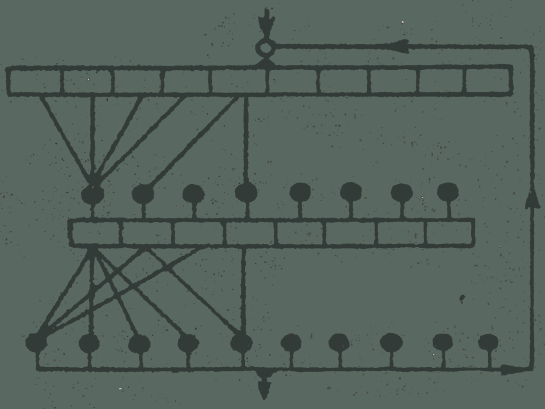
ТЕОРИЯ КОДИРОВАНИЯ



ИЗДАТЕЛЬСТВО
«МИР»

ТЕОРИЯ КОДИРОВАНИЯ

ПЕРЕВОД
С АНГЛИЙСКОГО

ПОД РЕДАКЦИЕЙ
Э. Л. БЛОХА

ИЗДАТЕЛЬСТВО «МИР»
МОСКВА 1964

Предлагаемый читателю сборник ознакомит его с наиболее характерными работами в области математической теории корректирующих кодов, выполненными за последние годы.

В работах широко используется аппарат алгебры и теории вероятностей в применении к теории кодирования.

Книга вызовет интерес у научных работников, инженеров и студентов, работающих в области корректирующих кодов и их технической реализации.

Редакция литературы по математическим наукам

ПРЕДИСЛОВИЕ

Применение теории групп и алгебры полиномов над полем Галуа к задачам кодирования привело в последние годы к бурному развитию математической теории корректирующих кодов.

В результате этого были найдены регулярные методы построения достаточно эффективных кодов, исправляющих независимые ошибки и пакеты ошибок.

Алгебраическая структура получаемых кодов допускает сравнительно простую техническую реализацию процесса кодирования, основанную на применении регистров сдвига.

Основные результаты, полученные в этом направлении к 1961 году, изложены в превосходной монографии Питерсона „Коды, исправляющие ошибки", которая недавно вышла в русском переводе. Три года, прошедшие после выхода в свет книги Питерсона, характерны резким увеличением числа публикаций, в которых еще шире используется аппарат общей алгебры для решения различных задач теории кодирования. Кроме того, значительное внимание уделяется вопросам, связанным с упрощением методики декодирования, практическое осуществление которого является одной из труднейших технических задач, стоящих на пути широкого применения корректирующих кодов.

Целью настоящего сборника является ознакомление читателя с некоторыми наиболее характерными работами последнего времени. Работы Матсона и Соломона, Цетерберга, Неймана посвящены разработке новых методов построения и исследования корректирующих кодов. Первые три статьи относятся к изучению циклических, а последняя — циклически перестановочных кодов.

В работе Элспаса и Шорта изучаются циклические коды с минимальной избыточностью, исправляющие единичные пакеты ошибок, а в работе Стоуна весьма важные в практическом отношении коды, исправляющие многократные пакеты ошибок.

В двух статьях Бергера описываются не групповые, но разделимые коды, особенно эффективные при обнаружении ошибок в асимметричных каналах. Так, в полностью асимметричном канале эти коды обнаруживают любое сочетание ошибок.

В работе Банерджи ставится интересная задача о построении с помощью вычислительной машины групповых кодов, исправляющих ошибки произвольного (но заранее заданного) типа. Статьи Галагера и Фано относятся к весьма перспективному направлению современной теории корректирующих кодов, они посвящены разработке таких методов декодирования, при которых число операций растет по линейному или по степенному (а не по экспоненциальному) закону с увеличением длины кодовых последовательностей. Теоретические выводы, полученные Галагером, иллюстрируются специально проведенным на вычислительной машине экспериментом. Статья Фано является чрезвычайно интересным дополнением к книге Возенкрафта и Рэйффена „Последовательное декодирование“, русский перевод которой вышел в 1963 г. В этой статье, так же как и в упомянутой книге, обсуждается вопрос о применении методов вероятностного декодирования вместо традиционного декодирования, основанного исключительно на использовании алгебраических свойств кода.

Статьи Компопиано и Джонсона посвящены дальнейшему уточнению некоторых границ для оценки зависимости избыточности кода от характера и типа исправляемых ошибок. Наконец, в последней работе Меггита анализируются принципы работы кодирующих и декодирующих схем (для циклических кодов), главная часть которых состоит из регистров сдвига с обратными связями. Характерной особенностью этой работы является последовательное использование матричного описания работы исследуемых устройств.

Можно надеяться, что сборник будет полезен научным работникам, инженерам и аспирантам, работающим в области теории корректирующих кодов и их технического осуществления.

Э. Л. Блох

НОВАЯ ТРАКТОВКА КОДОВ БОУЗА — ЧОУДХУРИ [1])

Х. Матсон и Г. Соломон

Обозначив через A некоторый (n, k)-код Боуза — Чоудхури, мы сначала сопоставляем каждому вектору a в A (с помощью разностных уравнений над $GF(2)$) [2]) полином $g_a(x)$, такой, что, координаты вектора a суть значения полинома $g_a(x)$ на множестве корней n-й степени из единицы. Степень этих полиномов такова, что минимальный ненулевой вес d векторов в A, есть по меньшей мере d_0, т. е. обычная нижняя оценка Боуза — Чоудхури.

Посредством нескольких основных теорем эта нижняя оценка улучшена в классе $(p, h+1)$-кодов, где $p=2h+1$ принимает некоторые простые значения. В частности, довольно просто доказано, что (23, 12)-код Голея имеет $d=7$; показано, что (47, 24)-код имеет $d \geqslant 9$, что, таким образом, улучшает на 4 обычную для этого кода нижнюю оценку $d_0=5$.

1. Введение. В конструировании автоматических высокоскоростных систем связи обычно в качестве „сообщения“ берется упорядоченная последовательность двузначных произвольных символов, скажем, 0, 1. Таким образом, сообщение должно состоять из векторов $a=(a_0, a_1, \ldots, a_{n-1})$, где каждое a_i есть 0 или 1. Так как обычное сложение и умножение по модулю 2 удобно для выполнения на электронной машине, будем рассматривать множество V всех этих последовательностей длины n из 0 и 1 как векторное пространство над $F=GF(2)$, полем из двух элементов со сложением, определенным поразрядно.

В реальных системах при передаче знака a_i могут происходить ошибки. Чтобы справиться с этим явлением, устанавливают, что будут передаваться только те векторы a, которые входят в некоторое подмножество A множества V, и тогда стараются выбрать A таким образом,

---

[1]) Mattson H. F., Solomon G., A new treatment of Bose — Chaudhuri codes, *Journal of society for industrial and applied mathematics*, 9 (1961), № 4, 654—669.

[2]) $GF(q)$ — поле Галуа из q элементов; q — степень простого числа. — *Прим. ред.*

что если передан $a \in A$ и получен $a^* \in V$, то все же возможно восстановить a по a^* при условии, что произошло не слишком много ошибок. Подмножество A обычно называется *кодом* или *кодом, исправляющим ошибки.* Элементы множества A будут называться кодовыми векторами.

В частности, если мы выберем A в качестве (линейного) подпространства V, можно очень просто описать число ошибок в отдельном сообщении, которые оно может исправить. Так, если мы определим *вес* $w(a)$ вектора $a \in V$ как число тех a_i, которые равны 1, то функция $\varrho(a, b) = w(a+b)$ (хэммингово расстояние, т. е. метрика в V) инварианта относительно сдвигов. Поэтому расстояние

$$d(a) = \min\{\varrho(a, b);\ b \in A,\ b \neq a\}$$

одинаково для всех $a \in A$; в частности $d(a) = d(0) = \min\{w(b);\ b \in A,\ b \neq 0\} = d$ как раз и есть это значение. Таким образом, взяв [1]) $e = [(d-1)/2]$, мы легко увидим, что если передан a и принят a^*, то ближайший к a^* элемент кода A есть сам a всякий раз, когда общее число ошибок в a^* не более чем e.

Определение эффективного процесса отыскания элемента кода A, ближайшего к данному элементу множества V — так называемая задача декодирования — представляет собой важную задачу в этой области. Другая важная проблема состоит в определении d для данного (n, k)-*группового кода* A, т. е. k-мерного подпространства пространства V.

Групповые коды, которыми мы здесь ограничимся, принадлежат главным образом Боузу и Рой-Чоудхури [1], которые дали конструктивную процедуру для получения большого класса циклических кодов, имеющих заданные корректирующие свойства. Как можно показать, их коды во многих случаях обладают большей способностью исправлять ошибки, чем это следует из их собственной теории. Эти коды были изучены ранее несколькими авторами [3, 4, 8, 9]. Их оценки свойств

[1]) Квадратные скобки означают, как обычно, целую часть (только здесь).

этих кодов исправлять ошибки основаны на методах, использующих матрицы, линейные рекурсивные последовательности и кольца полиномов.

Мы воспроизвели их оценки в общем случае, а в подклассе кодов Боуза и Чоудхури улучшили эти оценки.

Наши методы основаны на трактовке линейных рекурсий как конечно-разностных уравнений [7] и на освещении предмета исследований особенно простым образом. В частности, мы заменили трудную комбинаторную задачу определения числа единиц в кодовом векторе более легко поддающейся исследованию алгебраической задачей определения числа нулей некоторого полинома на данном конечном множестве.

Эта статья достаточно самостоятельна. Мы предполагаем известными, однако, некоторые основные алгебраические понятия, такие как элементарные свойства полиномов и существование расширения полей.

Тем не менее читатель сможет понять всю статью без особых усилий, даже если он имеет ограниченное знакомство с алгеброй.

Чтобы иллюстрировать основные теоремы статьи, мы в п. 5 подробно разбираем один частный случай ($p = 23$). Для читателя, не интересующегося деталями, этот пример, вероятно, даст хорошее представление о нашей работе.

Резюме. В п. 2 мы приводим ту часть работы [7], которая нам нужна для дальнейшего. Для полноты мы приводим доказательство (кстати совершенно простое).

В п. 3 мы определяем групповой код, полученный из линейной рекурсии, и выводим основной результат (лемма 2), который основывается на п. 2.

Применяя лемму 2 к кодам Боуза—Чоудхури, определяемым так же, как в [8], мы получаем простое доказательство ранее известного результата (теорема 1), дающего нижнюю оценку минимального ненулевого веса кодового вектора.

Остальная часть работы ограничивается некоторым классом $(p, h+1)$-кодов Боуза и Чоудхури, определенных для некоторых простых чисел $p = 2h+1$.

В п. 4 мы приводим главный результат об этих кодах, включающий, очевидно, сильное улучшение (теорема 2)

нижней оценки упомянутого выше минимального веса для (p, h)-подкода с векторами четного веса в случае $p \equiv 1 \pmod 8$.

В теореме 3 мы доказываем, что известная ранее нижняя оценка может быть всегда улучшена на 1, если эта нижняя оценка не является уже совершенно хорошей (именно, если d не равно h).

В теореме 5 мы даем соотношение между p и нечетными весами кодовых векторов.

В п. 5 мы даем частные результаты для $p=23$ (случай Голея [2]) и $p=47$. Наши методы дают краткое и простое доказательство того, что код Голея исправляет три ошибки, и без больших усилий мы увеличиваем от 5 до 9 упомянутую выше нижнюю оценку в случае $p=47$.

2. **Разностные уравнения над *GF*(2).** Пусть, как и всюду далее, $F=GF(2)$ означает поле из двух элементов 0, 1. Пусть $a_0, a_1, \ldots$ — линейная рекурсивная последовательность с элементом из F, в которой a_j определено рекурсией

$$a_{k+i}+b_1a_{k+i-1}+b_2a_{k+i-2}+\ldots+b_ka_i=0 \quad (i=0, 1, 2, \ldots), \quad (1)$$

где $b_1, \ldots, b_k \in F$ не зависят от i и значения $a_0, \ldots, a_{k-1}$ уже заданы.

Уравнение (1) есть просто разностное уравнение с постоянными коэффициентами. Чтобы найти общее решение уравнения (1), положим $a_j=\beta^j$ для всех j в (1), как и в классическом случае; получаем уравнение

$$\beta^i(\beta^k+b_1\beta^{k-1}+\ldots+b_{k-1}\beta+b_k)=0,$$

которое удовлетворяется, если β есть корень полинома $f(x)=x^k+b_1x^{k-1}+\ldots+b_k$. Чтобы получить конечное поле K, которое содержит F и все корни полинома $f(x)$, имеется стандартный алгебраический метод, который, как мы допускаем, может быть применен без особых затруднений.

Мы ограничим наше внимание случаем, когда все корни $\beta_1, \beta_2, \ldots, \beta_k$ различны. Пусть для любых $c_1, c_2, \ldots, c_k \in K$

$$a_j=c_1\beta_1^j+\ldots+c_k\beta_k^j \quad (j=0, 1, 2, \ldots). \quad (2)$$

Тогда эта последовательность $\{a_j\}$ удовлетворяет уравнению (1); она может лишь не принимать заданных зна-

чений для $j=0, 1, \ldots, k-1$. Но имеется единственное множество величин c в K, полученное в качестве решения системы k линейных уравнений, имеющей матрицу Вандермонда коэффициентов такую, что последовательность (2) удовлетворяет уравнению (1) при всех $j=0, 1, 2, \ldots$.

Таким образом, каждая линейная рекурсивная последовательность $\{a_i\}$, определенная для (1), получается в форме (2). Обратно, каждая последовательность вида (2) есть линейная рекурсия (1) над K. Ее элементы принадлежат F при условии, что $a_0, a_1, \ldots, a_{k-1}$ также принадлежат F.

Эти результаты мы сформулируем в виде леммы.

Лемма 1. *Если* $a_0, a_1, \ldots$ *есть последовательность элементов из* F, *данная рекурсией* (1), *и если полином*

$$f(x)=x^k+b_1x^{k-1}+\ldots+b_k$$

не имеет кратных корней, то существуют единственным образом определенные элементы $c_1, \ldots, c_k$ *из* K, *такие, что*

$$a_i=c_1\beta_1^i+\ldots+c_k\beta_k^i \quad (i=0, 1, 2, \ldots),$$

где $\beta_1, \ldots, \beta_k$ — *корни полинома* $f(x)$.

Детальное изложение этого предмета с дополнительными результатами о линейных рекурсивных последовательностях, не используемых здесь, имеется в [7].

3. **Коды.** Пусть n — нечетное целое, и пусть $f(x)=x^k+b_1x^{k-1}+\ldots+b_k$ — полином с коэффициентами из F (множество всех таких полиномов обозначается символом $F[x]$); предположим также, что $f(x)$ делит x^n+1. Пусть K означает наименьшее поле, содержащее F и все корни полинома x^n+1.

Пусть V — векторное пространство всех упорядоченных последовательностей длины n $a=(a_0, a_1, \ldots, a_{n-1})$ со сложением векторов a и $b=(b_0, b_1, \ldots, b_{n-1})$, определяемым правилом $a+b=(a_0+b_0, a_1+b_1, \ldots, a_{n-1}+b_{n-1})$. Мы определяем некоторый (n, k)-групповой

код A как следующее подпространство пространства V: [1])

$$A=\{a;\ a=(a_0,\ a_1,\ \ldots,\ a_{n-1})\in V,\ a_{i+k}+b_1a_{i+k-1}+\ldots+ \\ +b_ka_i=0,\ i=0,\ 1,\ \ldots,\ n-k-1\}. \quad (3)$$

Таким образом, чтобы построить вектор кода A, мы выбираем координаты $a_0,\ \ldots,\ a_{k-1}$ произвольно из F и определяем последующие координаты рекурсией в (3). A есть циклический код размерности k над [2]) F. Критерий цикличности кода состоит в том, что если $a(x)$ означает полином $a_0x^{n-1}+a_1x^{n-2}+\ldots+a_{n-1}$ из $F[x]$ и если $f^*(x)=(x^n+1)/f(x)$, то вектор a из V принадлежит циклическому коду A тогда и только тогда, когда соответствующий полином $a(x)$ кратен полиному $f^*(x)$. Этот результат доказан в [5] для всех n; мы по существу доказываем его в следствии 2 нашими методами для нечетных n.

Вспомним, что *вес* $w(a)$ вектора $a=(a_0, a_1, \ldots, a_{n-1})\in V$ есть общее число a_i, равных единице. Всюду далее мы будем использовать обозначение d для минимального ненулевого веса векторов a из A:

$$d=\min\{w(a);\ a\neq(0,0,\ldots,0),\ a\in A\}.$$

Дадим определение кодов Боуза—Чоудхури, имеющееся в [8]. Пусть β—любой примитивный корень n-й степени из единицы над F; выберем $s<n$ и пусть $f^*(x)$—полином над F наименьшей степени, который делит x^n+1 и содержит $\beta,\ \beta^2,\ \ldots,\ \beta^s$ среди своих корней. Пусть $f(x)=(x^n+1)/f^*(x)$.

Определение. *Код Боуза—Чоудхури для* β *и* s *определяется как код* A, *связанный с* $f(x)$ *посредством нашего определения* (3).

Известно, что для определенного выше кода A мы имеем $d\geqslant d_0=s+1$. Мы докажем позже этот результат (теорема 1) нашими методами.

