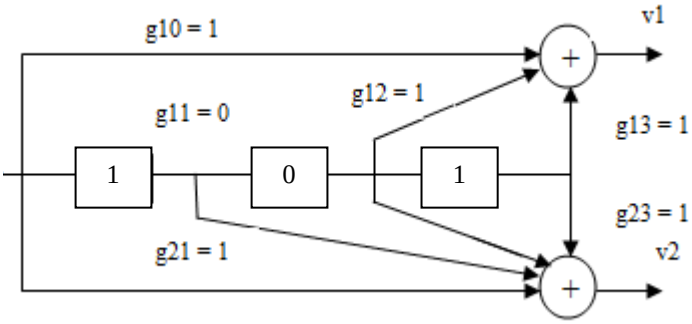
v1= 1;
v2= 1.

Такт 2. Поступил 0.



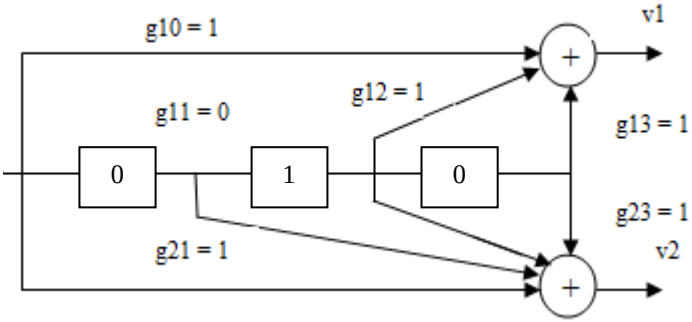
v1= 0;
v2= 1.

Такт 3. Поступила 1.



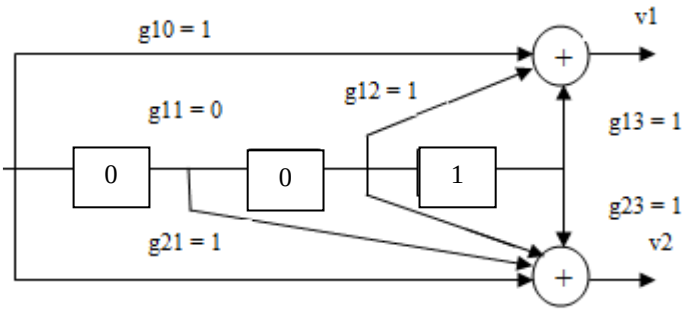
v1 = 0;
v2= 0.

Такт 4.Поступил 0.



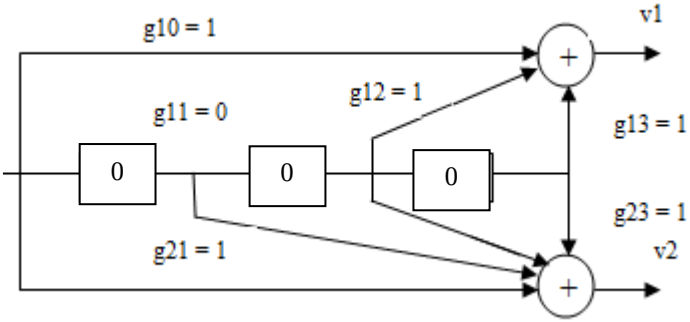
v1=1 ;
v2=0 .

Такт 5. Поступил 0.



v1= 1;
v2= 1.

Такт 6. Поступил 0.



v1= 1;
v2= 1.
Последовательность v1 = 100111.
Последовательность v2 = 110011.
В мультиплексоре на выходе формируется последовательность: 110100101111.

Пример 11.4.

Имеется один вход и один выход. Пусть m = 3 (в импульсном отклике 4 элемента); g[n] = {1011}; u[n] = {101}. Найти выходную последовательность, используя свертку.

Решение.

v[n] = g[n] * v[n] = .
1. v[0] = = g[0]*u[0] + g[1]*u[-1] + g[2]*u[-2] + g[3]*u[-3] = 1;
2. v[1] = = g[0]*u[1] + g[1]*u[0] + g[2]*u[-1] + g[3]*u[-2] = 0;
3. v[2] = g[0]*u[2] + g[1]*u[1] + g[2]*u[0] + g[3]*u[-1] = 0;
4. v[3] = = g[0]*u[3] + g[1]*u[2] + g[2]*u[1] + g[3]*u[0] = 1;
5. v[4] = = g[0]*u[4] + g[1]*u[3] + g[2]*u[2] + g[3]*u[1] = 1;
6. v[5] = = g[0]*u[5] + g[1]*u[4] + g[2]*u[3] + g[3]*u[2] = 1.

Полученная последовательность: 100111.

Задачи

Все задачи решить 4 способами: при помощи свертки, при помощи графа состояния, с помощью многочленов и с помощью таблиц переходов.

