Теория кодирования

Ирина Борисовна Просвирнина

- Коды, примеры кодов
- Порождающие матрицы
- Смежные классы группы по подгруппе, теорема Лагранжа
- Декодирование по лидеру смежного класса
- Декодирование по лидеру смежного класса с использованием синдромов
- Коды Хемминга

Коды

Определим **код** как представление множества символов строками, состоящими из единиц и нулей.

Это множество символов обычно включает буквы алфавита, цифры и, как правило, определенные контрольные символы.

Коды представляются бинарными строками с целью использования их компьютерами для хранения и передачи данных.

Коды

Желательно, чтобы коды обладали некоторыми свойствами.

Наиболее важное свойство кода состоит в том, что когда сообщение кодируется как двоичная строка, состоящая из конкатенации элементов кода, эта конкатенация однозначна.

При декодировании сообщения не должно возникать проблем с тем, какую букву представляет элемент кода. Такой код назовем **однозначно декодируемым** кодом.

Коды

Существует несколько способов достижения этой цели.
Один из них – кодирование всех символов двоичными строками одной длины. Такой код называется **блоковым**.
Например, если для кодирования каждого символа используется 8 бит, то известно, что каждые восемь бит представляют один символ передаваемого сообщения.
Блоковый код особенно полезен при ограничении длины кода для каждого посланного символа или буквы.

Коды

Другим способом построения однозначно декодируемого кода является использование префиксного кода.

Код C является **префиксным**, если обладает тем свойством, что элемент кода не может быть начальной строкой другого элемента кода.

Таким образом, при чтении строки из единиц и нулей, изображающей символ A, всегда известен момент завершения строки.

Префиксный код называют также **мгновенным** кодом.

Коды

- Разновидностью префиксного кода является **кома-код**.

При его использовании каждый символ кодируется строкой из единиц, в конце которой стоит нуль.

Значит, множество строк кода имеет вид $\{0, 10, 110, 11110, 111110, \ldots\}$.

Этот код имеет явный недостаток: элементы кода могут быть очень длинными и занимать большой объем памяти.

Коды

Часто необходимо сжимать данные, чтобы минимизировать объем памяти для их хранения или время для передачи данных.

Что касается минимизации объема памяти, то наиболее эффективным является **код Хаффмана**. Преимущество кода Хаффмана состоит также в том, что это мгновенный код.

Хорошо известным примером кода, минимизирующего время передачи, является **код Морзе**.

Коды

В кодах Хаффмана и Морзе буквы или символы, которые встречаются наиболее часто, имеют более короткий код.

Коды

В коде Морзе буквы разделены "пробелами", а слова – тремя "пробелами". В данном случае "пробелы" – это единицы времени.

Код Морзе					
A	· — — —	J	· — — —	S	· · ·
B	— · · ·	K	— · — — —	T	— — —
C	— · — ·	L	· — · ·	U	· · — — —
D	— · ·	M	— —	V	· · · —
E	·	N	— ·	W	· — —
F	· · — ·	O	— — —	X	— · · — — —
G	— — ·	P	· — — ·	Y	— · — —
H	· · · ·	Q	— — · — — —	Z	— — · ·
I	· ·	R	· — ·		

Коды

В процессе передачи данных могут возникать ошибки.

Все, что может стать причиной ошибок, называется неопределенным термином "шум".

Например, данные, полученные от отдаленного космического корабля, наверняка подвержены различного рода шумам.

Коды

В некоторых случаях интерес представляет только определение наличия ошибки, что соответствует ситуации передачи данных еще раз.

Коды, обладающие свойством определения наличия ошибок, называются **кодами, обнаруживающими ошибки**.

Коды

В другом случае, когда данные не могут быть переданы еще раз (например, данные от удаленного космического корабля), требуется дополнительная информация о данных с целью не только обнаружения, но и исправления ошибки.

Коды, позволяющие исправлять ошибки, называются **кодами, исправляющими ошибки**.

Коды

Может показаться разумным всегда использовать коды с исправлением ошибок.

Проблема использования кодов с исправлением ошибок и кодов с обнаружением ошибок состоит в том, что они должны включать в себя дополнительную информацию, поэтому они являются менее эффективными в отношении минимизации объема памяти.

Коды

К сожалению, использование кодов с исправлением ошибок и кодов с обнаружением ошибок не дает абсолютной гарантии того, что ошибка будет исправлена или обнаружена.

Коды

В качестве первого метода обнаружения ошибок рассмотрим бит контроля четности.

Продемонстрируем этот метод на примере кода ASCII.

Коды

ASCII-код является блоковым кодом, который использует 7 битов, поэтому любой закодированный символ изображается строкой из семи символов 1 и 0.

Восьмой бит добавляется таким образом, чтобы количество единиц было четным.

Поэтому, если код переданной строки получен с единственной ошибкой, то количество единиц будет нечетным, и получатель информации поймет, что произошла ошибка.

Коды

К сожалению, если произошло две ошибки, их нельзя будет обнаружить, поскольку количество единиц опять будет четным.