Если $\varphi(x)$—любой полином над F и z—любой корень этого полинома, то и z^2 есть также его корень. Этот

[1]) Читатель может заметить несущественное различие между нашим определением и определением некоторых авторов, которые „обращают“ код, т. е. в качестве нашего $f(x)$ используют $x^kf(1/x)$.

[2]) См. для определенности следствие 1 (III).

факт следует из того свойства, что если L любое поле, содержащее F, то $(\alpha+\beta)^2=\alpha^2+\beta^2$ для всех α, $\beta\in L$, что и означает $\varphi^2(z)=\varphi(z^2)$. Если $\varphi(x)$ неприводим над F, то все корни полинома $\varphi(x)$ получаются повторным возведением в квадрат любого из них; если $\varphi(z)=0$, то все корни полинома $\varphi(x)$ суть $z, z^2, z^4, \ldots, z^{2^{j-1}}$, где j есть степень полинома $\varphi(x)$. Так же, $z^{2^j}=z$. Попутно заметим, что выбор упомянутого выше s четным вполне естествен; в самом деле, если s нечетно, то β^{s+1} будет автоматически корнем полинома $f^*(x)$, так как $\beta^{(s+1)/2}$ корень по определению. Поэтому мы всегда в приведенном выше определении будем брать s четным.

Установим некоторые определения, основные в этой статье. Пусть n нечетно, и пусть, как прежде, $f(x)\in F[x]$ делит x^n+1. Пусть ξ—примитивный корень n-й степени из единицы, т. е. элемент $\xi\in K$, такой, что $\xi^n=1$ и ни одна меньшая степень ξ не равна 1. Положим

$$E(\xi)=\{e,\ 0\leqslant e\leqslant n;\ f(\xi^e)=0\}. \tag{4}$$

Другими словами, если $f(x)$ имеет степень k, то $E(\xi)$ состоит из целых чисел $e_1, e_2, \ldots, e_k$; $0\leqslant e_i\leqslant n$, таких, что $\xi^{e_1}, \xi^{e_2}, \ldots, \xi^{e_k}$ все являются корнями полинома $f(x)$. Например, если $n=7$, мы имеем

$$x^7+1=(x+1)(x^3+x+1)(x^3+x^2+1)=(x+1)f_0(x)f_1(x).$$

Предположим, что мы берем $f(x)=(x+1)f_0(x)$. Так как $n=7$ простое число, любой корень полинома x^7+1, отличный от 1, есть примитивный корень 7-й степени из 1. Если мы берем ξ корнем $f_0(x)$, то $E(\xi)=\{0, 1, 2, 4\}$. Но если мы берем ξ корнем $f_1(x)$, то так как корнями полинома $f_1(x)$ будут ξ, ξ^2, ξ^4, то корни полинома $f_0(x)$ теперь суть ξ^3, ξ^5, ξ^6. Таким образом, $E(\xi)=\{0, 3, 5, 6\}$.

Основным для всей работы является следующий результат. Как и выше, пусть n нечетное, пусть ξ—примитивный корень n-й степени из 1, пусть $f(x)$ делит x^n+1, и пусть A—код, связанный с $f(x)$ посредством (3). Тогда имеет место следующая лемма.

Лемма 2. *Для каждого* $a=(a_0, a_1, \ldots, a_{n-1})\in A$ *имеется полином* $g_a(x)$ *с коэффициентами из* K, *такой, что* $a_i=g_a(\xi^i)$ *для* $i=0, 1, \ldots, n-1$. *Если* ξ *фиксировано, этот полином единственным образом определяется*

вектором a. *Степень полинома* $g_a(x)$ *не превосходит* m, *где* m — *наибольшее целое в* $E(\xi)$.

Доказательство. Корни полинома $f(x)$ есть $\xi^{e_1}, \ldots, \xi^{e_k}$, $e_i \in E(\xi)$ и $f(x)$ не имеет кратных корней. Мы можем поэтому применить лемму 1 к нашему вектору $a=(a_0, a_1, \ldots, a_{n-1}) \in A$, так как рекурсия (3), определяющая вектор $a \in A$, является рекурсией типа (1). Мы получаем $c_1, c_2, \ldots, c_k \in K$, где все c_j зависят, конечно, от a и такие, что

$$a_i = c_1(\xi^{e_1})^i + \ldots + c_k(\xi^{e_k})^i \qquad (5)$$

для $i=0, 1, \ldots, n-1$. Но мы можем написать (5) в виде

$$a_i = c_1(\xi^i)^{e_1} + \ldots + c_k(\xi^i)^{e_k} \quad (i=0,1, \ldots, n-1) \qquad (6)$$

и это замечание приводит нас к определению полинома

$$g_a(x) = c_1 x^{e_1} + \ldots + c_k x^{e_k},$$

который, по лемме 1, единственным образом определяется вектором $a \in A$ (и выбором ξ). Уравнение (6) есть утверждение того, что $a_i = g_a(\xi^i)$.

Отображение $a \longrightarrow g_a(x)$ кода A в $K[x]$ является линейным над F, т. е. если $a, b \in A$, то

$$g_{a+b}(x) = g_a(x) + g_b(x).$$

Мы подчеркиваем, что $g_a(x)$ должен принимать значения 0 или 1 на группе корней n-й степени из единицы $Z=\{1, \xi, \ldots, \xi^{n-1}\}$. Вес вектора a поэтому есть n без числа корней полинома $g_a(x)$ на Z. Таким образом, комбинаторная задача определения веса вектора преобразуется в более алгебраизированную задачу определения числа корней полинома на данном множестве. В частности, в дальнейшем мы воспроизведем нижнюю оценку Боуза — Чоудхури d_0 для d, просто доказав, что мы можем выбрать ξ так, что степень каждого ненулевого $g_a(x)$ есть не более чем $n-d_0$ (число нулей $g_a(x)$ на Z тогда не более $n-d_0$, так что вес вектора a есть по меньшей мере $n-(n-d_0)=d_0$.)

Используя величину d_0, введенную Боузом и Чоудхури, с помощью [8] мы докажем теперь, что минимальный ненулевой вес d кода A есть по меньшей мере d_0.

Мы слегка перефразируем определение числа d_0, используя $f(x)$ вместо s. Как и прежде, пусть n нечетно и пусть $f(x) \in F[x]$ делит x^n+1, причем имеется примитивный корень β n-й степени из 1, который не есть корень полинома $f(x)$. Пусть A есть код Боуза—Чоудхури, связанный с $f(x)$ посредством (3).

Тогда имеет место теорема.

Теорема 1. *Пусть* β^{d_0}*—наименьшая положительная степень* β, *которая является корнем полинома* $f(x)$. *Тогда* d_0*—необходимо нечетно и* $d \geqslant d_0$.

Доказательство. Как отмечено раньше, достаточно доказать, что для некоторого первообразного корня ξ n-й степени из единицы множество $E(\xi)$ имеет $n-d_0$ своим максимальным членом. Нам было дано, что $\beta, \beta^2, \ldots, \beta^{d_0-1}$ не являются корнями полинома $f(x)$ и что β^{d_0} есть корень полинома $f(x)$. Отсюда немедленно следует, что $E(\xi)$ для $\xi=\beta^{-1}$ не содержит чисел $n-1$, $n-2, \ldots, n-(d_0-1)$, но содержит $n-d_0$.

4. Случай, когда n — простое число некоторого вида; главные результаты. В остальной части статьи мы ограничим наше внимание случаем, когда n есть нечетное простое число $p=2h+1$, такое, что $(x^p+1)=(x+1)f_0(x)f_1(x)$ и $f_0(x)$, $f_1(x)$ неприводимы над F[1]).

Сразу рассмотрим следующие три простых следствия.

(I) x^p+1 распадается на множители над F, как указано выше, тогда и только тогда, когда 2 имеет мультипликативный порядок h по модулю p. В самом деле, если n_i—степень полинома $f_i(x)$ $(i=0,1)$ и если ξ—корень полинома $f_i(x)$, то должно выполняться соотношение $\xi^{2^{n_i}}=\xi$, или $2^{n_i} \equiv 1 \pmod p$. Ни одна меньшая степень числа 2 не может удовлетворить этому сравнению, так как в противном случае мы не нашли бы всех корней полинома $f_i(x)$. Поэтому $n_0=n_1$ и, так как $n_0+n_1=2h$, имеет место равенство $n_0=n_1=h$. Обратно, если 2 имеет мультипликативный порядок h по

[1]) Мы благодарны Пранджу за совет исследовать коды Боуза—Чоудхури, связанные с такими простыми числами p.

$\bmod p$, то x^p+1 распадается на три неприводимых множителя $x+1$, $f_0(x)$ и $f_1(x)$. Таким образом, в этом случае мы можем (что мы и делаем) в качестве K взять $GF(2^h)$.

(II) Мы можем взять

$$f_0(x)=x^h+0\cdot x^{h-1}+\ldots+1 \quad \text{и} \quad f_1(x)=x^h+x^{h-1}+\ldots+1,$$

так как каждый $f_i(x)$ имеет степень h и $f_0(x)f_1(x)=x^{p-1}+x^{p-2}+\ldots+1$; последнее означает, что коэффициенты при x^{h-1} в $f_0(x)$ и $f_1(x)$ не могут быть одинаковыми.

(III) Степени числа 2 есть в точности квадратичные вычеты по $\bmod p$ и $p\equiv\pm 1 \pmod 8$. Действительно, циклическая группа имеет не более одной подгруппы данного порядка и подгруппа квадратичных вычетов по $\bmod p$ также имеет порядок h. Так как, в частности, 2 есть квадратичный вычет по $\bmod p$, закон квадратичной взаимности[1]) указывает, что $p\equiv\pm 1 \pmod 8$[2]). Мы будем от случая к случаю изменять нашу трактовку, чтобы иметь дело с одной или с другой из этих двух возможностей. Два примера таких простых чисел представляют собой $p=7$ и $p=17$. Обозначим через R множество наименьших положительных квадратичных вычетов по $\bmod p$:

$$R=\{r_i;\ 0<r_i<p,\ r_i\equiv 2^{i-1} \pmod p);\ i=1,\ldots,h\}.$$

Пусть $R'=\{s_1,\ldots,s_h\}$ означает множество наименьших положительных квадратичных невычетов по $\bmod p$; $R\cup R'=\{1,2,\ldots,p-1\}$. Пусть s_1 означает наименьший член в R' (заметим, что, таким образом, s_1 нечетно) и выберем обозначения, так, что $s_2\equiv 2s_1$; $s_3\equiv 2s_2,\ldots,s_1\equiv 2s_h \pmod p$, что мы вполне можем сделать, так как $R'\equiv s_1R \pmod p$. Таким образом мы выбрали $r_1=1$; как мы скоро увидим, самый выгодный выбор ξ ведет к $d_0=s_1$.

---

[1]) См. любую книгу по теории чисел, например [10], стр. 127.

[2]) Как второстепенный момент, который может представлять некоторый интерес, заметим, что для простых чисел, меньших чем 3 000 000, числа, сравнимые с $+1$ по $\bmod 8$, встречаются сравнительно реже, чем числа, сравнимые с -1 по $\bmod 8$. См. Shanks D., Quadratic residues and the distribution of primes, *Math. Tables Aids Comput.*, **13** (1959), 272—284 (*Math. Rev.*, **21** (1960), Rev. № 7186, p. 1325).

Мы определим наш код A как $(p, h+1)$-код, связанный посредством (3) с $f(x)=(x+1)f_0(x)$. В этом пункте мы дадим некоторые основные результаты о минимальном ненулевом весе d вектора a в A, и в следующем пункте мы дадим дальнейшие результаты относительно d для частных значений p. Код, связанный посредством (3) с $(x+1)f_1(x)$, эквивалентен коду A, как мы докажем ниже, после следствия 2; поэтому мы ограничим наше внимание кодом A. (Два кода называются *эквивалентными*, если один может быть получен из другого перестановкой координат.) Мы сейчас выполним некоторые из процедур предыдущего пункта для этого кода A. Различие между двумя случаями $p\equiv\pm 1 \pmod 8$ возникает из того, что для q нечетных и простых -1 есть квадратичный вычет по $\bmod q$ тогда и только тогда, когда $q\equiv 1 \pmod 4$. Другими словами,

$$p-1\in R, \text{ если } p\equiv 1 \pmod 8;$$
$$p-1\in R', \text{ если } p\equiv -1 \pmod 8.$$

Пусть ξ есть некоторый примитивный корень p-й степени из 1. Любой выбор ξ дает в качестве множества $E(\xi)$ либо $\{0\}\cup R$, либо $\{0\}\cup R'$; мы всегда будем выбирать ξ так, что $p-1$ не принадлежит $E(\xi)$. То есть выбираем ξ корнем полинома $f_1(x)$, если $p\equiv 1 \pmod 8$, выбираем ξ корнем полинома $f_0(x)$, если $p\equiv -1 \pmod 8$. (Напомним, что в (4) множество $E(\xi)$ определяется так, что $(\xi^e+1)f_0(\xi^e)=0$, где e пробегает все множество $E(\xi)$.) В теореме 1 d_0 определяется как наименьший нечетный показатель такой, что β^{d_0} есть корень полинома $f_0(x)$, где β — корень полинома $f_1(x)$. Другими словами, так как корни полинома $f_1(x)$ есть $\beta, \beta^2, \ldots, \beta^{2^{h-1}}$, т. е. $\beta^{r_1}, \beta^{r_2}, \ldots, \beta^{r_h}$, мы немедленно находим, что $d_0=s_1$. Таким образом, имеет место следующая теорема.

Теорема 1′. *При ограничениях этого пункта и предположении, что s_1 — наименьший квадратичный невычет по $\bmod p$ (s_1 необходимо нечетно), имеет место соотношение $d\geqslant s_1=d_0$.*

Подведем итог предыдущим замечаниям.

Лемма 3. *Выбор примитивного корня ξ p-й степени из единицы и структура полинома $g_a(x)$ леммы 2 таковы, что если $a \in A$, то*

$$g_a(x) = \begin{cases} c_0 + c_1 x^{r_1} + \ldots + c_h x^{r_h}, & (f_0(\xi) = 0),\ p \equiv -1 \pmod 8, \\ c_0 + c_1 x^{s_1} + \ldots + c_h x^{s_h}, & (f_1(\xi) = 0),\ p \equiv 1 \pmod 8. \end{cases}$$

В обоих случаях степень полинома $g_a(x)$ не превосходит

$$p - d_0 = p - s_1.$$

Для удобства нам понадобится множество показателей полинома $g_a(x)$ в общем виде. Будем говорить, что $g_a(x) = c_0 + c_1 x^{e_1} + \ldots + c_h x^{e_h}$, где $e_i = r_i$ для всех $i = 1, 2, \ldots, h$, когда $p \equiv -1 \pmod 8$ и $e_i = s_i$ для всех i, когда $p \equiv +1 \pmod 8$. При нашем выборе обозначений мы имеем $e_i \equiv 2^{i-1} e_1 \pmod p$ для всех i; корни полинома $f_0(x)$ суть $\xi^{e_1}, \ldots, \xi^{e_h}$. Заметим, что e_h не обязательно есть степень полинома $g_a(x)$ (даже когда $c_h \neq 0$). Исследуем теперь коэффициенты c_i полинома $g_a(x)$. Для этого мы прибегаем к следующей лемме, которой мы обязаны Риду [6].

Лемма 4. *Пусть K_0 — любое поле, содержащее F, и $h(x) = \sum c'_j x^j$ — любой полином над K_0. Пусть β — примитивный корень m-й степени из единицы в K_0, m — нечетное и степень полинома $h(x)$ меньше m. Тогда имеет место формула*

$$c'_j = \sum_{i=0}^{m-1} h(\beta^i)\, \beta^{-ji} \quad (j = 0, 1, 2 \ldots).$$

Доказательство. Сумма, приведенная выше, равна

$$\sum_{i=0}^{m-1} \sum_{k \neq j} c'_k \beta^{i(k-j)} + c'_j \sum_{i=0}^{m-1} 1;$$

в ней второе слагаемое есть просто c'_j, так как m нечетно. Остальная часть равна нулю, так как

$$\sum_{i=0}^{m-1} \beta^{i(k-j)} = \frac{x^m + 1}{x + 1}$$

и $x = \beta^{k-j} \neq 1$.

Применяя лемму 4 к нашему полиному и вспоминая лемму 2, мы получаем следующую лемму.

Лемма 5. *Пусть* $a \in A$. *Коэффициенты* $c_0, c_1, \ldots, c_h$ *полинома* $g_a(x)$ *выражаются формулами*

$$c_0 = \sum_{i=0}^{p-1} a_i \qquad (c_0 \in F),$$

$$c_j = \sum_{i=0}^{p-1} a_i \xi^{-ie_j} \quad (c_j \in K;\ j = 1, 2, \ldots, h).$$

В частности $c_0^2 = c_0$; $c_1^2 = c_2$; $c_2^2 = c_3$; $\ldots$; $c_h^2 = c_1$.

Следствие 1.

(I) *Линейное отображение* A *в* $F \times K$, *которое задается соответствием* $a \longrightarrow (c_0, c_1)$ (*где* $a \in A$ *и* $g_a(x) = c_0 + c_1 x^{e_1} + \ldots + c_h^{e_h}$), *является взаимно однозначным (ввиду одинаковой размерности над* F).

(II) *Более того, полином* $g_a(x)$ *удовлетворяет равенству*

$$g_a(\xi^i) = c_0 + T(c_1 \xi^{ie_1}), \qquad a \in A,$$

где T *обозначает способ перехода от* K *к* F (*если* $z \in K$, *то* $T(z)$ *определяется как* $z + z^2 + \ldots + z^{2^{h-1}}$).