Задача11.1.Пусть g1 = 1111, g2 = 1001. Изобразить кодер и закодировать последовательность 101.

Задача11.2.Пусть g1 = 0110, g2 = 1111. Изобразить кодер и закодировать последовательность 110

Задача11.3.Пусть g1 = 1010, g2 = 0111. Изобразить кодер и закодировать последовательность 111; 110011..

Задача 11.4.Используя сверточный код, закодировать последовательность 110001.

$g_1(z) = 1 + x^2$; $g_2(z) = x^2$.

ПРИЛОЖЕНИЯ

П. 1. Самостоятельная работа студентов

Студенты специальности10.05.03 (090303) "Информационная безопасность автоматизированных систем"выполняют курсовую работу или ИДЗ. Тему для курсовой работы можно выбрать из следующего списка:

Темы КР или ИДЗ

1. Теоретико - информационные характеристики канала связи.
2. Методы побуквенного кодирования и декодирования. Оптимальное кодирование.
3. Арифметическое кодирование и декодирование.
4. Процедуры сжатия и распаковки LZW.
5. Помехоустойчивое кодирование и декодирование. Коды Хемминга.
6. Циклическое кодирование и декодирование.
7. Коды БЧХ. Кодирование и декодирование.
8. Кодирование и декодирование сверточных кодов.

Примечание.Студент может предложить тему его КР или ИДЗ самостоятельно, предварительно согласовав её с преподавателем.

Указания к выполнению некоторых КР и ИДЗ

1. Методы побуквенного кодирование и декодирования. Оптимальное кодирование и декодирование

Студенту необходимо освоить алгоритмы оптимального кодирования и декодирования по данной теме. Построить программный модуль, в котором будут находиться процедуры кодирования и декодирования файлов. При вызове процедуры кодирования, подпрограмма должна запросить у пользователя используемые символы и их вероятности, размер блока для кодирования (посимвольно, биграммами, триграммами и т.д.), а также имя файла. Во время кодирования подпрограмма должна создать новый файл, куда будет записываться закодированная информация. При вызове процедуры декодирования, подпрограмма должна запросить у пользователя используемые символы и их вероятности, размер блока декодирования, а также имя закодированного файла. Далее подпрограмма должна декодировать файл и записать полученную информацию в новый файл.

2. Помехоустойчивое кодирование и декодирование. Коды Хемминга

Студенту необходимо освоить алгоритмы кодирования и декодирования по данной теме. Построить программный модуль, в котором будут находиться процедуры кодирования и декодирования файлов. При вызове процедуры кодирования, подпрограмма должна запросить у пользователя имя файла. Далее подпрограмма запрашивает у пользователя параметры кода, строит порождающую матрицу и кодирует файл, разбивая данные в файле на блоки, соответствующие размеру кода. Во время кодирования подпрограмма должна создать новый файл, где будет храниться закодированная информация. Реализовать подпрограмму имитации передачи файла по каналу связи. Целью подпрограммы будет внесение произвольных ошибок в закодированный файл. При внесении ошибок учесть корректирующую способность кода. При вызове процедуры декодирования подпрограмма должна запросить у пользователя имя закодированного файла с возможными ошибками и параметры кода. Далее подпрограмма должна локализовать и исправить все ошибки. Если обнаружены ошибки, локализовать которые не удалось, вывести сообщение о повреждении данных с указанием повреждённого блока. Во время декодирования подпрограмма должна создать новый файл, где будет храниться декодированная информация. В случае успеха, входной файл и файл с декодированной информацией должны быть идентичны.

3. Арифметическое кодирование и декодирование

Студенту необходимо освоить алгоритмы арифметического кодирования и декодирования. Построить программный модуль, в котором будут находиться процедуры кодирования и декодирования файлов. При вызове процедуры кодирования, подпрограмма должна запросить у пользователя используемые символы и их вероятности, а также имя файла. Далее подпрограмма запрашивает у пользователя размер блоков и кодирует файл, разбивая данные в файле на блоки по nбит. Во время кодирования подпрограмма должна создать новый файл, куда будут записываться полученные числа. При вызове процедуры декодирования, подпрограмма должна запросить у пользователя используемые символы и их вероятности, а также имя закодированного файла и размер блоков. Далее подпрограмма

должна декодировать числа и записать полученные последовательности в новый файл.

4. Процедуры сжатия и распаковки LZW

Студенту необходимо освоить алгоритмы сжатия и распаковкиLZW. Построить программный модуль, в котором будут находиться процедуры сжатия и распаковки файлов. При вызове процедуры сжатия, подпрограмма должна запросить у пользователя имя файла. Во время сжатия подпрограмма должна создать новый файл, где будет храниться сжатая информация. При вызове процедуры распаковки, подпрограмма должна запросить у пользователя имя сжатого файла. Во время распаковки подпрограмма должна создать новый файл, где будет храниться распакованная информация.