Код ASCII с битом контроля четности							
код	*символ*	*код*	*символ*	*код*	*символ*	*код*	*символ*
00000000	NUL	10100000	SP	11000000	@	01100000	‘
10000001	SOH	00100001	!	01000001	A	11100001	a
10000010	STX	00100010	”	01000010	B	11100010	b
00000011	ETX	10100011	#	11000011	C	01100011	c
10000100	EOT	00100100	$	01000100	D	11100100	d
00000101	ENQ	10100101	%	11000101	E	01100101	e
00000110	ACK	10100110	&	11000110	F	01100110	f
10000111	BEL	00100111	’	01000111	G	11100111	g
10001000	BS	00101000	(	01001000	H	11101000	h
00001001	HT	10101001	)	11001001	I	01101001	i
00001010	LF	10101010	*	11001010	J	01101010	j
10001011	VT	00101011	+	01001011	K	11101011	k
00001100	FF	10101100	’	11001100	L	01101100	l
10001101	CR	00101101	–	01001101	M	11101101	m
10001110	SO	00101110	.	01001110	N	11101110	n
00001111	SI	10101111	/	11001111	O	01101111	o
10010000	DLE	00110000	0	01010000	P	11110000	p
00010001	DC1	10110001	1	11010001	Q	01110001	q
00010010	DC2	10110010	2	11010010	R	01110010	r
10010011	DC3	00110011	3	01010011	S	11110011	s
00010100	DC4	10110100	4	11010100	T	01110100	t
10010101	NAK	00110101	5	01010101	U	11110101	u
10010110	SYN	00110110	6	01010110	V	11110110	v
00010111	ETB	10110111	7	11010111	W	01110111	w
00011000	CAN	10111000	8	11011000	X	01111000	x
10011001	EM	00111001	9	01011001	Y	11111001	y
10011010	SUB	00111010	:	01011010	Z	11111010	z
10011011	ESC	10111011	;	11011011	[	01111011	{
10011100	FS	00111100	<	01011100	\	11111100	|
00011101	GS	10111101	=	11011101	]	01111101	}
00011110	RS	10111110	>	11011110	^	01111110	~
10011111	US	00111111	?	01011111	_	11111111	DEL

Коды

- Рассмотрим код, в котором для кодирования используется простое повторение кодируемой строки заданное число раз.

Например, если при кодировании каждую строку кода нужно повторить один раз, то 10110 будет закодировано как 1011010110.

Если при кодировании каждую строку кода нужно повторить дважды, то 10110 будет закодировано как 101101011010110.

Коды

Если при кодировании каждая строка повторена один раз, то в результате получаем код с обнаружением ошибок.

Если произошла ошибка, то соответствующие позиции не будут совпадать.

Коды

Например, если закодированная строка имеет вид 1111110101101110 11, то ошибки имеются в третьем и в последнем битах.

Исправить ошибки мы не можем, поскольку не знаем, в какой из копий какая ошибка присутствует.

Если кодируемая строка повторяется дважды, то лучше всего выявить ошибку.

Если имеются три копии строки, то можно исправить код при наличии единственной ошибки.

Коды

Если имеется отличие в битах в соответствующих позициях строк, то выбирается значение, которое встречается дважды.

Например, если строка имеет длину 4 и нам передано 110110011101, то во второй позиции мы два раза получим 1 и один раз 0.

Таким образом, предполагаем, что правильное значение равно 1, и правильным вариантом закодированной строки является 1101.

Коды

- Конечно, если ошибка появляется в одной и той же позиции строки более одного раза, возникает проблема.

Если строка повторена так, что имеется n ее копий, то исправление ошибки дает правильный результат, если при повторении ошибка встречается в одной и той же позиции менее, чем $\left\lfloor \frac{n}{2} \right\rfloor$ раз.

Порождающие матрицы

С этого момента мы будем предполагать, что все строки кода имеют фиксированную длину n, и будем трактовать эти строки как векторы или $(1 \times n)$-матрицы.

Следовательно, можно использовать сложение векторов.

Определим сложение по модулю 2, так что $1 + 1 = 0$.

Таким образом,

$$11110001 + 10100111 = 01010110.$$

Порождающие матрицы

- Если B_n – множество всех двоичных строк длины n, то код C – подмножество множества B_n.

Напоминание

Группа представляет собой множество G вместе с бинарной операцией (или произведением) $*$ на $G \times G$, обладающей следующими свойствами:

1) $a * (b * c) = (a * b) * c$ для всех a, b и c из G (т.е. операция $*$ на G ассоциативна).
2) В G существует элемент 1, называемый единицей, который обладает свойством: $a * 1 = 1 * a$ для всех a из G.
3) Для каждого элемента a из множества G существует элемент a^{-1} из множества G, который называется обратным элементом для a и такой, что $a * a^{-1} = a^{-1} * a = 1$.

Напоминание

- Подмножество H группы G является **подгруппой** группы G, если H с той же самой операцией, что и G, также является группой.

Порождающие матрицы

- Если B_n – множество всех двоичных строк длины n, то код C – подмножество множества B_n.

Множество B_n вместе с указанной выше операцией сложения образует **группу** относительно сложения.

Каждая строка из B_n совпадает со своим обратным элементом, так что в этом смысле сложение и вычитание – это одно и то же.