(III) *Код* A *циклический, т. е. для каждого* $a = (a_0, a_1, \ldots, a_{p-1})$ *в* A *вектор* $a' = (a_1, a_2, \ldots, a_{p-1}, a_0)$ *содержится также в* A (*для* $a_i' = g_a(\xi^{i+1})$). *Таким образом,* $g_a(x\xi)$ *—полином для кодового вектора, соответствующего* $(c_0, \xi c_1)$ *при отображении, описанном в* (I). (Мы могли бы доказать этот результат, конечно, раньше.)

(IV) *Полином* $g_a(x)$, $a \in A$, *имеет степень* $p - d_0 = p - s_1$, *если только он не является константой (так как* $c_1, c_2, \ldots, c_h$ *либо все нули, либо все ненули).*

Следствие 2. *Пусть* β *—корень полинома* $f_1(x)$; *если* $a \in A$, *то*

$$\sum_{i=0}^{p-1} a_i \beta^{-i} = 0.$$

Доказательство. Так как $\beta = \xi^e$ для некоторого $e \not\equiv 0, e_1, e_2, \ldots, e_h \pmod p$, написанная величина есть 0 как коэффициент при x^e в $g_a(x)$, если $a \in A$. Пусть A_1 есть код, связанный по определению (3) с $f(x) = (x+1) f_1(x)$.

Покажем, что A и A_1 эквивалентны. Пусть S — квадратичный невычет по mod p; рассмотрим перестановку координат, сопоставляющую S_i символу i для всякого $i=0; 1, \ldots, p-1 \pmod p$. Если ξ выбрано в соответствии с леммой 3, то $a \in A$ отображается этой перестановкой в $b=(g_a(1), g_a(\beta), \ldots, g_a(\beta^{p-1}))$, где $\beta=\xi^S$. То есть, $b_i=g_a(\xi^{S_i})=a_{S_i}$. По следствию 1 (I), множество полиномов $g_a(x)$, $a \in A$, совпадает с соответствующим множеством полиномов для кода A_1, и по лемме 3 β удовлетворяет надлежащему выбору корня p-й степени из единицы для A_1, так как β и ξ не сопряжены. В случае когда $p \equiv -1 \pmod 8$, мы можем выбрать $S=-1$ и просто получить A_1 как „обратный“ („противоположный“) коду A, получающийся одним циклическим сдвигом.

Докажем теперь простой результат о четных весах кода A, когда $p \equiv 1 \pmod 8$. То есть мы исследуем величину d для кода, принадлежащего полиному $f(x)=f_0(x)$, в соответствии с определением (3). Результат аналогичен теореме 1 в том смысле, что рассуждения сосредоточиваются целиком на степени полинома $g_a(x)$.

Теорема 2. *Пусть d' означает минимальный ненулевой четный вес, достигаемый на A. Если $p \equiv 1 \pmod 8$, то $d' \geqslant 2d_0$.*

Доказательство. Пусть $a \in A$ и $w(a)$ четно. По леммам 3 и 5

$$g_a(x)=x^{d_0}(c_1+\ldots+c_h x^{s_h-d_0}),$$

и полином в круглых скобках имеет степень $p-2d_0$. Поэтому $w(a) \geqslant 2d_0$.

Имеет место следующая теорема.

Теорема 3. *Если $d_0 < h$, то $d \geqslant d_0+1$.*

Доказательство. Из теоремы 1 мы знаем, что $d \geqslant d_0$, и если $d=d_0$, то для некоторого $a \in A$ все корни полинома $g_a(x)$ являются корнями p-й степени из единицы. Так как d_0 нечетно, то $g_a(0)=c_0=1$; поэтому, если c означает старший коэффициент полинома $g_a(x)$, то постоянный член $1/c$ (полинома $(1/c)\,g_a(x)$) есть корень p-й степени из единицы. Таким образом, $c^p=1$ и по лемме 5 все коэффициенты c_j полинома $g_a(x)$ удовлетворяют равенству $c_j^p=1$. Таким образом, для всех i имеет

место равенство $a_i = g_a(\xi^i) = 1 + T(c\xi^{e_1 i})$; здесь $c_1\xi^{e_1 i}$ пробегают все значения корней p-й степени из единицы, а i изменяется от 0 до $p-1$. Таким образом, a_i принимает значение 0 h раз (когда $c_1\xi^{e_1 i}$ является корнем $f_0(x)$) и значение 1 h раз, а также значение $1+T(1) = 1+h\cdot 1$. Таким образом, если h нечетно, то a имеет вес h. Это противоречит тому, что $w(a) = d_0 < h$. Если h четно, то a имеет вес $h+1$, что также приводит к противоречию.

Заметим, что ничего нельзя достичь, разбивая условие теоремы на два: „$d_0 < h$ (h нечетно)“ и „$d_0 < h+1$ (h четно)“, так как d_0 нечетно. Заметим также, что мы доказали, что для этого класса простых чисел нижняя оценка Боуза—Чоудхури не является наилучшей из возможных, исключая, быть может, случай, когда $d_0 = h$, например при $p = 7$ (что действительно может иметь место лишь в этом случае).

Теорема 4. *Если* $d_0 < h$ *и* $p \equiv 1 \pmod 8$, *то* $d \geqslant d_0 + 2$.

Доказательство. Из теоремы 3 мы знаем, что $d \geqslant d_0 + 1$, и если $d = d_0 + 1$ является четным числом, то из теоремы 2 следует, что $d' = d_0 + 1 \geqslant 2d_0$, или $d_0 \leqslant 1$. Но, вообще говоря, $d_0 \geqslant 3$. Следовательно, $d > d_0 + 1$.

С помощью следующего следствия можно показать, что $d \geqslant d_0 + 2$, когда $p \equiv -1 \pmod 8$.

Следствие 3. *Пусть* $p \equiv -1 \pmod 8$, *и* $c_{j+1} = c_1^{2^j}$ *— старший коэффициент полинома* $g_a(x)$. *Если* $2^j - 1$ *взаимно просто с* $2^h - 1$ *и если* $d_0 < h$, *то* $d \geqslant d_0 + 2$.

Доказательство. Из теоремы 3 мы знаем, что $d \geqslant d_0 + 1$. Если $a \in A$ имеет вес $d_0 + 1$, то $c_0 = 0$ по лемме 5. Таким образом, $(1/x)\, g_a(x) = c_1 + \ldots + c_h x^{r_h - 1}$, и этот полином имеет степень $p - d_0 - 1$ по теореме 1. Таким образом, из равенства $w(a) = d_0 + 1$ следует, что все корни полинома $(1/x)\, g_a(x)$ являются корнями p-й степени из единицы. Но $c_{j+1}/c_1 = c_1^{2^j - 1}$ по предположению имеет тот же порядок, что и c_1; поэтому c_1 есть корень p-й степени из единицы. Это значит, что a имеет вес $h+1$ (так как $T(1) = 1$), как в доказательстве теоремы 3.

Соответствующий результат для случая $p \equiv 1 \pmod 8$ не имеет смысла, так как здесь $j = h/2$.

Наш следующий результат касается свойства цикличности кода A. Как и прежде, пусть вектор $a = (a_0, \ldots, a_{p-1})$ соответствует полиному $a(x) = a_0 x^{p-1} + \ldots + a_{p-1}$. Тогда циклический сдвиг $(a_1, a_2, \ldots a_{p-1}, a_0)$ вектора a соответствует полиному $xa(x)$, приведенному по $\bmod (x^p + 1)$. Если $a \in A$, то $a(x)$ кратно $f_1(x)$ и обратно, как мы заметили в п. 3. Отождествляя векторы $a \in V$ с соответствующими им полиномами $a(x)$, получим следующую теорему.

Теорема 5. *Пусть $a \in A$ имеет нечетный вес m. Тогда*

$$m^2 - m + 1 \geqslant p, \quad \text{если } p \equiv -1 \pmod 8$$

и

$$m^2 \geqslant p, \quad \text{если } p \equiv +1 \pmod 8.$$

Доказательство. Пусть A_1 — код, связанный с $(x+1) f_1(x)$. Пусть $a \in A$ имеет нечетный вес m. Так как A и A_1 эквивалентны, то должен существовать вектор $b \in A_1$ веса m. Рассмотрим теперь полином $a(x)\, b(x)$. Так как b имеет вес m, $a(x)\, b(x)$ есть сумма m циклических сдвигов вектора a, и поэтому $a(x)\, b(x)$ — вектор нечетного веса в A. По той же причине он и из A_1. Поэтому $a(x)\, b(x)$ кратен обоим полиномам $f_0(x)$ и $f_1(x)$ и отсюда — их произведению $f_0(x) f_1(x) = x^{p-1} + x^{p-2} + \ldots + 1$. Так как вес нечетный, $a(x)\, b(x) \equiv x^{p-1} + x^{p-2} + \ldots + 1 \pmod{x^p + 1}$. Но в полиноме $a(x)\, b(x)$ имеется не более m^2 членов, поэтому $m^2 \geqslant p$.

Чтобы усовершенствовать эту оценку в случае $p \equiv -1 \pmod 8$, мы используем специальную перестановку координат $i \to -i \pmod p$, которая, как известно, дает в этом случае эквивалентность между A и A_1. Тогда для каждой координаты $a_i = 1$ в a, член x^{p-1-i} полинома $a(x)$ подбирается в пару с членом x^{p-1+i} полинома $b(x)$; из m^2 членов произведения $a(x)\, b(x)$ имеется m одинаковых, именно $x^{p-1-i} x^{p-1+i} = x^{2p-2} (\equiv x^{p-2} \pmod{x^p + 1})$. Таким образом, m единиц сводятся к единственной единице, так что $p \leqslant m^2 - (m-1)$.

5. Частные значения p (47 и 23). Мы применим наш метод к (23,12)-коду Голея [2], который известен как код, исправляющий три ошибки, т. е. $d=\min\{w(a);\ a\neq 0,\ a\in A\}=7$ [5, стр. 7]. Разложим $x^{23}+1=(x+1)f_0(x)f_1(x)$ на неприводимые множители [1]), где

$$f_0(x)=x^{11}+x^9+x^7+x^6+x^5+x+1$$

и

$$f_1(x)=x^{11}+x^{10}+x^6+x^5+x^4+x^2+1.$$

Если β есть корень полинома $f_0(x)$, то его же корнями будут

$$\beta,\ \beta^2,\ \beta^4,\ \beta^8,\ \beta^{16},\ \beta^9,\ \beta^{18},\ \beta^{13},\ \beta^3,\ \beta^6,\ \beta^{12},$$

в то время как корнями полинома f_1 будут

$$\beta^5,\ \beta^{10},\ \beta^{20},\ \beta^{17},\ \beta^{11},\ \beta^{22},\ \beta^{21},\ \beta^{19},\ \beta^{15},\ \beta^7,\ \beta^{14}.$$

Код A есть множество всех рекурсивных последовательностей с элементами из F (длины 23), порожденных разностными уравнениями, связанными с $f_0(x)(x+1)$, как в п. 2. Для $a=(a_0, a_1, \ldots, a_{11}, a_{12}, \ldots, a_{22})$ общий член a_k находится по лемме 1 в виде

$$a_k=\sum_{i=0}^{11} c_i(\beta^{r_i k});\quad r_i\equiv 2^{i-1}\pmod{23};\quad 0<r_i<23 \qquad (9)$$

(и $r_0=0$), когда c_i $(i=0, 1\ldots, 11)$ определяется первыми 12 значениями $(a_0, a_1, \ldots, a_{11})$. Коэффициенты c_i содержатся в $K=GF(2^{11})$, наименьшем поле над F, содержащем корни 23-й степени из единицы.

Код A имеет размерность 12 в $V_{23}(F)$ и, конечно, является циклическим.

Мы можем рассматривать (9) как значения полинома $g_a(x)$, когда x пробегает все корни 23-й степени из единицы, именно

$$g_a(x)=\sum_{i=0}^{11} c_i x^{r_i}.$$

Тогда

$$a_k=\sum_{i=0}^{11} c_i(\beta^{r_i k})=\sum_{i=0}^{11} c_i(\beta^k)^{r_i},$$

[1]) Таблица неприводимых множителей полинома x^n+1 над F для нечетных $n\leqslant 35$ имеется в [5] стр. 22—23.

или

$$a_k = g_a(\beta^k).$$

Коэффициенты c_i—посредством леммы 5—легко получаются с помощью равенств

$$\begin{aligned} c_i &= c_1^{2^{i-1}} \quad (i = 1, \ldots, 11), \\ c_0 &= \sum_{i=0}^{22} a_i. \end{aligned} \tag{10}$$

Показатели r_i в (9) суть

1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12.

Мы замечаем, что каждому кодовому слову a ставится в соответствие пара (c_0, c_1). Коэффициент c_0 равен 0 или 1 в соответствии с весом вектора a, четным или нечетным; c_1 есть элемент поля $K = GF(2^{11})$ и находится по формуле

$$c_1 = \sum_{i=0} a_i \beta^{-i},$$

где β—фиксированный корень полинома $f_0(x)$, выбранный ранее.

Для каждого $c \in K$ мы имеем два кодовых слова: одно четного и одно нечетного веса. Таким образом, мы имеем естественное отображение нашего кода A на $F \times K$, как в следствии 1.

Заметим, что x^{18} есть член наибольшей степени в нашем полиноме, так как $r_i = 18$ и $g_a(x)$, очевидно, не может иметь более 18 корней. Поэтому минимальный вес a в A есть минимальное количество единиц, которые полином $g_a(x)$ принимает в качестве своих значений на множестве корней 23-й степени из 1. Мы немедленно получаем нижнюю оценку Боуза—Чоудхури, именно $23 - 18 = 5$.

Мы можем это сделать, однако, лучше, исследуя коэффициенты полинома $g_a(x)$. Если $w(a) = 5$, то $c_0 = 1$ и наш полином есть (пусть $c = c_1$ в (10))

$$g_a(x) = c^{2^6} x^{18} + c^{2^4} x^{16} + c^{2^7} x^{13} + \ldots + 1.$$

Произведение корней полинома $g_a(x)$ есть, конечно, $1/c^{2^6}$.

Если все 18 корней полинома $g_a(x)$ есть корни 23-й степени из единицы, то их произведение есть также корень 23-й степени из единицы: $(c^{2^6})^{23}=1$, поэтому $(c^{2^6})^{2^5}=c$—корень 23-й степени из единицы. В этом случае каждый показатель над c может быть приведен по mod 23 и мы имеем

$$g_a(x)=c^{18}x^{18}+c^{16}x^{16}+c^{13}x^{13}+\ldots+1.$$

Если мы положим $y=cx$, то получим

$$g_a(x)=y^{18}+y^{16}+y^{13}+y^{12}+\ldots+y^2+y+1.$$

Это есть полином над F, который если имеет корень β, то должен иметь и все с ним сопряженные (получаемые возведением в квадрат). Значит, $g_a(x)$ может иметь только 12 корней из единицы в качестве своих нулей, т. е. $w(a)$ есть по меньшей мере 11. Поэтому рассмотрим только те c, которые не являются корнями 23-й степени из единицы. Мы заключаем, что $w(a)>5$.

Теперь мы сразу исключим из рассмотрения $w(a)=6$. В самом деле, если $w(a)=6$, то $c_0=0$ по (10) и

$$g_a(x)=cx+c^2x^2+\ldots+c^{24}x^{16}+c^{26}x^{18}=$$
$$=x(c+c^2x+\ldots+c^{26}x^{17}).$$

Произведение 17 ненулевых корней есть, конечно,

$$c/c^{26}=(c^{26-1})^{-1}.$$

Если все эти 17 корней есть корни 23-й степени из единицы, то

$$(c^{26-1})^{23}=1.$$

Но 2^6-1 взаимно просто с $2^{11}-1$, так что c само должно быть корнем 23-й степени из 1 и из этого снова следует, что $w(a)$ может принимать значения вплоть до 11.

Таким образом мы получили, что $w(a)\geqslant 7$. Однако мы знаем о существовании вектора a с весом $w(a)=7$ (см. [5]), что и завершает исследование.

С другой стороны, мы могли бы использовать наши общие результаты, чтобы доказать, что $d\geqslant 7$ для этого кода следующим образом.

Мы имеем $d_0=s_1=5$ и $h=11$; $c_1^{2^6}$—старший коэффициент полинома $g_a(x)$. 2^6-1 взаимно просто с $2^{11}-1$, поэтому $d\geqslant 7=d_0+2$, по следствию 3.

Следовало, быть может, отметить, что (23,12)-код A по нашему определению не есть в точности код Голея, но получен из него соответствующей перестановкой координат.

Некоторые результаты для $p=47$. В этом примере мы полагаемся на основные результаты п. 4. Мы снова имеем $d_0=5$, так как

$$R=\{1, 2, 4, 8, 16, 32, 17, 34, 21, 42, 37, 27, 7, 14, 28, 9, 18, 36, 25, 3, 6, 12, 24\}.$$

Здесь $g_a(x)$ имеет степень $r_{10}=42$, так что старший коэффициент $c_{10}=c_1^{29}$. Теперь 2^9-1 взаимно просто с $2^{23}-1$, так как $(2^{23}-1)-2^5(2^9+1)(2^9-1)=31$; таким образом, наибольший делитель $(2^{23}-1, 2^9-1)$ есть или 1 или 31. Он не может быть равным 31, так как $2^5\equiv 1 \pmod{31}$, в то время как $2^9\equiv 2^4\not\equiv 1 \pmod{31}$. Таким образом, по следствию 3, $d\geqslant 7$.