Пример плана оформления пояснительной записки поКР или ИДЗ

Курсовая работа
Процедуры сжатия и распаковки LZW

1. Титульный лист.
2. Введение.
3. Постановка задачи.
4. Описание алгоритма сжатия.
5. Описание алгоритма распаковки.
6. Описание программной реализации (структуры данных, спецификации подпрограмм).
7. Тестовые данные (входная информация, сжатая информация, коэффициент сжатия).
8. График зависимости степени сжатия от объёма входного файла.
9. Исходный код программы.
10, Заключение.
11. Достоинства и недостатки алгоритма.
12. Список использованных источников.

П. 2. Полезная информация
Примитивные многочлены над GF (2)

Степень	Простые многочлены
2	$x^2 + x + 1$
3	$x^3 + x + 1$
4	$x^4 + x + 1$
5	$x^5 + x^2 + 1$
6	$x^6 + x + 1$
7	$x^7 + x^3 + 1$
8	$x^8 + x^4 + x^3 + x^2 + 1$
9	$x^9 + x^4 + 1$
10	$x^{10} + x^3 + 1$
11	$x^{11} + x^2 + 1$
12	$x^{12} + x^6 + x^4 + x + 1$
13	$x^{13} + x^4 + x^3 + x + 1$
14	$x^{14} + x^{10} + x^6 + x + 1$
15	$x^{15} + x + 1$
16	$x^{16} + x^{12} + x^3 + x + 1$
17	$x^{17} + x^3 + 1$
18	$x^{18} + x^7 + 1$
19	$x^{19} + x^5 + x^2 + x + 1$
20	$x^{20} + x^3 + 1$
21	$x^{21} + x^2 + 1$
22	$x^{22} + x + 1$
23	$x^{23} + x^5 + 1$
24	$x^{24} + x^7 + x^2 + x + 1$
25	$x^{25} + x^3 + 1$
26	$x^{26} + x^6 + x^2 + x + 1$
27	$x^{27} + x^5 + x^2 + x + 1$
28	$x^{28} + x^3 + 1$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Иванов, И.В.*Теория информационных процессов систем: учебное пособие. / И.В. Иванов. - Белгород: Изд-во БГТУ, 2007. - 156 с.
2. *Вернер М.*Основы кодирования. / М. Вернер. – Изд-во «Техносфера» 2006 г. – 288с.
3. *ЛидовскийВ.В.*.Теория информации. / В. В. Лидовский – Изд-во М.: Спутник+: Наука, 2004. – 111 с.
4. *ЧечётаС.И.*Введение в дискретную теорию информации и кодирования: учебное издание. / С.И. Чечёта – М.: МЦНМО, 2011. – 224 с.
5. *Цымбал*, В.П. Теория информации и кодирование. / В.П. Цымбал – Киев, Издательское объединение «Вища школа», 1977. – 288 с.
6. *Цымбал*, В.П. Задачник по теории информации и кодированию. / В.П. Цымбал – Киев, Издательское объединение «Вища школа», 1976. – 276 с.
7. *Блейхут, Р.*Теория и практика кодов, контролирующих ошибки. / Р. Блейхут – М.: Изд-во «Мир», 1986. – 576 с.: ил.
8. *Колесник, В.Д.*Кодирование при передаче и хранении информации (Алгебраическая теория блоковых кодов): Учеб.пособие для вузов/В.Д. Колесник – М.: Высш. шк., 2009. – 550 с.: ил.
9. *Михальчик*, Е.В. Описание формата сжатия данных Deflate. Ссылка: http://compression.ru/download/articles/lz/mihalchik_deflate_decoding.html
10. *Сэломон*, Д. Сжатие данных, изображений и звука.Москва:Техносфера, 2004. - 368с.
11. *Питерсон У.*Коды, исправляющие ошибки./*У. Питерсон., Э.Уэлдон* - Изд. "Мир", 1976. - 596 с.

Учебное издание

ТЕОРИЯ ИНФОРМАЦИИ

Методические указания к выполнению практических заданий и индивидуальных домашних заданийдля студентов специальности 10.05.03 (090303) - Информационная безопасность автоматизированных систем

Составители **Сергиенко** Елена Николаевна
Иванова Ксения Сергеевна
Вожакова Юлия Викторовна

Подписано в печать 19.10.15. Формат 60х84/16. Усл. печ. л. 5,1.
Уч.- изд. л. 5,5.
Тираж 35 экз. Заказ Цена
Отпечатано в Белгородском государственном технологическом университете
им. В. Г. Шухова
308012, г. Белгород, ул. Костюкова, 46