Порождающие матрицы

Код C называется **линейным**, если C – подгруппа группы B_n.

В действительности C – линейное векторное пространство.

Однако, рассматривая код C, мы будем использовать здесь только те свойства, что C является группой и что элемент из C – вектор, поэтому может быть умножен на матрицу соответствующего размера.

Порождающие матрицы

- Будет использован также закон дистрибутивности для матриц, согласно которому

$$A(B + C) = AB + AC$$

для любых матриц A, B и C, произведение которых определено.

Порождающие матрицы

- **Весом строки** c кода, обозначаемым $wt(c)$, называется количество единиц в строке.

Например, если $c = 1011010$, то $wt(c) = 4$.

Порождающие матрицы

- Предположим, что имеется такая $(k \times n)$-матрица G, что ее первые к столбцов и строк образуют единичную матрицу I_k размера $(k \times k)$, все столбцы которой различны.

Таким образом, матрица G имеет вид $[I_k \mid A_{n-k}]$.

Например,

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

является такой матрицей.

Порождающие матрицы

Матрица $G = [I_k \,|\, A_{n-k}]$ называется **порождающей матрицей**.

Рассмотрим строки порождающей матрицы как векторы или строки кода.

Обозначим это множество строк S.

Например, для матрицы

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$S = \{100101, 010110, 001011\}.$$

Порождающие матрицы

Пусть C – код, образованный всеми векторами, которые являются конечными суммами строк из S.

C – подгруппа B_n.

В нашем примере

$$S = \{100101, 010110, 001011\}.$$

Строку 110011 получаем, складывая первые две строки из S, поэтому 110011 будет принадлежать C.

Говорят, что группа C **порождена** множеством S.

Порождающие матрицы

- Говорят, что группа C **порождена** множеством S.

Это S является также минимальным множеством, порождающим код C, поскольку никакие элементы из S не являются суммами других элементов из S.

То, что группа C порождена множеством S, обозначим $\boldsymbol{C = S^*}$.

Код C вида $[I_k | A_{n-k}]$ (т.е. порожденный строками матрицы $[I_k | A_{n-k}]$) называется $\boldsymbol{[n, k]}$**-кодом**.

Порождающие матрицы

Теорема 1

$[n, k]$-код C содержит 2^k строк.

Доказательство.

Первые k битов строки из C определяют элементы C.

Позиции, в которых находятся единицы в первых k битах строки из C, показывают, какие строки из S были просуммированы.

Например, если единицы находятся в первом и третьем битах строки из C, то эта строка получена в результате сложения первой и третьей строк кода C.

Порождающие матрицы

Теорема 1

$[n, k]$-код C содержит 2^k строк.

Доказательство. Первые k битов строки из C определяют элементы C.

Позиции, в которых находятся единицы в первых k битах строки из C, показывают, какие строки из S были просуммированы.

Поскольку существует 2^k различных способов сформировать первые к битов, то в коде C имеется 2^k строк. ■

Порождающие матрицы

- Если необходимо передать строки сообщения длины k, то мы кодируем их, умножая справа на матрицу G.

Таким образом, если $w = w_1 w_2 w_3 \ldots w_k$ или $(w_1, w_2, w_3, \ldots, w_k)$, то мы кодируем это сообщение строкой wG.

Порождающие матрицы

В нашем примере закодируем 110, или $(1,1,0)$, как

$$(1,1,0)\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (1,1,0,0,1,1),$$

или 110011.

Заметим, что строка сообщения совпадает с первыми тремя битами закодированной строки.

Порождающие матрицы

В общем случае строка сообщения длины k будет совпадать с первыми k битами закодированной строки, поскольку единичная матрица I_k, образующая первые k столбцов матрицы

$$G = [I_k \mid A_{n-k}],$$

просто повторяет исходную строку.

Поэтому мы можем осуществлять декодирование, взяв первые k битов закодированной строки.

Порождающие матрицы

Заметим также, что выше, при умножении $(1,1,0)$ на

$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$, мы получили

$$(1,1,0,0,1,1) =$$

$$= 1 \cdot (1,0,0,1,0,1) + 1 \cdot (0,1,0,1,1,0) + 0 \cdot (0,0,1,0,1,1)$$

Таким образом, закодированная строка является суммой векторов из множества

$S = \{100101, 010110, 001011\}$

и, следовательно, принадлежит C, поскольку C – группа.

Порождающие матрицы

В общем случае, если $S = (s_1, s_2, s_3, \dots, s_k)$ – множество строк порождающей матрицы и $w = (w_1, w_2, w_3, \dots, w_k)$ – код сообщения, то закодированная строка имеет вид

$$w_1 s_1 + w_2 s_2 + w_3 s_3 + \cdots + w_k s_k.$$

Она представляет собой сумму строк из S, так как каждое w_i равно либо 1, либо 0 и, следовательно, принадлежит C, поскольку C – группа, порожденная множеством S.

Порождающие матрицы

- Отметим, что G имеет вид $[I_k \mid A_{n-k}]$ и что I_k передает сообщение.

А что делает A_{n-k}?