Теорема 5 исключает векторы веса 7. Векторы веса 8 мы будем исключать с помощью следующего процесса (который пригоден также для векторов веса 7): пусть $a\in A$ имеет вес 8. Тогда $g_a(x)$ имеет в точности 39 нулей $\beta_1, \ldots \ldots \beta_{39}$ на множестве $Z=\{1, \xi, \xi^2, \ldots \xi^{46}\}$. Таким образом, мы разлагаем $g_a(x)$ (в поле, быть может, превосходящем поле K) в виде

$$g_a(x)=c_{10}(x)(x+\gamma_1)(x+\gamma_2)(x+\beta_1)\ldots(x+\beta_{39}),$$

где γ_1 и γ_2 отличны от 0 и не являются корнями 47-й степени из единицы.

Пусть $\alpha_1, \ldots, \alpha_8$—корни 47-й степени из единицы, отличные от всех β_i; таким образом,

$$(x+\alpha_1)\ldots(x+\alpha_8)(x+\beta_1)\ldots(x+\beta_{39})=x^{47}+1.$$

Наш первый вывод состоит в том, что $\gamma_1=\gamma_2=\gamma$, так как $\gamma_1+\gamma_2+\beta_1+\ldots+\beta_{39}=0$ есть коэффициент при x^{41} в $g_a(x)$ и $\alpha_1+\alpha_2+\ldots+\alpha_8+\beta_1+\ldots+\beta_{39}=0$. Таким образом, $\gamma_1+\gamma_2=\alpha_1+\ldots+\alpha_8$. Теперь каждое $\alpha_i=\xi^h$ и координаты с номерами j_i вектора a—как раз и есть все его единичные позиции. Так как $p\equiv -1 \pmod 8$, мы выбираем ξ корнем полинома $f_0(x)$; таким образом, ξ^{-1}—корень β полинома $f_1(x)$.

Применим следствие 2, чтобы заключить, что

$$\alpha_1+\alpha_2+\ldots+\alpha_8=0\,(=\beta_1+\ldots+\beta_{39}).$$

Теперь определим

$$\sum_{i=0}^{39} b_i x^{39-i}=(x+\beta_1)\ldots(x+\beta_{39}).$$

Тогда $b_0=1$ и $b_1=\beta_1+\ldots+\beta_{39}=0$. Теперь имеем

$$g_a(x)=c_{10}x(x^2+\gamma^2)(x^{39}+b_2x^{37}+b_3x^{36}+\ldots+b_{39}).$$

Отсюда немедленно следует, что коэффициент x^{40-i} в $g_a(x)$ есть

$$c_{10}(\gamma^2 b_i+b_{i+2}) \quad (i=0,\ 1,\ \ldots,\ 37). \tag{11}$$

Более того,

$$c_{10}\gamma^2\beta_1\ldots\beta_{39}=c_1$$

или

$$(\gamma^2 c_{10}/c_1)^{47}=1, \tag{12}$$

что попутно доказывает, что γ^2, и отсюда все сопряженные с ним элементы, включая γ, содержатся в K. Окончательно определим полином

$$\sum_{i=0}^{8} d_i x^{8-i}=(x+\alpha_1)\ldots(x+\alpha_8),$$

в котором $d_0=1$ и $d_1=0$.

Выведем теперь различные соотношения между величинами b_i, d_i и γ, которые вместе с (12) приведут к противоречию. Мы сначала используем (11) и наш список для R. Так как 40, 39, $38\notin R$, мы имеем

$$\begin{aligned} b_2&=\gamma^2,\\ b_3&=b_1=0,\\ b_4&=\gamma^4. \end{aligned}$$

Продолжая этот процесс и замечая, что $37=r_{11}$ и $36=r_{18}$ и т. д., мы получаем

$$\begin{aligned} b_5&=c_{11}/c_{10}=c_{10},\\ b_6&=c_{18}/c_{10}+\gamma^6,\\ b_7&=\gamma^2 c_{10}. \end{aligned} \tag{13}$$

Имеет место соотношение

$$(x^8+d_2x^6+d_3x^5+\ldots+d_8)(x^{39}+b_2x^{37}+b_4x^{35}+\ldots+b_{39})=$$
$$=x^{47}+1. \tag{14}$$

Исследуем коэффициенты при x^{46}, x^{45}, ..., полученные перемножением в левой части (14), сначала получая $b_2+d_2=0$ как коэффициент при x^{45}; таким образом, $b_2=d_2=\gamma^2$ из (13).

Выражая следующие несколько коэффициентов и используя $d_2=\gamma^2$ и (11), мы находим

$$\begin{aligned}
&d_3=0,\\
&0=b_4+b_2d_2+d_4=d_4,\\
&0=b_5+b_3d_2+d_5=c_{10}+d_5,\\
&0=b_6+b_4d_2+b_2d_4+d_6=c_{18}/c_{10}+d_6,\\
&0=b_7+b_5d_2+b_2d_5+d_7=\gamma^2d_5+d_7,\\
&0=b_8+b_6d_2+b_2d_6+d_8=c_8/c_{10}+\gamma^2d_6+d_8.
\end{aligned} \tag{15}$$

После того как мы определили все d_i в (15), мы ищем противоречия в предыдущих равенствах. Мы выписываем коэффициент при x^{35} в левой части (14), именно

$$0=b_{12}+b_{10}d_2+b_7d_5+b_6d_6+b_5d_7+b_4d_8=$$
$$=c_{10}(b_7+d_7)+\gamma^4c_8/c_{10}+\gamma^6d_6+b_6d_6, \tag{16}$$

где используем (13) и (15), которые также дают $b_7=d_7$ и т. д., так что (16) принимает вид $\gamma^4c_8/c_{10}+(c_{18}/c_{10})^2=0$. По лемме 5 это уравнение эквивалентно уравнению

$$\gamma^2c_7/c_9=c_{18}/c_{10}. \tag{17}$$

Умножим теперь (17) на c_9c_{10}/c_1c_7, чтобы получить $\gamma^2c_{10}/c_1=c_9c_{18}/c_1c_7$, что по (12) должно быть корнем 47-й степени из единицы. Используя снова лемму 5, пишем

$$c_9c_{18}/c_1c_7=c_1^{2^{17}+2^8-2^6-1}.$$

Теперь докажем, что $c_1^{47}=1$, показывая для этого, что $2^{17}+2^8-2^6-1$ взаимно просто с $2^{23}-1$. Пусть δ означает наибольший общий делитель этих двух чисел. Тогда δ делит

$$2^6(2^{17}+2^8-2^6-1)-(2^{23}-1)=3(2^{12}-21),$$

и поэтому δ делит

$$3 \cdot 2\,(2^{23}-1)-(2^{12}+21)\,3\,(2^{12}-21)=3 \cdot 439.$$

Теперь, если q есть простой делитель числа δ, то $2^{23} \equiv 1 \pmod{q}$, так что $q \equiv 1 \pmod{46}$. Таким образом, $3 \nmid \delta$[1]) и так как 439 — простое число, нам остается только заметить, что $439 \equiv -21 \pmod{46}$, так что $439 \nmid \delta$ также. Поэтому $\delta = 1$, что дает $c_1^{47} = 1$, или $w(a) = h = 23$ (согласно доказательству теоремы 3), и в этом заключается противоречие. Таким образом, для этого кода $d \geqslant 9$.

Так как Прандж нашел в A много векторов веса 11, то для этого (47, 24)-кода новым результатом будет $9 \leqslant d \leqslant 11$.

ЛИТЕРАТУРА

1. Bose R. C., Ray-Chaudhuri D. K., On a class of error correcting binary group codes, *Inf. and Control*, 3 (1960), 68—79. [Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 2, ИЛ, 1961, 83—94.]
2. Golay J. E., Notes on digital coding, *Proc. I. R. E.*, 37 (1949), 657.
3. Gorenstein D., Zierler N., A class of cyclic linear error-correcting codes in p^m symbols, Group Report 55-19, Lincoln Laboratory (1960); A class of error-correcting codes in p^m symbols, *J. Soc. Indust. Appl. Math.*, 9 (1961), 207—214.
4. Peterson W. W. Error-correcting codes, M. I. T. Press and J. Wiley ans Sons, Inc., New York, N. Y., 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]
5. Prange E., Cyclic error-correcting codes in two symbols, Report № AFCRC-TN-57-103, USAF Cambridge Research Laboratory, Bedford, Mass., 1957.
6. Reed I., Solomon G., A decoding procedure for a polynomial code, Group Report 47-24, Lincoln Laboratory, 1959.
7. Solomon G., Linear recursive sequences as finite difference equations, Group Report 4737, Lincoln Laboratory, 1960.
8. Weiss E., Some connections between linear recursive sequences and error-correcting codes: informal lectures, Gr. Report 55-22, Lincoln Laboratory, 1960.
9. Residue class rings and linear recursive sequences, Group Report 55-24, Lincon Laboratory, 1960.
10. Weyl H., Algebraic Theory of Numbers, Ann. Math. Studies № 1, Princeton Univ. Press, 1940, [Русский перевод: Вейль Г., Алгебраическая теория чисел, ИЛ, М., 1947.]

[1]) $m \nmid \delta$ означает, что δ не делит m. — *Прим. ред.*

ЗАМЕТКА О НОВОМ КЛАССЕ КОДОВ [1])

Г. Соломон

В данной статье в качестве чисто алгебраической задачи рассматривается построение корректирующих групповых (p, k)-кодов (p нечетно), т. е. линейных отображений последовательностей нулей и единиц длины k в последовательности нулей и единиц длины p. Эта задача связана с нахождением нулей некоторых полиномов на множестве корней p-й степени из единицы. Упомянутые полиномы характеризуются с помощью элементов подгрупп наименьших полей, содержащих корни p-й степени из единицы. Кроме того, мы вводим так называемые коды, порождаемые регистрами скачкообразных сдвигов. Такими являются $[p, (p+1)/2]$-коды, исправляющие одну ошибку, где p — простое число, такое, что 2 имеет мультипликативный порядок $p-1$. Эти нециклические коды могут быть названы псевдоциклическими. Кодирование и декодирование осуществляются достаточно легко.

I. Введение

В своей трактовке кодов Боуза—Чоудхури Маттсон и Соломон [1] ввели новый взгляд на все циклические коды. Каждому кодовому слову нечетной длины p ставится в соответствие некоторый полином. Компоненты кодового вектора суть значения, принимаемые этим полиномом на множестве корней p-й степени из единицы. Корректирующие свойства кодов интерпретируются задачей отыскания нулей полиномов на множестве корней p-й степени из единицы. Такой подход дал некоторые новые интересные результаты для циклических кодов и, в частности, для кодов Боуза—Чоудхури.

Мы применяем новый подход ко всем групповым кодам и пересматриваем задачу в этом контексте. Мы устанавливаем таким образом изоморфизм между всеми групповыми кодами и подгруппами прямого произведения

---

[1]) Solomon G., A note on a new class of codes, *Inf. and Control*, 4 (1961), № 4, 364—370.

некоторых полей и, используя этот подход, получаем новый класс кодов — псевдоциклические коды. Это коды, которые получаются с помощью регистров сдвига, а длина кодовой комбинации $p \equiv \pm 3 \pmod 8$ такова, что 2 имеет мультипликативный порядок[1]) $p-1$. Они аналогичны некоторым $[p, (p+1)/2]$-кодам Боуза — Чоудхури, например, (23, 12), (17, 9) и (47, 24)[2]).

II. Общее представление групповых кодов

Рассмотрим векторное пространство $V_p(F)$ размерности p (p — нечетно) над полем F из двух элементов 0 и 1. Каждому $a=(a_0, a_1, \ldots, a_{p-1}) \in V_p(F)$ мы ставим в соответствие полином $g_a(x)$ степени, не большей чем $p-1$, такой, что $g_a(\beta^i)=a_i$, где β есть первообразный корень p-й степени из 1. (Для $a=(0, 0, \ldots, 0)$ положим $g_a(x)=0$.) В предположении, что

$$g_a(x)=\sum_{i=0}^{p-1} c_i x^i,$$

условие $g_a(x)=0$ или 1 для $x=\beta^i$ $(i=0, 1, \ldots, p-1)$ приводит к тому, что $g_a^2(x)=g_a(x)$. При $x=\beta^i$

$$\sum_{i=0}^{p-1} c_i^2 x^{2i}=\sum_{i=0}^{p-1} c_i x^i.$$

Имеем таким образом $\sum_{i=0}^{p-1} (c_i^2+c_{2i}) x^{2i}=0$ при $x=\beta^i$, где все степени x приводятся по модулю p. Это ведет к таким условиям, накладываемым на c_i:

$$c_0^2=c_0, \quad c_i^2=c_{2i}, \quad i=1, 2, \ldots, p-1.\ ^3)$$

---

[1]) Иными словами, число 2 есть первообразный корень по модулю p. — *Прим. перев.*

[2]) Для которых $p \equiv \pm 1 \pmod 8$. — *Прим. ред.*

[3]) Вместо поля F мы можем выбрать любое конечное поле, скажем $L=GF(q^m)$ (поле Галуа из q^m элементов), и поставить условие, что p и q взаимно просты. Таким образом, мы получим из уравнения $g_a(x)^{q^m}=g_a(x)$ условия, накладываемые на c_i, и установим аналогичное представление для $V_p(L)$.

Заметим, что полином $g_a(x)$ содержит очень мало независимых констант. Таковыми являются $c_0 \in F$ и $c_{l_1}, c_{l_2}, \ldots, c_{l_{r-1}}$, где c_{l_1} не сопряжено с c_{l_j} при $l_i \neq l_j$, т. е. $c_{l_j} \neq c_{l_i}^{2^s}$ для любого S.

Мы можем действительно выразить c_i в явной форме через компоненты a_i и корни p-й степени из 1. Используя полученный ранее результат [2], имеем [1]

$$c_k = \sum_{i=0}^{p-1} a_i (\beta^i)^{-k}.$$

Видим, что

$$c_0 = \sum_{i=0}^{p-1} a_i, \quad c_1 = \sum_{i=0}^{p-1} a_i \beta^{-i}, \quad c_2 = c_1^2, \quad c_4 = c_1^4, \ldots .$$

Ясно, что c_i содержится в наименьшем поле K над F, содержащем корни p-й степени из единицы.

Каждому $a \in V_p(F)$ поставим в соответствие множество элементов поля K вида $(c_0, c_1, c_{i_1}, c_{i_2}, \ldots, c_{i_{r-1}})$, где $c_0 = 0$ или 1 в зависимости от четности общего числа единиц в векторе a; c_1 есть коэффициент при x; c_{i_1} есть коэффициент при x^{i_1}, где i_1 есть наименьшее целое такое, что $i_1 \not\equiv 2^s \pmod p$ ни при каком s; i_2 есть наименьшее целое, большее чем i_1, такое, что ни при каком s $i_2 \not\equiv 2^s \pmod p$ или $i_2 \not\equiv i_1 2^s \pmod p$ и т. д. Ясно, что соответствие $a \longrightarrow (c_0, c_1, c_{i_1}, \ldots, c_{i_{r-1}})$ есть аддитивный изоморфизм (зависящий от начального выбора корня β) между $V_p(F)$ и подгруппой прямого произведения поля F с r экземплярами поля K ($F \times K^r$). Если p — простое

---

[1] Выдержка из доказательства: пусть $g(x) = \sum_{i=0}^{m} c_i x^i$, $m < p$; тогда $c_k = \sum g(x) x^{-k}$, где суммирование ведется по всем корням p-й степени из 1. Развертывая $g(x)$ в формуле для c_k, т. е. записывая $\sum [\sum c_i x^i] x^{-k}$, меняя порядок суммирования и используя тот факт, что $x^p + 1 = (x+1)(\sum_{i=0}^{p-1} x^i) = 0$ (p простое нечетное), получим требуемое.

число, то $V_p(F)$ изоморфно в точности прямому произведению $F \times K^r$.

Если φ есть отображение пространства $V_k(F)$ в пространстве $V_p(F)$, то, очевидно, имеется подгруппа этого прямого произведения, которая соответствует $\varphi(V_k)$. Элементы пространства $\varphi(V_k)$ есть значения, принимаемые полиномом $g_a(x)$ на множестве корней p-й степени из единицы. Полином $g_a(x)$ можно представить в следующем виде:

$$g_a(x) = c_0 + \sum_{j=0}^{o(2)-1} (c_1 x)^{2^j} + \sum_{k=1}^{r-1} \sum_{j=0}^{o(i_k)-1} (c_{i_k} x^{i_k})^{2^j},$$

где $o(i)$ есть наименьшее целое m_i, такое, что $i 2^{m_i} \equiv 1 \pmod p$.

Некоторые значения r суть $r(7) = 2$, $r(13) = r(11) = 1$, $r(31) = 6$. Коды Боуза—Чоудхури получаются при $r(n) \geqslant 2$. Они естественно получаются, если положить некоторые c_i равными нулю. Существует разностное уравнение и полином, ассоциированный с этим отображением, и эти коды порождаются регистром сдвига [3].

Заметим попутно, что r зависит от мультипликативного порядка δ числа 2 по модулю p, т. е. δ есть наименьшее целое, такое, что $2^\delta \equiv 1 \pmod p$. Это определяет число сопряженных первообразных корней p-й степени из единицы и отсюда — степень полинома (и (или) разностного уравнения), который порождает код. Число r есть попросту число неприводимых над F множителей полинома $(x^p + 1)/(x + 1)$. Полное обсуждение этого вопроса, включающее детальный анализ тех простых чисел p, для которых $r = 2$, можно найти в [1].

Для некоторых $p \equiv \pm 3 \pmod 8$ имеет место равенство $r_2(p) = 1$. Это значит, что неприводимыми множителями полинома $x^p + 1$ являются $(x + 1)$ и $(x^{p-1} + x^{p-2} + \ldots + 1)$. Для этих p мы строим новый класс кодов. Некоторые значения таких p суть 11, 13, 19, 29 и т. д. Не все $p \equiv \pm 3 \pmod 8$ обладают этим свойством; например, при $p = 43$ число 2 имеет мультипликативный порядок 14, $r(p) = 3$, при $p = 157$ число 2 имеет порядок 52, $r(p) = 3$.

III. Корректирующие свойства

Групповой (p, k)-код исправляет (или обнаруживает) [1] t ошибок, если $t = \left[\frac{d}{2}\right]$ [2]), d — нечетно (d — четно), где d есть минимальный вес всех ненулевых векторов пространства $\varphi(V_k) \subset V_p$; d равно наименьшему числу единиц вектора $a \neq 0$ в пространстве $\varphi(V_k)$. Ясно, что d равно минимальному числу единиц, принимаемых в качестве своих значений полиномом $g_a(x)$, когда $a \neq 0$ пробегает все пространство $\varphi(V_k)$. Иными словами, $p-d$ есть максимальное число нулей полинома $g_a(x)$ (по всем a), когда x пробегает все множество корней p-й степени из единицы. Корректирующие свойства любого кода определяются алгебраическими свойствами полиномов этого специального вида: какое максимальное число нулей может иметь полином $g_a(x)$ на множестве корней p-й степени из единицы?

Интересным фактом является то, что все групповые коды могут рассматриваться с этой более алгебраической точки зрения. Очевидно, определение нулей полиномов может оказаться непростой задачей, однако мы можем допустить, что коэффициенты пробегают некоторую подгруппу поля K и вычислить значения полинома $g_a(x)$ для таких c даже вручную.

IV. Новый класс кодов

A. Коды, порожденные регистром скачкообразного сдвига

Рассмотрим теперь некоторый класс кодов. Это нециклические (p, k)-коды для простых p вида $8l \pm 3$ и $r(p) = 1$. Циклическими кодами для таких p являются просто $(p, 1)$ и $(p, p-1)$-коды с известными корректирующими свойствами. Они порождаются разностными уравнениями,

---

[1]) При четном d код обнаруживает t ошибок и исправляет $t-1$ ошибку. — *Прим. ред.*

[2]) $\left[\frac{d}{2}\right]$ означает наибольшее целое, не превосходящее $\frac{d}{2}$.

связанными с полиномами $x+1$ или $x^{p-1}+x^{p-2}+\ldots+1$. Если бы мы исследовали (p, k)-коды при $k\neq 1$, $k\neq p-1$, то мы должны были бы обратиться к старым методам, их порождающим. Мы должны были бы найти линейные преобразования T, которые отображают $V_k(F)$ в $V_p(F)$.

Мы образуем новый код, все еще использующий схему регистра сдвига, которая порождает кодовые последовательности, но мы не сможем производить сдвиг обычным путем; мы будем производить в регистре скачкообразный сдвиг [1]) и полученные таким образом значения используем иначе, чем ранее. Мы собираемся отыскать новый класс кодов, которые исправляют одну ошибку. Наши результаты таковы.

Существует класс кодов длины p, исправляющих по меньшей мере одну ошибку и содержащих $(p+1)/2$ информационных символов. Эти коды порождаются простым способом с использованием схемы регистра сдвига для кодовых слов длины $p-1$.

B. Коды

Пусть p — простое число типа $r(p)=1$, например 11, 13. Тогда для β, первообразного корня p-й степени из единицы, и любого $a=(a_0, a_1, \ldots, a_{p-1})\in V_p(F)$ мы построим полином [2])

$$g_a(x)=c_0+cx+c^2x^2+c^4x^4+\ldots+c^{2^{(p-1)/2}}x^{p-1}+$$
$$+c^{2^{(p+1)/2}}x^{p-2}+\ldots+c^{2^{(p-2)}}x^{(p+1)/2},$$

такой, что $g_a(\beta^i)=a_i\in F$. Как отмечено выше, $c_0=\sum_{i=0}^{p-1}a_i$ есть вес по модулю 2 вектора a и $c=\sum_{i=0}^{p-1}a_i\beta^{-i}$.

Заметим, что для $c=0$ мы получим вектор, состоящий либо из одних нулей, либо из одних единиц. Для $c=1$,

---

[1]) Сдвиг более чем на одну позицию. — *Прим. ред.*

[2]) В соответствии с (1) $g_a(x)=c_0+\sum_{j=0}^{p-1}(cx)^{2^j}$ и степени x приведены по mod p. — *Прим. ред.*

$c_0 = 1$ мы получим полином $g_a(x) = \sum_{i=0}^{p-1} x_i$, который обращается в 0 при $x = \beta^i$, $i = 1, \ldots, p-1$. Следовательно, для таких пар (1,1) мы получили вектор веса 1. Для $c = \beta^k$ и $c_0 = 1$ нулями полинома $g_a(x)$ являются $x = \beta^i$, $i = 0, \ldots, p-1$, $i \neq k$. Таким образом, когда c пробегает все степени β^i, мы получаем все возможные векторы веса 1.

Чтобы обеспечивалось исправление одной ошибки, полином $g_a(x)$ должен иметь не более чем $p-3$ нуля на множестве корней p-й степени из единицы. Выберем множество коэффициентов c так, чтобы это было выполнено. Расположим полином $g_a(x)$ по убывающим степеням x:

$$g_a(x) = c^{2^{(p-1)/2}} x^{p-1} + c^{2^{(p+1)/2}} x^{p-2} + \ldots + cx + c_0.$$

Для исключения векторов веса 1 поставим условие $c^p \neq 1$, т. е. чтобы c не было корнем p-й степени из единицы. Мы немедленно обнаруживаем, что для того, чтобы полином $g_a(x)$ имел $p-2$ нуля на множестве корней p-й степени из единицы, должно выполняться условие $c_0 = 0$, так как в этом случае вес кодового вектора равен 2. Произведение ненулевых корней полинома $g_a(x)$ дается выражением $c/c^{2^{(p-1)/2}} = c^{-(2^{(p-1)/2}-1)}$. Так как по предположению все они суть корни p-й степени из единицы, то их произведение должно удовлетворять условию $c^{(2^{(p-1)/2}-1)p} = 1$. Таким образом, если мы выберем подгруппу C поля $GF(2^{p-1})$ элементов c,

$$c \in C \ni c^{(2^{(p-1)/2}-1)p} \neq 1,$$

то код (c_0, c), $c_0 \in F$, $c \in C$, будет исправлять одну ошибку и иметь размерность на 1 большую, чем размерность группы C. Теперь мы выберем группу C для $p = 8n \pm 3$ так, чтобы ее размерность была равна $(p-1)/2$.

Поле K, которое содержит корни p-й степени из единицы, для этого p имеет вид $GF(2^{p-1})$, т. е. $p-1$ есть степень наименьшего неприводимого полинома $f(x) = x^{p-1} + x^{p-2} + \ldots + 1$, который имеет β в качестве своего корня. Поле $K = GF(2^{p-1})$, как мы сейчас увидим,

имеет подполя́. Для $p=8n+3$ поле $GF(2^{p-1})$ содержит подполя́ степени 2 и $4n+1$ над F.

Пусть α — порождающий элемент мультипликативной группы K^* поля $GF(2^{p-1})=K$, т. е. $K^*=\{a^i\}$, $i=0, \ldots$ $\ldots, 2^{p-1}-1$. Заметим теперь, что, если $\beta=\alpha^{2^{4n+1}+1}$, то степени β образуют мультипликативную группу порядка $2^{4n+1}-1$ и поэтому содержатся в подпо́ле $GF(2^{4n+1})$ поля K, имеющем степень $4n+1$ над F. Выберем $\gamma=$ $=\alpha^{(2^{p-1}-1)/3}$ являющееся, таким образом, кубическим корнем из единицы и содержащееся поэтому в поле $GF(2^2)$. Множество $C=\gamma GF(2^{4n+1})$ есть, таким образом, аддитивная подгруппа поля $GF(2^{p-1})$ и обладает тем очевидным свойством, что

$$C \ni c^{\left(2^{(p-1)/2}-1\right)p} \neq 1.$$

Если $p=8n+5$, мы замечаем, что поле $GF(2^{8n+4})$ имеет подполя́ степени 2, 4, $2n+1$, $4n+2$ и, поступая как в предыдущем случае, выбираем $\gamma=\sqrt[5]{1}$

$$\gamma=\alpha^{(2^{p-1}-1)/5}$$

и полагаем $C=\gamma GF(2^{4n+2})$. Кодовые слова, связанные с парами (c_0, c), $c_0 \in F$, $c \in C$, дадут $[p, (p+1)/2]$-код, который исправляет по меньшей мере одну ошибку.

Проведем параллель с $[p, (p+1)/2]$-кодами Боуза — Чоудхури, где $r(p)=2$. Имеется взаимно однозначное соответствие между каждым словом из p символов и парой (c_0, c), где $c_0 \in F$ и $c \in GF(2^{(p-1)/2})$. Эти коды исправляют минимум одну ошибку, но обычно значительно больше. (23,12)-код Голея исправляет 3 ошибки и (47, 24)-код — минимум три ошибки. Корректирующие свойства этих циклических кодов, по-видимому, зависят как от величины числа p, так и от принадлежности его к некоторому классу вычетов по модулю 8. По-видимому, эти псевдоциклические $[p, (p+1)/2]$-коды могут исправлять более одной ошибки. Исчерпывающие корректирующие свойства должны быть еще исследованы [1]).

[1]) Это исследование сопряжено с серьезными теоретико-числовыми трудностями. Для $p=11$ вручную, а для $p=13$ и 19 на вычислительной машине „Минск-1“ удалось установить, что псевдоциклические коды исправляют только одну ошибку. — *Прим. перев.*

Для кодов, исправляющих n ошибок, мы должны установить более глубокий критерий, которому удовлетворяет подгруппа C. Установив достаточный критерий, мы можем действительно образовать необходимую подгруппу и найти метод кодирования.

C. Кодирование

Рассмотрим один из возможных методов кодирования. Мы хотим кодировать векторы $(g_c(\beta^i))$ при

$$g_c(x) = c_0 + cx + c^2x^2 + \ldots + c^{2^{(p-1)/2}}x^{p-1},$$

где

$$c \in \{\gamma GF(2^{(p-1)/2}),$$

$$\gamma^3 = 1 \text{ для } p \equiv 3 \pmod 8;$$

$$\gamma^5 = 1 \text{ для } p \equiv -3 \pmod 8,\ c_0 \in F\}.$$

Пусть α есть последовательность нулей и единиц длины $p-1$, которая порождает пространство $V_{p-1}(F)$ с помощью процесса кодирования Боуза—Чоудхури, реализуемого регистром сдвига. Построим матрицу $B = \|b_{ij}\|$, $b_{ij} = (\beta^i)^{2^j-1}$, $i = 0, 1, \ldots, p-1$, $j > 0$, где $\beta = \alpha^{2^{(p-1)/2}+1}$, $b_{i0} = 1$. Степени β получаются сдвигом регистра $2^{(p-1)/2}+1$ раз.

Мы производим, так сказать, скачкообразный сдвиг регистра. Кодовые слова $a = (a_0, a_1, \ldots, a_{p-1})$ длины p очень просто получаются применением матрицы B к вектору-столбцу $c = [c_0, c, c^2, \ldots, c^{2^{p-1}}]$. Эти операции могут быть выполнены схемой регистра сдвига применительно к длине $p-1$. Bc есть вектор-столбец длины p, компоненты которого суть нули и единицы.

Информационные символы, которые должны быть переданы, т. е. $(x_0, x_1, \ldots, x_{(p-1)/2})$ соответствуют символам $c_0 = x_0$, $c = \gamma \sum_{i=1}^{(p-1)/2} x_i\delta_i \in GF(2^{(p-1)/2})$, где δ_i есть базис поля $GF(2^{(p-1)/2})$. Мы переводим последовательность $(x_0, x_1, \ldots, x_{(p-1)/2})$ в c_0 и c, фиксируя c_0 и порождая c регистром сдвига. Мы можем просто передать $x_1, \ldots, x_{(p-1)/2}$

как те степени элементов δ, которые соответствуют двоичному числу, представленному последовательностью $(x_1, \ldots, x_{(p-1)/2})$, используя для δ первообразные элементы поля $GF(2^{(p-1)/2})$.

D. Декодирование

Если при передаче произошла одна ошибка, мы получим $(g_{c'}(\beta^i))$, где $c' = c + \beta^k$, что указывает на искажение символа, стоящего на k-м месте. Чтобы вычислить c по $c' = \sum_{i=0}^{p-1} a_i' \beta^{-i}$, нужно просто сдвинуть регистром ту степень β, которая, будучи прибавленной к c', дает элемент группы C. Он будет единственным. Мы также прибавляем 1 к вычисленному значению $c_0' = \sum_{i=0}^{p-1} a_i'$, так как при передаче произошла одна ошибка. Техника декодирования здесь идентична той, которая используется в кодах Боуза—Чоудхури [2].

Этот класс кодов, исправляющих одну ошибку, служит интересным примером некоторых возможных выгод, которые может принести умножение на векторном пространстве над F. Для p, таких, что $r(p) \geqslant 2$, мы можем использовать коды Боуза—Чоудхури или псевдоциклическую их модификацию.

ЛИТЕРАТУРА

1. Mattson H. F., Solomon G., A new treatment of Bose—Chaudhuri codes, *J. Soc Ind. Appl. Math.*, **9** (1961), № 4, 654—669. [Русский перевод: Матсон Х., Соломон Г., Новая трактовка кодов Боуза—Чоудхури, см. настоящий сборник, стр. 7—29.]
2. Peterson W. W., Error-correcting codes, M.I.T and Wiley, New York (1961). [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]
3. Reed I. S., Solomon G., Decoding procedure for a polynomial code, Group report 47-24, Lincoln Laboratory, 1959.
4. Solomon G., Linear recursive sequences as finite differences equations, Group report 47-37, Lincoln Laboratory, 1960.

ЦИКЛИЧЕСКИЕ КОДЫ, ИСПРАВЛЯЮЩИЕ КРАТНЫЕ ОШИБКИ И ПОСТРОЕННЫЕ С ПОМОЩЬЮ НЕПРИВОДИМЫХ ПОЛИНОМОВ [1])

Л. Х. Цеттерберг

В статье изучается класс кодов, исправляющих кратные ошибки и которые строятся с помощью регистров сдвига. Производящая матрица удовлетворяет уравнению, в левой части которого стоит неприводимый полином с коэффициентами 0 и 1. Ошибки классифицируются в терминах циклов ошибок и предлагается метод определения того, являются ли циклы ошибок различными. Чтобы облегчить анализ, вводится классификация ошибок по числу искаженных символов (вес).

Исследуются некоторые частные коды либо для исправления пакетов ошибок, либо для исправления кратных ошибок некоторых весов. Рассматриваются пакеты длины 2, 3, 4, 5 и 6 главным образом с помощью таких кодов, которые пока возможны только теоретически. Найдены коды для исправления пакетов длины 2, 3 и 4. Степень полинома изменяется от 6 до 18, что дает широкие пределы, в которых могут изменяться длины кодов. В случае единичной и двойной смежной ошибки дано достаточное условие существования кода теоретически максимальной длины. В деталях изучается исправление всех единичных и двойных ошибок в кодах длины 2^p+1. В качестве специального случая исправления тройной ошибки анализируется также код Голея длины 23. В заключение эти и другие известные примеры кодов, выполнимых с помощью регистра сдвига, сравниваются с результатом исчерпывающего исследования подходящих полиномов степени 8, 9, 10 и 11.

I. Введение

В недавнем прошлом большое число кодов было построено с использованием техники регистров сдвига. [2]) При использовании этих кодов могут быть исправлены не только

---

[1]) Zetterberg L. H., Cyclic codes from irreducibe polynomials for correction of multiple errors, *IRE Transactions on Information Theory*, **IT—8** (1962) № 1, 13—20.

[2]) Для удобства читателя обращение к литературным источникам, указанным в библиографии к данной статье, может быть с успехом заменено обращением к книге Питерсона „Коды, исправляющие ошибки", ИЛ, М., 1964.— *Прим. перев.*

единичные и смежные двойные ошибки, но также и более длинные пакеты ошибок [1, 2]. Эта техника применима также к исправлению всех единичных и двойных ошибок [3] и вообще ко всем ошибкам с весом меньшим или равным некоторому числу [4]. Свойства регистра сдвига описываются посредством матрицы T с циклическим свойством $T^n = I$, где n—длина кода и $T^q \neq I$, если $q < n$. Эта матрица удовлетворяет полиномиальному уравнению, которое является характеристическим. Во всех упомянутых случаях коды строятся с помощью полиномов, являющихся произведением некоторых неприводимых множителей.

В статье будет показано, что возможно построение кодов, которые могут исправлять кратные ошибки, хотя эти коды порождаются неприводимыми полиномами.