Рассмотрим снова пример лекции:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Порождающие матрицы

- Если (w_1, w_2, w_3) – строка сообщения, то закодированная строка имеет вид

$$(w_1, w_2, w_3)\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} =$$

$$= (w_1, w_2, w_3, w_1 + w_2, w_2 + w_3, w_1 + w_3)$$

Порождающие матрицы

- $(w_1, w_2, w_3)\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} =$

$= (w_1, w_2, w_3, w_1 + w_2, w_2 + w_3, w_1 + w_3)$

Таким образом, четвертый бит закодированной строки должен быть равен $w_1 + w_2$, пятый бит должен быть равен $w_2 + w_3$ и шестой бит должен быть равен $w_1 + w_3$.

Следовательно, если закодированная строка имеет вид $(w_1, w_2, w_3, w_4, w_5, w_6)$, то $w_4 = w_1 + w_2$, $w_5 = w_2 + w_3, w_6 = w_1 + w_3$.

Порождающие матрицы

- Если закодированная строка имеет вид $(w_1, w_2, w_3, w_4, w_5, w_6)$, то $w_4 = w_1 + w_2$, $w_5 = w_2 + w_3, w_6 = w_1 + w_3$.

Если любая закодированная строка после передачи не удовлетворяет этим соотношениям, то понятно, что при передаче произошла ошибка.

Порождающие матрицы

- Если закодированная строка имеет вид $(w_1, w_2, w_3, w_4, w_5, w_6)$, то $w_4 = w_1 + w_2$, $w_5 = w_2 + w_3, w_6 = w_1 + w_3$.

Например, если получена закодированная строка 101100, то, учитывая, что $w_1 = 1, w_2 = 0$ и $w_3 = 1$, должны иметь, что

$$w_4 = 1 = 1 + 0, w_5 = 0 = 0 + 1, w_6 = 0 = 1 + 1.$$

Поскольку соотношение для w_5 не выполняется, то становится понятным, что имеется ошибка.

Порождающие матрицы

- Таким образом, **матрица A_{n-k} служит для контроля точности передачи данных** так же, как ранее это делал бит контроля четности.

Порождающие матрицы

В общем случае, если имеется закодированная строка $w_1 w_2 w_3 \ldots w_k \ldots w_n$ и

$$R = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & A_{1,k+1} & A_{1,k+2} & \cdots & A_{1,n} \\ 0 & 1 & 0 & \cdots & 0 & A_{2,k+1} & A_{2,k+2} & \cdots & A_{2,n} \\ 0 & 0 & 1 & \ddots & \vdots & A_{3,k+1} & A_{3,k+2} & \cdots & A_{3,n} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 & 1 & A_{k,k+1} & A_{k,k+2} & & A_{k,n} \end{bmatrix},$$

то для $i > k$ имеем

$$w_i = w_1 A_{1,i} + w_2 A_{2,i} + w_3 A_{3,i} + \cdots + w_k A_{k,i}.$$

Порождающие матрицы

$$w_i = w_1 A_{1,i} + w_2 A_{2,i} + w_3 A_{3,i} + \cdots + w_k A_{k,i}$$

для $i > k$.

Закодированная строка должна удовлетворять всем этим $n - k$ соотношениям.

Коды

- Следующая проблема – исправление ошибки, если известно, что произошла единственная ошибка. Изложенный ниже метод известен как **использование лидеров смежных классов**.

Метод будет проиллюстрирован нашим примером, в котором

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Теорема Лагранжа

- Рассмотрим код, в котором для кодирования используется простое повторение кодируемой строки заданное число раз.

Например, если при кодировании каждую строку кода нужно повторить один раз, то 10110 будет закодировано как 1011010110.

Если при кодировании каждую строку кода нужно повторить дважды, то 10110 будет закодировано как 101101011010110.

Теорема Лагранжа

- Конечно, если ошибка появляется в одной и той же позиции строки более одного раза, возникает проблема.

Если строка повторена так, что имеется n ее копий, то исправление ошибки дает правильный результат, если при повторении ошибка встречается в одной и той же позиции менее, чем $\left\lfloor \frac{n}{2} \right\rfloor$ раз.

Лемма 1 Для фиксированной подгруппы H группы G левые смежные классы подгруппы H группы G образуют разбиение группы G.

Доказательство

Каждый левый смежный класс непустой, поскольку для левого смежного класса $a * H$ элемент $a = a * 1$ находится в $a * H$.

Лемма 1 Для фиксированной подгруппы H группы G левые смежные классы подгруппы H группы G образуют разбиение группы G.

Доказательство

Предположим, что пересечение $a * H$ и $b * H$ непустое; пусть элемент с принадлежит пересечению.

Следовательно, $c = a * h = b * h'$ для некоторых h и h' из H.

Умножая обе части уравнения на h^{-1}, получаем

$a * h * h^{-1} = b * h' * h^{-1}$,

поэтому по определению обратного элемента имеем: $a = b * (h' * h^{-1})$.

Лемма 1 Для фиксированной подгруппы H группы G левые смежные классы подгруппы H группы G образуют разбиение группы G.

- Подмножество H группы G является **подгруппой** группы G, если H с той же самой операцией, что и G, также является группой.

Теорема Лагранжа

Лемма 2 Если G – конечная группа и H – подгруппа группы G, то все левые смежные классы подгруппы H группы G содержат одно и то же количество элементов, а именно, количество элементов, которое находится в подгруппе H.

Доказательство

Пусть $a * H$ – левый смежный класс подгруппы H из G.

Определим $f: H \to a * H$ таким образом: $f(h) = a * h$. f – взаимно однозначное соответствие, или биекция. ■

Теорема Лагранжа

Теорема (Лагранж). Если G – конечная группа и H – подгруппа группы G, то порядок H делит порядок G.

Доказательство

Если p – порядок H, q – количество левых смежных классов H в G и n – порядок G, то согласно двум предыдущим леммам $n = pq$. ■

Декодирование по лидеру смежного класса

Итак,

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

так что

$$C = \{000000, 100101, 010110, 001011, 110011, 011101, 101110, 111000\}.$$

Декодирование по лидеру смежного класса

- **Весом строки** c кода, обозначаемым $wt(c)$, называется количество единиц в строке.

Например, если $c = 1011010$, то $wt(c) = 4$.

Декодирование по лидеру смежного класса

Предположим, что имеется такая $(k \times n)$-матрица G, что ее первые к столбцов и строк образуют единичную матрицу I_k размера $(k \times k)$, все столбцы которой различны.

Таким образом, матрица G имеет вид $[I_k \,|\, A_{n-k}]$.

Например,

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

является такой матрицей.

Декодирование по лидеру смежного класса

Матрица $G = [I_k \,|\, A_{n-k}]$ называется **порождающей матрицей**.

Рассмотрим строки порождающей матрицы как векторы или строки кода.

Обозначим это множество строк S.

Например, для матрицы

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$S = \{100101, 010110, 001011\}.$$

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

- Говорят, что группа C **порождена** множеством S.

Это S является также минимальным множеством, порождающим код C, поскольку никакие элементы из S не являются суммами других элементов из S.

То, что группа C порождена множеством S, обозначим $\boldsymbol{C = S^*}$.

Код C вида $[I_k | A_{n-k}]$ (т.е. порожденный строками матрицы $[I_k | A_{n-k}]$) называется $\boldsymbol{[n, k]}$**-кодом**.

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

Теорема 1

$[n, k]$-код C содержит 2^k строк.

Доказательство.

Первые k битов строки из C определяют элементы C.

Позиции, в которых находятся единицы в первых k битах строки из C, показывают, какие строки из S были просуммированы.

Например, если единицы находятся в первом и третьем битах строки из C, то эта строка получена в результате сложения первой и третьей строк кода C.

Коды

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

Теорема 1

$[n, k]$-код C содержит 2^k строк.

Доказательство. Первые k битов строки из C определяют элементы C.

Позиции, в которых находятся единицы в первых k битах строки из C, показывают, какие строки из S были просуммированы.

Поскольку существует 2^k различных способов сформировать первые к битов, то в коде C имеется 2^k строк. ■

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

- Если необходимо передать строки сообщения длины k, то мы кодируем их, умножая справа на матрицу G.

Таким образом, если $w = w_1 w_2 w_3 \dots w_k$ или $(w_1, w_2, w_3, \dots, w_k)$, то мы кодируем это сообщение строкой wG.

Декодирование по лидеру смежного класса

В нашем примере закодируем 110, или $(1,1,0)$, как

$$(1,1,0)\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = (1,1,0,0,1,1),$$

или 110011.

Заметим, что строка сообщения совпадает с первыми тремя битами закодированной строки.

Декодирование по лидеру смежного класса

Пусть C – групповой код, а $C^{\perp}$ – двойственный ему код.

Теорема 2 Строка t принадлежит коду $C^{\perp}$ тогда и только тогда, когда она ортогональна каждой строке из S, множества порождающих элементов кода C.

Теорема 2 Строка t принадлежит коду $C^{\perp}$ тогда и только тогда, когда она ортогональна каждой строке из S, множества порождающих элементов кода C.

Доказательство

Если строка t принадлежит коду $C^{\perp}$, то она ортогональна каждой строке из множества S, поскольку она ортогональна каждой строке из кода C и $S \subseteq C$.

Теорема 2 Строка t принадлежит коду $C^{\perp}$ тогда и только тогда, когда она ортогональна каждой строке из S, множества порождающих элементов кода C.

Доказательство

Предположим, что $S = \{s_1, s_2, s_3, \dots, s_k\}$ и строка t ортогональна строке s_i для всех $s_i \in S$, так что $t \cdot s_i = 0$ для всех $s_i \in S$.

Каждый элемент из C имеет вид

$$w_1 s_1 + w_2 s_2 + w_3 s_3 + \cdots + w_k s_k,$$

где каждое w_i равно 1 или 0.

$$t \cdot (w_1 s_1 + w_2 s_2 + w_3 s_3 + \cdots + w_k s_k) =$$
$$= w_1 (t \cdot s_1) + w_2 (t \cdot s_2) + \cdots + w_k (t \cdot s_k) =$$
$$= 0 + 0 + \cdots + 0 = 0. \blacksquare$$

Декодирование по лидеру смежного класса

- Если (w_1, w_2, w_3) – строка сообщения, то закодированная строка имеет вид

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [I_3 | A_3] .$$

$$G^{\perp} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [A_3^t | I_3] ,$$

$+ w_3)$

Декодирование по лидеру смежного класса

- Если закодированная строка имеет вид $(w_1, w_2, w_3, w_4, w_5, w_6)$, то $w_4 = w_1 + w_2$, $w_5 = w_2 + w_3, w_6 = w_1 + w_3$.