Посредством подходящего выбора полинома исправляются либо пакеты ошибок, либо кратные ошибки некоторого веса. Алгебраические свойства этих кодов изучаются средствами теории полей Галуа [5]. Все построенные коды могут быть реализованы методом Абрамсона [1] и Меггита [6, 7]. Благодаря матрицам, которые связаны с полиномами, последние представляются в форме регистров сдвига. Декодирующее устройство будет использовать детектор набора ошибок по одному набору для каждого цикла ошибок. В то же время в статье отмечено, что специальное упрощение детектора, которое возможно в случае кодов Файра, здесь неприменимо.

II. Принципы исправления ошибок

Пусть первые $n-r$ символов кодового слова длины n—информационные, а последние r—проверочные. Тогда кодовое слово описывается двоичной последовательностью $\{a_i\}$, $i = 0,1, \ldots, n-1$, где первые $n-r$ знаков выбираются произвольно и последние r удовлетворяют системе линейных уравнений вида

$$\sum_{i=0}^{n-1} a_i c_{ji} = 0, \quad j = 1, \ldots, r \qquad (1)$$

со сложением по модулю 2. Для циклических кодов двоичные коэффициенты c_{ij} порождаются невырожденной квадратной матрицей T порядка r, обладающей свойством $T^n = I$ и $T^s \neq I$ при $1 \leqslant s < n$. Пусть c_i образуют вектор-столбец $\{c_{1i}, c_{2i}, \ldots, c_{ni}\}$, и пусть u_1 — данный ненулевой вектор-столбец; тогда определим

$$c_i = T^i u_1, \quad i = 0, 1, \ldots, n-1.$$

При передаче некоторые из символов последовательности $\{a_i\}$ могут быть искажены; принятая последовательность будет обозначаться символом $\{b_i\}$, $i = 0,1, \ldots, n-1$. Запишем

$$b_i = a_i + e_i \pmod 2 \tag{2}$$

и назовем последовательность $\{e_i\}$ $(i = 0,1, \ldots, n-1)$ вектором ошибок или просто ошибкой. Ошибка представляется в форме полинома $E(t) = \sum_{i=0}^{n-1} e_i t^i$ с неопределенными коэффициентами t. На приемном конце вычисляется вектор

$$v = \left(\sum_{i=0}^{n-1} b_i T^i \right) u_1, \tag{3}$$

который с учетом (1) будет записываться в виде

$$v = \left(\sum_{i=0}^{n-1} e_i T^i \right) u_1. \tag{4}$$

В частности, при безошибочной передаче $v = 0$. И кодовое слово и вектор ошибок являются элементами векторного пространства $\mathcal{V}_n$ последовательностей длины n, состоящих из 0 и 1. Аналогично пространство векторов u и v размерности r, состоящих из элементов 0 и 1, обозначается символом $\mathcal{V}_r$. Среди всех возможных ошибок мы будем интересоваться только небольшим подмножеством S и для этих ошибок мы должны потребовать, чтобы они все давали различные ненулевые векторы пространства в $\mathcal{V}_r$. Тогда по вычисленному вектору v возможно будет опознать ошибку.

III. Циклы цепи

Матрица T описывает линейную последовательную цепь с обратной связью [8, 9]. Начальное состояние описывается вектором u, в то время как Tu, T^2u, ..., $T^nu = u$ определяют различные состояния, которые цепь проходит с течением времени. Векторы u и v принадлежат одному и тому же циклу тогда и только тогда, когда существует такое целое s, что $v = T^su$.

Для изучения циклической структуры матрицы T вводится ее характеристический полином $\varphi(x)$. В этой статье рассматриваются матрицы, характеристические полиномы которых имеют степень r и неприводимы. Для любой из них, кроме нулевой, имеется m циклов длины n_1, где m_1, n_1 и r удовлетворяют соотношению $m_1n_1 = 2^r - 1$.

Когда $2^r - 1$ разлагается на множители, может быть выбрано несколько подходящих длин циклов. При исправлении единичной ошибки Хафмен [8] оперировал полиномами, имеющими максимальную длину циклов $n_1 = 2^r - 1$, в то время как исправление кратных ошибок требует, чтобы $n_1 = n$ было собственным делителем числа $2^r - 1$. Тогда возможно задать $m_1 = m$ векторов $u_1, u_2, \ldots, u_m$, которые будут порождать все возможные циклы. Некоторое произвольное состояние описывается выражением T^iu_r, $i = 0,1, \ldots, n-1$. Дальнейшее изучение циклов цепи будет зависеть от некоторых свойств полей Галуа. Над простым полем J_2 с элементами 0 и 1 это поле $GF(2^r)$ содержит 2^r элементов. Пусть x есть корень уравнения $\varphi(x) = 0$, где $\varphi(x)$ — неприводимый полином степени r; тогда окажется, что $1, x, x^2, \ldots, x^{r-1}$ линейно независимы.

Любой элемент поля может быть записан в виде

$$F(x) = \sum_{i=0}^{r-1} f_i x^i, \quad f_i \in J_2. \tag{5}$$

Известно, что мультипликативная группа ненулевых элементов поля Галуа циклична. Символом $\mathcal{G}_q$ обозначим подгруппу порядка q. В частности, матрица T будет порождать поле Галуа, так как $\varphi(T) = 0$. Благодаря соотношению

$$F(T) = \sum_{i=0}^{n-1} b_i T^i \tag{6}$$

определяется отображение пространства $\mathcal{V}^{\rho}_{n}$ на поле Галуа. Более того, $F(T) \to F(T)u$ и ненулевой элемент u пространства $\mathcal{V}^{\rho}_{r}$ определяет отображение поля Галуа в векторное пространство $\mathcal{V}^{\rho}_{r}$. Это отображение взаимно однозначно. Для этого достаточно показать, что ненулевые элементы поля соответствуют ненулевым элементам векторного пространства и наоборот. Предположим, что $F(T)$ — ненулевой элемент. Как элемент поля он имеет обратный и соотношение $F(T)u = 0$ означает $Iu = 0$. Таким образом, ненулевой вектор u даст $F(T)u \neq 0$. Очевидно, если $u \neq 0$, $F(T)u \neq 0$, то $F(T) \neq 0$.

Лемма 1. *Если $F(T)$ — элемент поля Галуа $GF(2^r)$ и u — ненулевой вектор пространства $\mathcal{V}^{\rho}_{r}$, то отображение $F(T) \to F(T)u$ является взаимно однозначным отображением поля Галуа в пространстве $\mathcal{V}^{\rho}_{r}$.*

Лемма позволяет переформулировать необходимое и достаточное условие для исправления ошибки, которое теперь должно состоять в том, что элементы множества полиномов $E(T) = \sum_{i=0}^{r-1} e_i T^i$, соответствующего множеству S ошибок $\{e_i\}$, обязаны быть все различными и ненулевыми.

IV. Классы смежности мультипликативной группы поля $GF(2^r)$ по подгруппе $\mathcal{G}_n$

Определим классы смежности мультипликативной группы $\mathcal{G}$ поля Галуа по подгруппе $\mathcal{G}_n$ порядка n. Пусть y — первообразный элемент порядка $M = 2^r - 1$, т. е. элементы $1, y, y^2, \ldots, y^{M-1}$ составляют циклическую группу. Тогда $1, y^m, y^{2m}, \ldots, y^{(n-1)m}$ суть элементы подгруппы $\mathcal{G}_n$. Элемент x поля порождает класс смежности по подгруппе $\mathcal{G}_n$: $x, xy^m, xy^{2m}, \ldots, xy^{(n-1)m}$. Элементы x_1 и x_2 принадлежат одному и тому же классу смежности, только если существует такое целое $0 \leqslant s < n$, что $x_2 = x_1 y^{sm}$. Классы смежности не пересекаются. В качестве представителей классов смежности могут быть выбраны элементы $1, y, y^2, \ldots, y^{m-1}$.

Лемма 2. *Элементы $F_1(T)$ и $F_2(T)$ поля $GF(2^r)$ порождают один и тот же цикл цепи тогда и только*

тогда, когда они принадлежат к одному классу смежности группы $\mathcal{G}$ по подгруппе $\mathcal{G}_n$.

Доказательство. Предположим, что $F_1(T)$ и $F_2(T)$ принадлежат одному классу смежности. Тогда если u — ненулевой вектор пространства $\mathcal{V}^n$, $F_1(T)u$ и $F_2(T)u$ определят два положения цепи. Так как

$$F_2(T)u = [T^s F_1(T)]u = T^s[F_1(T)u], \tag{7}$$

они принадлежат одному циклу. С другой стороны, если для некоторого целого s выполнено (7), то

$$[F_2(T) + T^s F_1(T)]u = 0, \tag{8}$$

откуда по лемме 1 следует, что $F_2(T) + T^s F_1(T) = 0$. Таким образом, $F_1(T)$ и $F_2(T)$ принадлежат одному классу смежности.

V. Соотношение между векторами ошибок, классами смежности по подгруппе $\mathcal{G}_n$ и подгруппой $\mathcal{G}_m$

Пусть дан код длины n и матрица T, с помощью которой реализуется кодирующая схема. Спрашивается, к какому классу смежности группы $\mathcal{G}$ по подгруппе $\mathcal{G}_n$ будет относиться вектор ошибок $\{e_i\}$ при преобразовании $E(T) = \sum_{i=0}^{n-1} e_i T^i$. Один из путей решения этой задачи состоит в том, что сначала надо найти матрицу U порядка r, которая является первообразным корнем порядка $2^r - 1$ из единицы, такую, что $U^m = T$. Матрицы U и T порождают одинаковые поля с точностью до автоморфизма.

Так как $E(T) = E(U^m)$ — элемент поля, порожденного матрицей U, существует такое целое q, что $E(T) = U^q$. Если $1, U, U^2, \ldots, U^{m-1}$ суть лидеры (главные элементы) классов смежности, то E отождествляется с классом смежности, имеющим показатель $q_0 \equiv q \pmod m$ при $0 \leqslant q_0 < m$.

Однако при анализе конкретных кодов действительное вычисление числа q_0 не является необходимым. Достаточно знать, что некоторые ошибки соответствуют различным классам смежности группы $\mathcal{G}_n$. При этом полезен следующий результат.

Лемма 3. *Соотношение* $E(T) \to E(T)^n$ *есть взаимно однозначное отображение классов смежности группы* $\mathcal{G}_n$ *на подгруппу* $\mathcal{G}_m$ *группы* $\mathcal{G}$, *где* $mn = 2^r - 1$.

Доказательство. Очевидно $E(T)^n$ принадлежит подгруппе $\mathcal{G}_m$. Используя введенные обозначения, получаем $E(T) = U^q$, где $q = q_0 + q_1 m$, q_0 — фиксировано для всех элементов класса смежности. Каждый элемент отображается на один и тот же элемент группы $\mathcal{G}_m$, так как

$$E(T)^n = U^{nq_0} T^{q_1 nm} = U^{nq_0}. \tag{9}$$

Пусть $E(T)$ и $F(T)$ принадлежат различным классам смежности, т. е.

$$E(T) = U^{q_0 + q_1 m}; \quad F(T) = U^{s_0 + s_1 m} \tag{10}$$

и $m > q_0 > s_0 \geqslant 0$. Предположим, что $E(T)$ и $F(T)$ отображаются на один и тот же элемент в $\mathcal{G}_m$, тогда

$$E(T)^n = U^{q_0 n} = U^{s_0 n} = F(T)^n. \tag{11}$$

Из этого следует требование $U^{(q_0 - s_0)n} = 1$. Однако $0 < q_0 - s_0 < m$; следовательно, это равенство противоречит предположению о порядке матрицы U. Таким образом, различные классы смежности отображаются на различные элементы в $\mathcal{G}_m$. Окончательно, число классов смежности в $\mathcal{G}_n$ равно числу элементов в $\mathcal{G}_m$.

Пример: $r = 4$, $n = 5$, $m = 3$; $E_1(t) = 1$, $E_2(t) = 1 + t$; $E_3(t) = 1 + t^3$. Решение T уравнения $f(x) = x^4 + x^3 + x^2 + x + 1 = 0$ имеет цикл длины 5. Находим

$$\begin{aligned} &E_1(T)^5 = 1, \\ &E_2(T)^5 = (1+T)(1+T^4) = T^3 + T^2 + 1, \\ &E_3(T)^5 = (1+T^2)(1+T^{12}) = (1+T^3)(1+T^2) = \\ &\qquad = T^3 + T^2 = (T^3 + T^2 + 1)^2. \end{aligned} \tag{12}$$

В совокупности эти элементы образуют группу $\mathcal{G}_m$ и поэтому ошибки, определяемые полиномами $E_1(t)$, $E_2(t)$ и $E_3(t)$, все принадлежат различным циклам цепи.

VI. Классификация ошибок

В пространстве $\mathcal{V}_n$ векторов $\{e_i\} = \{e_0, e_1, \ldots, e_{n-1}\}$ вводится оператор сдвига S, такой, что $S\{e_i\} = \{e_{n-1}, e_0, e_1, \ldots, e_{n-2}\}$. Повторным применением оператора сдвига определяется вектор $S^q\{e_i\}$. Множество векторов, которое образуется последовательным применением операторов $S^0, S^1, \ldots, S^{n-1}$, называется циклом ошибок. Оператор сдвига разбивает пространство $\mathcal{V}_n$ на непересекающиеся классы, называемые циклами.

Легко показать, что если вектор $\{e_i\}$ соответствует элементу $E(T)$ посредством отображения, определенного равенством (6), то вектор $S\{e_i\}$ соответствует элементу $SE(T)$. Следовательно, сдвиги в векторе ошибок соответствуют сдвигам в цепи, реализующей матрицу T. Таким образом, все ошибки внутри цикла могут быть различимы приемником. Алгебраически это объясняется замечанием, следующим за леммой 1, и циклическим свойством матрицы T. Для выяснения корректирующих способностей кода достаточно рассмотреть по одному представителю каждого цикла ошибок. Для удобства этот вектор $\{e_i\}$ дается в нормализованном виде с $e_0 = 1$.

Вес вектора определяется числом его единиц. Очевидно, все члены цикла имеют одинаковый вес. Кроме оператора S, представляет интерес другое преобразование, сохраняющее вес. Этот оператор Q определяется соотношением

$$Q\{e_i\} = \{e_i\}; \quad e_i \longrightarrow e_{2i \bmod n}, \tag{13}$$

которое означает, что элемент, стоящий на i-м месте, передвигается на $2i$-е место по модулю n для всех i.

Лемма 4. *Преобразование Q, определенное соотношением* (13), *сохраняет вес. Если $\{e_i\}$ соответствует элементу $E(T)$, то $Q\{e_i\}$ соответствует элементу $E(T)$*[2].

Доказательство. Свойство сохранения веса следует из взаимно однозначного соответствия между векторами $\{e_i\}$ и $Q\{e_i\}$. Предположим, что это соответствие не имеет места; тогда должны найтись два целых числа $i_1 < i_2 < n$, такие, что $j \equiv 2i_2 \equiv 2i_1 \pmod n$. Однако n нечетно, а потому из сравнения $2(i_2 - i_1) \equiv 0 \pmod n$

следует $i_2 - i_1 \equiv 0 \pmod{n}$. Отсюда и из неравенств $0 \leqslant i_2 - i_1 < n$ следует, что $i_2 = i_1$, что противоречит предположению. Обозначим с помощью равенства

$$E(T) = \sum_{i=0}^{n-1} e_i T^i \tag{14}$$

элемент поля $GF(2^r)$, соответствующий вектору $\{e_i\}$. Тогда с учетом условия $T^n = 1$ $Q\{e_i\}$ дает

$$\sum_{i=0}^{n-1} e_i T^{2i \bmod n} = \sum_{i=0}^{n-1} e_i T^{2i} = E(T^2) = E(T)^2. \tag{15}$$

Повторным применением оператора Q строится множество полиномов $E(T)$, $E(T^2)$, $E(T^4)$, ..., $E(T^q)$, где $q = 2^{r-1}$. Очевидно, r последовательных применений оператора Q восстанавливают первоначальный вектор. Множество $\{e_i\}$, $Q\{e_i\}$, ..., $Q^{r-1}\{e_i\}$ образует класс векторов ошибок, называемый классом автоморфизмов (a—класс), порожденным вектором $\{e_i\}$. Такое название нами выбрано потому, что подстановка $T \longrightarrow T^2$ есть автоморфизм поля Галуа $GF(2^r)$. Соответственно $E(T)$, $E(T^2)$, ..., $E(T^q)$, где $q = 2^{r-1}$ называется классом автоморфизмов, порожденным в поле Галуа элементом $E(T)$.

Число различных векторов в классе автоморфизмов равно либо r, либо его делителю. Может случиться, что некоторые из членов принадлежат одному и тому же циклу ошибок. Чтобы выяснить, сколько различных циклов покрываются классом, должно быть проведено детальное исследование ошибок. Однако изучение классов автоморфизмов в поле Галуа может дать лишь частичное знание этого числа.

Пусть y—первообразный корень степени 2^{r-1} из единицы, и пусть $1, y, y^2, \ldots, y^{m-1}$ являются представителями классов смежности группы $\mathcal{S}$ по подгруппе $\mathcal{S}_n$. Элемент y^ν при последовательном применении преобразования $T \longrightarrow T^2$ будет превращаться в элементы $y^{2\nu}$, $y^{4\nu}$, ... Число различных классов смежности, к которым они принадлежат, обозначается символом $d(\nu, m)$ и определяется сравнением $\nu(2^q - 1) \equiv 0 \pmod{m}$, которое не выполняется ни при каком целом q меньшем, чем $d(\nu, m)$.