Например, если получена закодированная строка 101100, то, учитывая, что $w_1 = 1, w_2 = 0$ и $w_3 = 1$, должны иметь, что

$$w_4 = 1 = 1 + 0, w_5 = 0 = 0 + 1, w_6 = 0 = 1 + 1.$$

Поскольку соотношение для w_5 не выполняется, то становится понятным, что имеется ошибка.

Декодирование по лидеру смежного класса

- Таким образом, **матрица A_{n-k} служит для контроля точности передачи данных** так же, как ранее это делал бит контроля четности.

Декодирование по лидеру смежного класса

- $$G^{\perp} r_i^t = 0$$

$$G^{\perp}(w_1 r_1^t + w_2 r_2^t + w_3 r_3^t) =$$
$$= w_1 G^{\perp} r_1^t + w_2 G^{\perp} r_2^t + w_3 G^{\perp} r_3^t =$$
$$= 0 + 0 + 0 =$$
$$= 0.$$

Таким образом, если умножить матрицу $G^{\perp}$ на транспозицию любого элемента из C, то получим 0.

В общем случае, если

$$G = [I_k | A_{n-k}] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & A_{1,k+1} & A_{1,k+2} & \cdots & A_{1,n} \\ 0 & 1 & 0 & \cdots & 0 & A_{2,k+1} & A_{2,k+2} & \cdots & A_{2,n} \\ 0 & 0 & 1 & \ddots & \vdots & A_{3,k+1} & A_{3,k+2} & \cdots & A_{3,n} \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 & 1 & A_{k,k+1} & A_{k,k+2} & & A_{k,n} \end{bmatrix},$$

то

$$G^{\perp} = [A^t_{n-k} | I_{n-k}] = \begin{bmatrix} A_{1,k+1} & A_{2,k+1} & \cdots & A_{k,k+1} & 1 & 0 & 0 & \cdots & 0 \\ A_{1,k+2} & A_{2,k+2} & \cdots & A_{k,k+2} & 0 & 1 & 0 & \cdots & 0 \\ A_{1,k+3} & A_{2,k+3} & \cdots & A_{k,k+3} & 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \vdots & \ddots & 0 \\ A_{1,n} & A_{2,n} & & A_{k,n} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Декодирование по лидеру смежного класса

Скалярное произведение i-ой строки матрицы G на j-ю строку матрицы $G^{\perp}$ равно

$$0 + \cdots + 0 + A_{i,j} + 0 + \cdots + 0 + A_{i,j} + 0 + \cdots + 0 = 0,$$

так что в общем случае

$$G^{\perp} r_i^t = 0,$$

где r_i^t – транспозиция i-ой строки матрицы G.

Если умножить матрицу $G^{\perp}$ на транспозицию любого элемента из C, то используя те же рассуждения, мы получим в результате 0.

Декодирование по лидеру смежного класса

Мы также получаем еще один замечательный результат.

Если два элемента b_1 и b_2 из B_n принадлежат одному и тому же смежному классу, образованному в B_n с использованием группы C, то

$$G^{\perp} b_1^t = G^{\perp} b_2^t.$$

Если два элемента b_1 и b_2 из B_n принадлежат одному и тому же смежному классу, образованному в B_n с использованием группы C, то $G^{\perp}b_1^t = G^{\perp}b_2^t$.

Доказательство

Если b_1 и b_2 принадлежат одному смежному классу, то $b_1 = b_2 + c$ для некоторого $c \in C$.

Следовательно, $b_1^t = b_2^t + c^t$ и

$$G^{\perp}b_1^t = G^{\perp}(b_2^t + c^t) =$$
$$= G^{\perp}b_2^t + G^{\perp}c^t =$$
$$= G^{\perp}b_2^t + 0 =$$
$$= G^{\perp}b_2^t,$$

так как $G^{\perp}c^t = 0$ для всех $c \in C$. ■

Декодирование по лидеру смежного класса

- Поскольку $G^{\perp}b^{t}$ одно и то же для всех b из класса смежности, то можно выбрать любое b из класса смежности и определить это значение.

Таким образом, в каждую строку приведенной выше таблицы можно добавить это общее значение образа $G^{\perp}$, так как элементы каждой строки определяют класс смежности.

Мы выбираем лидера смежного класса, потому что он простейший, и помещаем значение его образа во второй столбец.

Эти значения называются **синдромами**.

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

Снова вернемся к примеру:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [I_3 | A_3],$$

$$G^{\perp} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [A_3^t | I_3]$$

Декодирование по лидеру смежного класса

Уже известно, что первый синдром есть

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

Находим, что

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

поэтому второй синдром ест$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$.

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix},$$

поэтому $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ – третий синдром.