Ясно, что если μ есть наибольший общий делитель чисел ν и m, то $d(\nu, m) = d(1, m/\mu)$.

Лемма 5. *Если вектор ошибок* $\{e_i\}$ *соответствует элементу поля* $GF(2^r)$, *принадлежащему к классу смежности, который представлен элементом* y^ν, *то посредством преобразования* Q *из него получаются такие векторы ошибок* $\{e_i\}$, $Q\{e_i\}$, $Q^2\{e_i\}$, ..., $Q^{d-1}\{e_i\}$ *одинакового веса, что они порождают различные циклы цепи, когда* $d = d(\nu, m)$.

Доказательство. Согласно лемме 2 достаточно показать, что $Q^j\{e_i\}$, $j = 0, 1, \ldots, d-1$ относятся к различным классам смежности. Класс смежности, соответствующий вектору $Q^j\{e_i\}$, есть y^c, $c \equiv \nu 2^j \pmod m$. Положим, что $j > i$ и они дают один и тот же класс смежности. Тогда $\nu(2^j - 2) \equiv 0 \pmod m$ или $\nu(2^{j-i} - 1) \equiv 0 \pmod m$. Из определения следует, что $j - i = d = d(\nu, m)$, а это противоречит предположению, так как $j - i \leqslant d - 1$.

Лемма 6. *Между неприводимыми множителями полинома* $x^m + 1$ *и классами автоморфизмов, порожденными элементами подгруппы* $\mathscr{S}_m$, *имеется взаимно однозначное соответствие. В частности число различных элементов в классе равно степени соответствующего полинома.*

Доказательство. Пусть $z, z^2, \ldots, z^{m-1}, z^m = 1$, суть элементы подгруппы $\mathscr{S}_m$. Тогда $z^\nu, z^{2\nu}, \ldots, z^q$, где $q = 2^{d-1}$ и $d = d(\nu, m)$ образуют a-класс, порожденный элементом z^ν. Из теории полей Галуа известно, что все эти элементы являются корнями одного неприводимого полинома $\varphi(x)$, делящего полином $x^m + 1$. Повторяя эти рассуждения для z^μ, не содержащегося в предыдущем a-классе, мы исчерпаем всю подгруппу $\mathscr{S}_m$.

VII. Число циклов ошибок данного веса

Оператор сдвига S, определенный в предыдущем разделе, разбивает множество ошибок на непересекающиеся классы, называемые циклами ошибок. При рассмотрении числа этих классов удобно обращаться с ошибками в нормализованном виде. Таким образом, из каждой комбинации

ошибок веса w можно посредством оператора S получить столько различных комбинаций, каков вес w в нормализованном виде. Для получения в точности w комбинаций не должен встретиться ни один период длиной менее чем n.

Пусть $A(a, b)$ означает число нормализованных ошибок с периодом a, имеющих b отличных от нуля символов внутри периода. Число циклов ошибок этого типа будет тогда $A(a, b)/b$.

Обозначим символом $M(n, w)$ число циклов ошибок веса w в коде длины n и пусть d есть наибольший общий делитель чисел n и w. Тогда, если r — делитель числа d,

$$\left.\begin{aligned} M(n, w) &= \sum_{r|d} \frac{r}{w} A\left(\frac{n}{r}, \frac{w}{r}\right) \\ (n, w) &= d, \end{aligned}\right\} \tag{16}$$

так как множества комбинаций с различными периодами не имеют общих элементов. Всего имеется $\binom{n-1}{w-1}$ комбинаций ошибок в нормализованном виде, где $\binom{a}{b}$ есть биномиальный коэффициент. Сумма всех комбинаций ошибок различных периодов даст

$$\left.\begin{aligned} \sum_{r|d} A\left(\frac{n}{r}, \frac{w}{r}\right) &= \binom{n-1}{w-1}, \\ (n, w) &= d. \end{aligned}\right\} \tag{17}$$

Из ряда уравнений типа (17), используя хорошо известные результаты теории чисел, можно найти и удобно выразить с помощью функции Мёбиуса $\mu(m)$ функцию $A(n, w)$. Функция $\mu(m)$ задается равенствами $\mu(1) = 1$, $\mu(a) = 0$, если a содержит множитель p^2, p — простое; $\mu(p_1, p_2, \ldots, p_s) = (-1)^s$, если $p_1, p_2, \ldots, p_s$ — различные простые числа. Таким образом,

$$A(n, w) = \sum_{f|d} \mu(f) \binom{\frac{n}{f} - 1}{\frac{w}{f} - 1}. \tag{18}$$

Объединение равенств (16) и (18) и изменение порядка суммирования дают требуемое.

Теорема 1. *Число циклов ошибок веса w в коде длины n есть*

$$M(n, w) = \frac{1}{w} \sum_{j|d} \binom{\frac{n}{j} - 1}{\frac{w}{j} - 1} S(j), \tag{19}$$

где

$$S(j) = \sum_{k|j} k\mu\left(\frac{j}{k}\right), \tag{20}$$

$$d = (n, w). \tag{21}$$

В случае d простого

$$M(n, w) = \frac{1}{w}\left\{(p-1)\binom{\frac{n}{p} - 1}{\frac{w}{p} - 1} + \binom{n-1}{w-1}\right\}. \tag{22}$$

VIII. Анализ кодов, исправляющих ошибки

Изучение векторов ошибок и циклических свойств кодов привело к некоторым соотношениям, упрощающим анализ исправления ошибок, если дан неприводимый полином. Процесс упрощения применим как в случае пакетов ошибок, так и в общем случае кратных ошибок. Приведем основные результаты.

Теорема 2. *Пусть T удовлетворяет неприводимому уравнению степени r над полем J_2 с циклом, длина которого n есть собственный делитель числа $2^r - 1$. Множество S ошибок, которые должны быть исправлены, распадаются на циклы ошибок $\{S_j\}$, и из каждого цикла произвольно выбирается одна ошибка $\{e_{ji}\}$. Необходимое и достаточное условие исправления ошибки состоит в том, что все элементы*

$$E_j(T)^n = \sum_{i=0}^{n-1} e_{ji} T^i \tag{23}$$

различны и отличны от нуля.

Доказательство. Посредством отношения $\{e_{ji}\} \to E_j(T)u$, где u — ненулевой вектор пространства $\mathcal{V}^0_r$, каждая ошибка определяет одно состояние цепи. Полный цикл ошибок порождает все состояния цикла цепи, и отсюда все ошибки внутри цикла ошибок различимы.

Чтобы выяснить, порождают ли различные циклы ошибок также различные циклы цепей, не равные нулевому состоянию при отсутствии ошибки, следует объединить леммы 2 и 3; они покажут, что это будет иметь место тогда и только тогда, когда элементы $E_j(T)$ не равны нулю и отличаются друг от друга, когда ошибки $\{e_{ji}\}$ принадлежат отдельным циклам ошибок.

При рассмотрении конкретных кодов некоторые случаи могут быть исключены в самом начале еще при сравнении числа ошибок и циклов цепи. Дальнейшее упрощение, как будет показано ниже, достигается посредством изучения a-классов в пространстве $\mathcal{V}_n$ в поле $GF(2^r)$.

В следующем разделе будут даны некоторые частные результаты. Что же касается получения результатов, обладающих высокой степенью общности, то в большинстве случаев это оказывается достаточно трудным. Отсюда для каждого значения n длины кода рассмотрение зачастую должно вестись отдельно. Существенным исключением является случай исправления двойной ошибки, уже рассмотренный Меггитом [7].

IX. Исправление единичной и смежных двойных ошибок

Пусть $r=2p$ есть степень полинома и $n=(2^{2p}-1)/3$ — длина кода. Тогда существует 3 цикла ошибок. Чтобы исправить единичную и смежную двойную ошибки, необходимо и достаточно согласно теореме 2 найти примитивный корень n-й степени из единицы (обозначим его z) такой, что $(1+z)^n \neq 1$, так как очевидно, что $(1+z)^n \neq 0$.

Обозначим полином деления круга порядка n символом $\Phi_n(x)$. Тогда z удовлетворяет уравнению $\Phi_n(x)=0$. Это доказывает первую часть следующей теоремы.

Теорема 3. 1) *Для исправления кодом длины* $n=(2^{2p}-1)/3$ *единичных и смежных двойных ошибок необходимо и достаточно, чтобы* $\Phi_n(x)$ *не делил полинома* $(1+x)^n+1$.

2) *Достаточным условием является неравенство*

$$2(2^{2p-2}-1) < 3\varphi(n), \tag{24}$$

где $\varphi(n)$ *— функция Эйлера.*

Доказательство. Для доказательства последней части рассмотрим те значения z. которые удовлетворяют обоим уравнениям

$$\begin{aligned}(1+x)^n &= 1,\\ x^n &= 1.\end{aligned} \tag{25}$$

Пусть $n = N(p-1)$, где

$$N(p) = 2^{2p} + 2^{2(p-1)} + \ldots + 64 + 16 + 4 + 1. \tag{26}$$

Умножая первое из уравнений (25) на $x^{N(p-2)}$ и используя второе уравнение, получаем

$$(1+x)^{N(p-2)}\,(1 + x^{N(p-2)}) = x^{N(p-2)}. \tag{27}$$

Так как из уравнения $\Phi_n(x) = 0$ следует $x^n + 1 = 0$ и $\Phi_n(x) = 0$ имеет только простые корни, не все его корни могут удовлетворить уравнению (27), если степень $2N(p-2)$ меньше, чем степень $\varphi(n)$ полинома $\Phi_n(x)$. Это и есть условие выполнения неравенства (24). Численные подсчеты показывают, что неравенство (24) выполняется для $p = 3$, 4, 5, 7, 8 и 9. Применением теоремы 2 и конкретных полиномов показано также, что исправление ошибок возможно и при $p = 2$ и 6. Для $p = 2$ единственным является полином $x^4 + x^3 + x^2 + x + 1$, и соответствующий анализ проведен в примере 1. При $p = 6$ единственной возможностью является полином $x^{12} + x^5 + x^4 + x^3 + x^2 + x + 1$, взятый из [10].

Во всех случаях может исправляться еще одна ошибка. Имея в виду исправление пакета ошибок, следует полагать в этой ошибке $\{e_i\}$ $e_0 = 1$, $e_2 = 1$ и все остальные $e_i = 0$. Возможность исправления следует из леммы 5, так как $1 + T$ и $1 + T^2$ принадлежат одному a-классу.

X. *Дальнейшие результаты относительно исправления пакетов ошибок*

Число пакетов ошибок длины l или менее равно 2^{l-1}. Если $l = 1$, то пакет обращается в единичную ошибку. Рассмотрим целые $m > 2^{l-1}$, являющиеся делителями числа $2^r - 1$.

Пакеты длиной от 2 до 6 изучены численными методами и с помощью полиномов степени $6 \leqslant r \leqslant 18$. Таблица

неприводимых полиномов вместе с данными о длине циклов содержится в [10]. Расчеты выполнены на IBM 7090 способом, указанным в теореме 2.

Таблица I

Коды, исправляющие пакеты ошибок длины l, использующие неприводимые полиномы степени r. n — длина кода, m — число циклов цепи. Дано число испытанных и испытанных успешно полиномов

l	r	n	m	Число полиномов	
				испытанных	успешно испытанных
2	6	21	3	1	1
	8	85	3	4	3
	10	341	3	15	11
	12	1365	3	24	16
	14	5461	3	189	129
	16	21845	3	513*	344
	18	87381	3	1296**	871
3	6	9	7	1	1
	8	51	5	2	1
	9	73	7	4	2
	12	819	5	18	9
	12	585	7	12	7
	15	4681	7	93**	41
	16	13107	5	256	87
4	10	93	11	3	0
	12	455	9	12	0
	12	315	13	6	2
	18	29127	9	432	7
5	12	195	21	4	0
	16	3855	17	64	0
	18	13797	19	216	0
6	12	117	35	3	0
	12	105	39	2	0

* Таблица [10] содержит 513 полиномов этой категории вместо 512.
** Было испытано 93 из 150 полиномов. Не был испытан ни один полином на стр. 27 и 28 [10].

В табл. 1 собраны данные о числе испытанных и успешно испытанных полиномов. По существу были испытаны все полиномы некоторой категории, за исключением тех случаев, которые отмечены в таблице. Заметим, что если полином имел несимметричное расположение коэффициентов, испытывалась только одна из возможностей $\varphi(x)$ и $x^r\varphi(1/x)$. С точки зрения исправления ошибок некоторые полиномы равноценны своим взаимным. Таблица содержит также графы l, r, n и m, означающие длину пакета, степень полинома, длину кода и число циклов цепи соответственно. Из таблицы видно, что сначала испытывались коды, которые соответствуют наименьшим возможным m и, следовательно, наибольшим возможным n. Однако в случае $r=12$ рассмотрено по два различных n для $l=3$ ($n=819, 585$), $l=4$ ($n=455, 315$) и $l=6$ ($n=117, 105$). В результате этих исследований найдено несколько кодов для пакетов длины 2 и 3; именно, не менее одного для каждой испытанной комбинации параметров. Для $l=2$ этот результат находится в полном согласии с теоремой 3 и соответствующими численными подсчетами. По-видимому, достаточно трудно отыскать коды теоретически максимальной длины, исправляющие более длинные пакеты. Не найдено ни одного кода для $l=5$ и 6, тогда как для $l=4$ имеются некоторые возможности при $r=18$; однако длина n кода в этом случае выглядит практически совершенно недостижимой.

Рассматривая более короткие коды, мы, по-видимому, сможем получить большее количество кодов для исправления пакетов длины 4, 5, 6 и т. д. Об этом свидетельствует положительный результат для комбинации $l=4$, $r=12$, $n=315$. В качестве иллюстрации приведем следующий пример с расчетами, выполненными вручную.

Пример 2. Пакет длины $l=3$, степень $r=8$, длина кода $n=51$, полином $x^8+x^4+x^3+x+1$ или в восьмеричной системе 433

$$\begin{aligned} E(T) &= 1 \\ &= 1+T \\ &= 1+T+T^2, \\ E(T)^n &= 1 = a^5 \\ &= T^3+T^2 = a \\ &= T^7+T^6+T^5+T^3+T^2+1 = a^3. \end{aligned}$$

Из этого заключаем

$$E(T) = 1 + T^2 \qquad E(T^n) = a^2$$
$$= 1 + T, \qquad = a^4.$$

Таким образом, может быть исправлена одна из крайних ошибок и удобно положить $e_0 = 1$, $e_3 = 1$.

В табл. II внесены все полиномы, которые делают возможным исправление ошибок с помощью укороченного кода проанализированной длины. Коэффициенты полинома представлены в восьмеричной форме, причем сначала коэффициенты записываются в виде двоичного числа, начинающегося со старшей степени. Это число разбивается на группы по три двоичных числа, начиная справа, а слева, если это необходимо, добавляются нули. Например, $x^6 + x^4 + x^2 + x + 1$ записывается в виде 001 010 111, или 127.

Таблица II

Неприводимые полиномы для исправления пакетов ошибок длины l; длина кода n, степень полинома r

l	r	n	Полиномы
2	6	21	127
	8	85	477; 567; 613
	10	341	2017; 2107; 2123; 2143; 2231; 2355; 2547; 2633; 2653; 3277; 3367
	12	1365	10077; 10757; 11103; 11155 11545; 11735; 11763; 12133 12265; 13033; 13737; 13773 14007; 14667; 15457; 16017
3	6	9	111
	8	51	433
	9	73	1027; 1113
	12	819	10041; 10467; 10653; 11031; 11673; 13223; 13377; 14037 14513
	12	585	11433; 11637; 12153; 13113 13157; 14177
4	12	315	10027; 11105

Эффективность кода можно оценивать по соотношению между заданным числом контрольных символов и данной длиной пакета, с одной стороны, и полученной при этом длиной кодовой комбинации, с другой. В таком случае описанные здесь коды менее эффективны, чем коды Абрамсона [1, 2] для исправления пакетов длины 2 и 3. Соответствующие длины кода приведены в нижеследующей таблице (табл. IIа), где r — степень полинома.

Таблица IIа

Длина пакета	Длина кода	
	приводимый полином	непривсдимый полином
2	$(2^r-2)/2$	$(2^r-1)/3$
3	$(2^r-4)/4$	$(2^r-1)/5$

Сравнение эффективности кодов Файра [7] и кодов, представленных здесь, оказывается в пользу последних.

XI. Исправление двойной ошибки

Пусть $r=2p$ есть степень полинома $\varphi(x)$ и $n=2^p+1$ есть длина цикла. Для кода этого типа рассмотрим исправление всех единичных и двойных ошибок. Число циклов цепи есть $m=2^p-1$.