Декодирование по лидеру смежного класса

000000	100101	010110	001011	110011	011101	101110	111000
100000	000101	110110	101011	010011	111101	001110	011000
010000	110101	000110	011011	100011	001101	111110	101000
001000	101101	011110	000011	111011	010101	100110	110000
000100	100001	010010	001111	110111	011001	101010	111100
000010	100111	010100	001001	110001	011111	101100	111010
000001	100100	010111	001010	110010	011100	101111	111001
100010	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

Продолжая процесс, получаем следующую таблицу.

000000	0 0 0	100101	010110	001011	110011	011101	101110	111000
100000	1 0 1	000101	110110	101011	010011	111101	001110	011000
010000	1 1 0	110101	000110	011011	100011	001101	111110	101000
001000	0 1 1	101101	011110	000011	111011	010101	100110	110000
000100	1 0 0	100001	010010	001111	110111	011001	101010	111100
000010	0 1 0	100111	010100	001001	110001	011111	101100	111010
000001	0 0 1	100100	010111	001010	110010	011100	101111	111001
100010	1 1 1	000111	110100	101001	010001	111111	001100	011010

Декодирование по лидеру смежного класса

- Предположим, что получена переданная строка 101100.

Умножение матрицы $G^{\perp}$ на транспозицию строки дает

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Декодирование по лидеру смежного класса

Отсюда следует, что 101100 находится в строке 6.

Лидер смежного класса – 000010, элемент крайнего левого столбца строки, содержащей 101100.

Элемент кода C, расположенный в верхней строке столбца, содержащего 101100, равен 101110.

Согласно способу построения таблицы имеем

$$101100 = 101110 + 000010,$$

поэтому можем предположить, что переданная строка 101100 должна иметь вид 101110, и в пятом бите была ошибка.

Декодирование по лидеру смежного класса

Этот метод намного быстрее, поскольку требует только умножить матрицу $G^{\perp}$ на транспозицию строки-сообщения, найти строку, содержащую синдром, и найти переданное сообщение.

Лидер для этого сообщения – индикатор ошибки, а элемент кода C, расположенный в первой строке столбца, содержащего переданное сообщение, есть исправленное сообщение.

Декодирование по лидеру смежного класса

Заметим, однако, что процесс можно сделать еще быстрее.

При этом потребуются только два первых столбца приведенной выше таблицы.

Декодирование по лидеру смежного класса

- Предположим, что получено сообщение 110000.

Умножение матрицы $G^{\perp}$ на его транспозицию дает

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

Декодирование по лидеру смежного класса

- $$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

Потому синдром $-\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, а лидер смежного класса – 001000.

Декодирование по лидеру смежного класса

- Синдром – $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, лидер смежного класса – 001000.

Поскольку лидер указывает, что ошибка появилась в третьем бите, то, добавляя 001000 к 110000, получаем 111000 – исправленный код.

Декодирование по лидеру смежного класса

- Таким образом, этот метод прост.

Умножить транспозицию сообщения на матрицу $G^{\perp}$, чтобы найти синдром.

Далее, нужно найти лидера смежного класса и прибавить его к сообщению, чтобы получить исправленный код.

Обратите внимание на то, что используются только первые два столбца таблицы!

Декодирование по лидеру смежного класса

- Однако, теперь возникла еще одна проблема.

Если полученным сообщением будет строка 101001,

то синдром будет $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

и в соответствующей строке будет три элемента с весом 2.

Декодирование по лидеру смежного класса

- Вспомним, что строка 100010 была выбрана произвольно.

Любая из таких строк равновероятно может содержать ошибку, поэтому в данном случае совершенно безнадежно использовать синдромы для исправления ошибки.

Кроме того, мы пытаемся в данном случае исправить строку с двумя ошибками вместо одной.

Коды Хемминга

В рассматриваемом примере существуют определенные трудности при попытке исправить код для некоторых строк, поскольку все лидеры имеют вес 1.

Эти трудности устраняются путем использования матрицы, называемой матрицей Хемминга, в качестве порождающей матрицы.

Коды Хемминга

Теорема 1

$[n, k]$-код C содержит 2^k строк.

Доказательство. Первые k битов строки из C определяют элементы C.

Позиции, в которых находятся единицы в первых k битах строки из C, показывают, какие строки из S были просуммированы.

Поскольку существует 2^k различных способов сформировать первые к битов, то в коде C имеется 2^k строк. ■

Коды Хемминга

- Если необходимо передать строки сообщения длины k, то мы кодируем их, умножая справа на матрицу G.

Таким образом, если $w = w_1 w_2 w_3 \dots w_k$ или $(w_1, w_2, w_3, \dots, w_k)$, то мы кодируем это сообщение строкой wG.

Коды Хемминга

В нашем примере закодируем 110, или $(1,1,0)$, как

$$(1,1,0)\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} = (1,1,0,0,1,1),$$

или 110011.

Заметим, что строка сообщения совпадает с первыми тремя строками матрицы

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Коды Хемминга

Для изучения матриц Хемминга необходимо понятие расстояния и его связь с весом каждой из строк.

Начнем с теоремы о весах строк.

Коды Хемминга

Теорема 3

Для строк c и c' вес $wt(c + c') \leq wt(c) + wt(c')$.

Доказательство

Пусть $c = c_1 c_2 \dots c_n$ и $c' = c'_1 c'_2 \dots c'_n$.

Если $c_i + c'_i = 1$, то либо $c_i = 1$, либо $c'_i = 1$.

Поэтому существованию каждой единицы в $c + c'$ соответствует существование единицы либо в c, либо в c'. ■

Коды Хемминга

- **Расстояние Хемминга**, или просто расстояние между двумя строками кода c и c', имеющими одинаковую длину, — это число соответствующих битов в строке, где одна строка имеет цифру 1, а другая имеет цифру 0.

Будем обозначать функцию расстояния через $\delta(c, c')$.

Коды Хемминга

- Например, если $c = 101011$ и $c' = 110010$, то $\delta(c, c') = 3$, так как две строки отличаются во второй, третьей и шестой позициях.

Очевидно, что чем больше расстояние между двумя строками кода, тем больше ошибок можно обнаружить.

Поскольку δ названа функцией расстояния, мы должны показать, что она обладает основными свойствами функции расстояния.

Коды Хемминга

Теорема 4

Функция расстояния Хемминга имеет следующие свойства:

a) для строк c и c' расстояние $\delta(c, c') = 0$ тогда и только тогда, когда $c = c'$;

b) для строк c и c' расстояние $\delta(c, c') = \delta(c', c)$;

c) для строк c, c' и c'' выполняется соотношение $\delta(c, c'') \leq \delta(c, c') + \delta(c', c'')$.

Коды Хемминга

Доказательство

Первые два пункта следуют из определения.

Докажем с).

Для строк c и c'' вес $wt(c + c'') = \delta(c, c'')$.

Если $c = c_1 c_2 \dots c_n$ и $c'' = c_1'' c_2'' \dots c_n''$, то $c_i + c_i''$ вносит 1 в вес $wt(c + c'')$ тогда и только тогда, когда $c_i = 0$ и $c_i'' = 0$.

Но это верно в том и только в том случае, когда c_i и c_i'' различны, что вносит 1 в $\delta(c, c'')$.

Коды Хемминга

Доказательство

Заметим, что для любой строки c' строка $c' + c'$ состоит только из нулей.

Обозначим такую строку через $\mathbf{0}$.

$\mathbf{0} + c = c$ для каждой строки c.

Следовательно,

$$\delta(c, c'') = wt(c + c'') =$$
$$wt(c + \mathbf{0} + c'') =$$
$$wt(c + c' + c' + c'') \leq$$
$$wt(c + c') + wt(c' + c'') =$$
$$\delta(c, c') + \delta(c', c'').$$

Коды Хемминга

- Важно знать минимальное расстояние между двумя строками кода.

Если C – код, то **минимальное расстояние кода** C, обозначаемое $D(C)$, равно наименьшему расстоянию между двумя строками из C.

Коды Хемминга

В приведенной далее теореме сформулирован важный критерий для определения числа ошибок, которые могут быть исправлены или обнаружены с использованием кода.

Коды Хемминга

Теорема 5

Для кода C,

a) если $D(C) = k + 1$, то использование кода позволяет обнаружить вплоть до k ошибок;

b) если $D(C) = 2k + 1$, то использование кода позволяет исправить вплоть до k ошибок.

Коды Хемминга

Доказательство

а) Если $D(C) = k + 1$, то для данной строки $c \in C$ отсюда следует, что c отличается от любой другой строки кода по крайней мере в $k + 1$ позициях.

Поэтому, если переданная строка c имеет k или менее ошибок, то она не может быть другой строкой кода, и ошибка определена.

Коды Хемминга

Доказательство

b) Если строка c передана как c' с k или менее ошибками, то для любой строки $c \in C$ имеем $\delta(c, c') \leq k$.

Если для некоторой строки c'' из C расстояние $\delta(c', c'') \leq k$, то $\delta(c, c') + \delta(c', c'') \leq 2k$.

Но $\delta(c, c'') \leq \delta(c, c') + \delta(c', c'')$ и $\delta(c, c'') \geq 2k + 1$, что приводит к противоречию.

$\Rightarrow c'$ можно исправить на c – единственную строку, расстояние которой от c' меньше, чем $k + 1$. ■

Коды Хемминга

- Возникает проблема определения $D(C)$, наименьшего расстояния между любыми двумя строками из кода C.

Коды Хемминга

Теорема 6 Минимальное расстояние $D(C)$ кода C равно $W(C) = \min\{wt(c): c \in C \text{ и } c \neq \mathbf{0}\}$.

Доказательство

По определению $D(C)$, существуют $w, w' \in C$ такие, $\delta(c, c'') = wt(c + c'')$.

$c + c'' \in C \Rightarrow W(C) \leq D(C)$.

Обратно, для $c \in C$ имеем $wt(c) = wt(c + \mathbf{0}) = \delta(c, \mathbf{0}) \geq D(C)$.

Следовательно, $W(C) \geq D(C)$.

Отсюда $W(C) = D(C)$.■

Коды Хемминга

Задание

a) Построить код Хэмминга.
b) Изучить корректирующие способности кода Хэмминга.
c) Показать, что синдромы указывают позиции ошибок.