Единичные ошибки:

$$E(T)=1, \qquad E(T)^n=1.$$

Двойные ошибки:

$$E(T)=1+T^j, \quad 1\leqslant j<n; \quad E(T)^n=T^j+T^{-j}.$$

Заметим сразу, что $E(T)^n$ отлично от нуля. В противном случае $T^j+T^{-j}=0$, откуда $T^{2j}+1=0$ и потому n делит j, что противоречит условию. Пусть $1+T^{j_1}$ и $1+T^{j_2}$ представляют ошибки из различных циклов ошибок. Исследуем, принадлежат ли они также и различным классам смежности. Если они принадлежат одному

классу смежности, то

$$T^{j_1}+T^{-j_1}=T^{j_2}+T^{-j_2}, \tag{28}$$

или

$$(T^{j_1-j_2}+1)\,(T^{j_1+j_2}+1)\,T^{-j_1}=0. \tag{29}$$

Если T — первообразный корень степени n из единицы, то должно выполняться одно из следующих соотношений:

$$\left.\begin{aligned} j_1-j_2 &\equiv 0 \pmod n,\\ j_1+j_2 &\equiv 0 \pmod n. \end{aligned}\right\} \tag{30}$$

Так как $1\leqslant j_1,\ j_2<n$, то или $j_1=j_2$ или $j_1=n-j_2$, откуда следует, что ошибки принадлежат одному циклу ошибок, а это противоречит условию. Но тогда все двойные ошибки, принадлежащие различным циклам ошибок, принадлежат и различным циклам цепи.

Далее рассмотрим возможность того, что двойная ошибка принадлежит тому же циклу цепи, что и единичная. Условие этого состоит в равенстве

$$T^j+T^{-j}=1. \tag{31}$$

Достаточно рассмотреть j нечетные, так как T^j+T^{-j} принадлежит циклической группе нечетного порядка. Согласно (31) $1+T^j$ и $1+T^{2j}$ принадлежат одному классу смежности. Тогда (28) выполняется при $j_1=j$ и $j_2=2j$ и (30) дает единственную возможность $3j\equiv 0 \pmod n$. Таким образом, n делит $3j$, но не делит j, так как $n>j$. Если существует такое целое $s>1$, что $sn=3j$, то s не должно быть равным 3 или четным. Тогда $s\geqslant 5$ и $n<j$, что приводит к противоречию. Окончательно $n=3j$. Если длина кода не делится на 3, различные циклы ошибок дают различные циклы цепи.

Положим $n=3j$, и пусть $\varphi(x)$ — неприводимый полином с длиной цикла n. Тогда $\varphi(x)$ делит $x^{3i}+1$. Но $\varphi(x)$ не может делить x^j+1, не приводя к противоречию с предположением о длине цикла; таким образом, $\varphi(x)$ делит $x^{2j}+x^j+1$. Отсюда уравнение $\varphi(T)=0$ дает $T^{2j}+T^j+1=0$. Это соотношение равносильно равенству (31); $1+T^j$ и 1 должны принадлежать одному классу смежности. Поэтому условие, наложенное на n, является также и необходимым. Легко доказать, что $n=2^p+1$

делится на 3 тогда и только тогда, когда p нечетно. Эти результаты суммированы в теореме, первая часть которой доказана Меггитом [7]; приведем ее здесь без доказательства.

Теорема 4. *Если* $r=2p$ *и* p *четно, все неприводимые полиномы степени* r *и цикла длины* 2^p+1 *могут исправлять все единичные и двойные ошибки. При* p *нечетном такой полином найти невозможно.*

XII. Частные случаи исправления тройных ошибок

Известно [11], что существует код длины 23 с 11 проверочными символами, который может исправлять все единичные, двойные и тройные ошибки. Здесь будет доказано, что этот код может быть получен с помощью регистра сдвигов[1]).

Существует два неприводимых полинома $\varphi_1(x)$ и $\varphi_2(x)$ степени 11, каждый из которых дает 89 циклов цепи длины 23. Так как $\varphi_2(x)=x^{11}\varphi_1(1/x)$, то они эквивалентны с точки зрения исправления ошибок и мы рассмотрим только полином $\varphi_1(x)=x^{11}+x^9+x^7+x^6+x^5+x+1$. Для определения числа циклов ошибок в формуле (22) следует положить $p=1$. Единичные ошибки определяют один цикл ошибок, двойные ошибки — 11 циклов ошибок и тройные ошибки — 77 циклов; все вместе они дают 89 циклов ошибок и таким образом обеспечивается исправление всех ошибок кратности не более 3. Далее ошибки классифицируются по a-классам. Имеется 11 различных элементов в каждом a-классе как в поле Галуа (в силу того, что $d(\nu, 89)=11$), так и в множестве ошибок (в силу леммы 5). Это означает, что все двойные ошибки принадлежат одному a-классу, тогда как тройные ошибки могут быть разбиты на 7 таких классов. В качестве

[1]) Референт указал, что подобное доказательство содержится в статье [12]. Однако доказательство, приведенное здесь, использует a-классы, что значительно упрощает рассмотрение.

возможных представителей ошибок берем в полиномиальной форме

$$E_A(x)=1, \qquad E_4(x)=1+x+x^5,$$
$$E_B(x)=1+x, \qquad E_5(x)=1+x+x^7,$$
$$E_1(x)=1+x+x^2, \qquad E_6(x)=1+x+x^9,$$
$$E_2(x)=1+x+x^3, \qquad E_7(x)=1+x+x^{13}.$$
$$E_3(x)=1+x+x^4,$$

Если T удовлетворяет полиному $\varphi_1(x)=0$, то

$$E_A(T)^{23}=1=1,$$
$$E_B(T)^{23}=1+T^8+T^5+T^9+T^{10}=z,$$
$$E_1(T)^{23}=T+T^2+T^6+T^8+T^9=z^{77},$$
$$E_2(T)^{23}=T^2+T^{10}=z^{70},$$
$$E_3(T)^{23}=T^2+T^4+T^8+T^9+T^{10}=z^{24},$$
$$E_4(T)^{23}=T^2+T^3+T^4+T^5+T^6+T^8+T^9=z^{22},$$
$$E_5(T)^{23}=T+T^2+T^6=z^{59},$$
$$E_6(T)^{23}=1+T+T^2+T^3+T^5+T^6+T^7+T^{10}=z^{79},$$
$$E_7(T)^{23}=1+T+T^4+T^5+T^7+T^8+T^9=z^{17}.$$

Все элементы выражаются в виде степеней z, и каждый из них есть примитивный элемент подгруппы, так как 89 есть простое число. Проверкой показателей степени устанавливается, что все элементы принадлежат различным a-классам.

Различные a-классы ошибок соответствуют различным классам поля, что по лемме 5 и теореме 2 является достаточным условием для исправления ошибок.

Для получения списка показателей степени можно сначала классифицировать все 89 показателей по a-классам; таким образом, 1, 2, 4, 8, 16, 32, 39, 64, 78, 67 и 45—для первого класса, затем 3, 6, 12, 24, 48, 7, 14, 28, 56, 23, 46—для второго класса и так далее. Ни один из элементов не соответствует степени z^3, но последовательное определение степеней z^6, z^{12} и z^{24} устанавливает соответствие между $E_3(T)^{23}$ и показателем 24. Затем переходим к следующему a-классу, начинающемуся показателем 5, и выполняем операции того же типа.

XIII. Полиномы для исправления кратных ошибок

В этом разделе будут рассмотрены более общие полиномы для исправления кратных ошибок. Мы не будем требовать, чтобы полином $f(x)$ был неприводимым и чтобы решение T уравнения $f(x)=0$ с необходимостью имело цикл цепи, равный длине кода. Здесь все еще может использоваться техника регистра сдвигов и свойства кода могут изучаться в терминах вектора v, определенного в разделе II. Исследование для этих полиномов ведется посредством некоторого алгоритма. Первоначально [13] это было установлено и доказано в терминах двоичных фильтров, введенных Хафменом [8]. Здесь, однако, это будет установлено в полиномиальной форме, вполне уместной для построения кодов. Такая же связь была открыта Меласом [3]; поэтому здесь доказательство будет опущено.

Теорема 5. *Для исправления всех ошибок кратности q или менее в коде длины n, построенном с помощью полинома $f(x)$, необходимо и достаточно, чтобы число отличных от нуля коэффициентов произведения $t(x)f(x)$ было не менее $2q+1$ для всех полиномов $t(x)$ степени $n-r-1$ или меньше.*

Как следствие отсюда получается, что полином $f(x)$ должен иметь не менее $2q+1$ отличных от нуля коэффициентов, чтобы исправлять все ошибки веса q или менее.

На основании этой теоремы была составлена программа работы электронной машины с целью отыскания возможных полиномов степени 8 и 9 для исправления единичных и двойных ошибок. Аналогичный эксперимент был проведен со степенями 10 и 11 для исправления всех ошибок кратности не более 3.

Таблицы III и IV содержат список полиномов и относящиеся к ним результаты для наибольших длин n, для которых они могут быть использованы. Полиномы описаны в терминах их неприводимых множителей. Для неприводимых полиномов даны длины циклов. Полиномы и их множители представлены в восьмеричной форме. Если полином $f(x)$ может исправлять желаемое число

Таблица III

Полиномы степени 8 и 9, которые исправляют все единичные и двойные ошибки в коде длины n. Даны либо неприводимые множители полинома, либо длина L цикла, если полином неприводим

r	N_0	Полином	Структура	n
8	1	427	31; 37	15
	2	433	$L=51$	13
	3	435	$L=255$	14
	4	453	$L=255$	13
	5	465	3; 147	13
	6	471	$L=17$	17
	7	507	13; 45	13
	8	515	$L=255$	14
	9	557	3; 3; 23	14
	10	573	$L=85$	13
	11	727	$L=17$	17
9	1	1143	3; 3; 57	20
	2	1207	$L=511$	19
	3	1303	15; 147	19
	4	1413	3; 15; 31	22
	5	1467	13; 165	21

ошибок, то этим же свойством обладает и полином $x^r f(1/x)$. Если эти полиномы различны, то указан только тот полином, который представляется меньшим двоичным числом. Полиномы № 6 и 11 степени 8 неприводимы и представляют собой частные случаи полиномов теоремы 4. Полином № 3 степени 11 тождествен полиному, изученному в разделе XII. Обнаружено также, что полином № 1 степени 8 представляет собой пример полинома, порождающего код Боуза—Чоудхури и таковым же является полином № 3 степени 10. Наиболее эффективные полиномы степени 8, 10 и 11 могут быть получены с помощью уже существующей теории. В то же время наилучшие полиномы степени 9 дают циклическую структуру, которая не объясняется этими теориями. Интересно также отметить, что многие неприводимые полиномы, приведенные в таблицах, имеют

Таблица IV

Полиномы степени $r=10$ и 11, которые исправляют все единичные, двойные и тройные ошибки в коде длины n. Даны либо неприводимые множители полинома, либо длина L цикла, если полином неприводим

r	N_0	Полином	Структура	n
10	1	2327	$L=1023$	14
	2	2353	3; 13; 73	14
	3	2467	3; 31; 37	15
	4	2473	13; 221	14
	5	2635	13; 203	14
	6	2647	3; 13; 67	14
	7	2723	13; 211	14
	8	3227	15; 203	14
11	1	4355	3; 1517	17
	2	4517	3; 3; 3; 75	17
	3	5343	$L=23$	23

длину цикла большую, чем длина кода. Поэтому различные комбинации ошибок хотя и располагаются внутри одного цикла цепи, но разделены настолько, что могут быть различимыми.

Вычисления, приведенные в разделе XIII, были выполнены на FACIT EDB и запрограммированы Ангсмарком. Программирование на машине IBM 7090 для задач, изложенных в разделе X, было выполнено Брандстрёмом и Алмеклйнтом. Я обязан также Брандстрёму за плодотворное обсуждение содержания статьи.

ЛИТЕРАТУРА

1. Abramson N. M., A class of systematic codes for nonindependent errors, *IRE Trans. Information Theory*, **IT-5**, December (1959), 150—157.
2. Abramson N. M., Error correcting codes from linear circuits, Stanford Electronic Labs., Stanford University, Stanford, Calif., Tech. Rept. № 2002—1; June 13 (1960).
3. Melas C. M., A cyclic code for double error correction, *IBM J. Res. and Dev.*, **4**, July (1960), 364—366.

4. Bose R. C., Ray-Chaudhuri D. K., On a class of error correcting binary group codes, *Inf. and Control*, **3**, March (1960), 68—79. [Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 2, ИЛ, М., 1961, 83—94.]
5. Van der Waerden B. L., Algebra I, Springer Verlag, Berlin, Germany; 1955.
6. Meggitt J. E., Error correcting codes for correcting bursts of errors, *IBM J. Res. and Dev.*, **4** (1960), 329—334.
7. Meggitt J. E., Error correcting codes and their implementation for data transmission systems, presented at IRE meeting, Delft, The Netherlands; September 1960.
8. Huffman D., D., A linear circuit viewpoint on error-correcting codes, *IRE Trans. Information Theory*, **IT-2**, September (1956), 20—28.
9. Elspas B., The theory of autonomous linear sequential networks, *IRE Trans. Circuit Theory*, **CT-6**, March (1959), 45—60. [Русский перевод: Элспас Б., Теория автономных линейных последовательных сетей, Кибернетический сб., вып. 7, ИЛ, М., 1963, 90—128.]
10. Marsh R. W., Table of irreducible polynomials over *GF*(2) through degree 19, Office of Tech. Services, U. S. Dept. of Commerce, Washington, D. C., PB161693, October 1957
11. Golay M. J. E., Notes on digital coding, *Proc. IRE* (correspondence) **37**, June (1949), 657.
12. Prange E., Cyclic error-correcting codes in two symbols, Cambridge Res. Ctr., Bedford Mass., TN-57-103; September 1957.

ЗАМЕТКИ О ЦИКЛИЧЕСКИ ПЕРЕСТАНОВОЧНЫХ КОДАХ, ИСПРАВЛЯЮЩИХ ОШИБКИ [1])

П. Нейман

Рассматриваются корректирующие коды с n двоичными цифрами в кодовом слове, в которых расстояния достаточно велики (порядка $n/2$) и число кодовых слов относительно мало (порядка $2n$). В частности, показывается связь разностных множеств с циклическими кодами, в которых половина кодовых слов — циклические перестановки одного из них, а остальные — их дополнения. Благодаря этому существование или несуществование определенных оптимальных и близких к оптимальным кодов устанавливается в терминах соответствующих разностных множеств, для которых известны различные теоремы существования и методы построения. Несмотря на простоту кодирования и декодирования при больших n, эти коды невыгодны для передачи, так как скорость передачи асимптотически равна нулю. Тем не менее они представляют интерес для использования в переключательных цепях, т. е. в устройствах матриц распределения нагрузок.

I. Введение

В настоящей заметке содержится несколько результатов, связывающих разностные множества с известными кодами, исправляющими ошибки, каждый из которых порождается из одного своего кодового слова с помощью циклических перестановок. На основании этих результатов устанавливается существование или несуществование различных оптимальных и близких к оптимальным кодов.

Имеются два основания для изучения этих кодов. Одно из них состоит в том, что любой циклически перестановочный код прост для кодирования и декодирования при исправлении ошибок на основе критерия максимального правдоподобия. Эти коды рассмотрены Нейманом [11]. Другим основанием является то, что такие коды могут

[1]) Neumann P. G., A note on cyclic permutation error-correcting codes, *Information and Control*, 5 (1962), № 1, 72—86.

быть использованы в логических устройствах переключательных цепей, т. е. в матрицах распределения нагрузок (Такахаси и Гото [17], Цзянь [4], [5], Нейман [12]).

Двоичный корректирующий код обозначается через $A(n, d; N)$, где N — число кодовых слов, n — число двоичных цифр в каждом слове и d — *кодовое расстояние*, т. е. минимальное расстояние между кодовыми словами (как обычно, расстояние между кодовыми словами определяется как число соответственных позиций, в которых кодовые слова различны). Если $d=2e+1$, то код исправляет e ошибок; если $d=2e+2$, то код исправляет e ошибок и обнаруживает $e+1$ ошибку. Корректирующие коды подробно рассматриваются в книге Питерсона [13].

Циклически перестановочным кодом $P'(n, d'; 2n)$ называется код, в котором n кодовых слов являются циклическими перестановками одного слова, а остальные n кодовых слов являются двоичными дополнениями первых n кодовых слов. Циклически перестановочный код $P(n, d; 2n+2)$ получается из $P'(n, d'; 2n)$ включением кодового слова, состоящего только из нулей, и кодового слова, состоящего только из единиц. Таким образом, любое кодовое слово, отличное от 0 или 1, порождает код своими циклическими перестановками и дополнениями. Вследствие того что число кодовых слов относительно мало ($2n$ или $2n+2$), скорость передачи при больших n асимптотически равна нулю, изменяясь как $(1+\log_2 n)/n$. С другой стороны, расстояния весьма велики, поскольку их порядок равен $n/2$. Если расстояние d нечетно, то хорошо известно, что можно увеличить d на единицу, добавив один разряд; код, полученный в результате добавления соответствующего проверочного разряда, обозначим через $P^*(n+1, d+1; 2n+2)$, где d нечетно. Например, циклически перестановочный код $P(7, 3; 16)$ приводится в таблице 1 (это известный код Хэмминга). Там же указан соответствующий код $P^*(8, 4; 16)$. Этот пример подробно рассматривается в конце второго раздела.

Рассматриваются четыре случая, именно: $n=4m-1$, $4m$, $4m+1$ и $4m+2$. Если положить $n=4m-1$ и $N=8m$, то кодовое расстояние любого кода $A(4m-1, d; 8m)$ не превышает $2m-1$. Код $P(4m-1, 2m-1; 8m)$, если он существует, *оптимален* в том смысле, что не существует

другого кода $A(4m-1,\ 2m-1;\ N)$ с теми же n и d, который содержит более, чем $8m$ кодовых слов (Плоткин [14]). Боуз и Шрикханде показали [2], что коды $A(4m-1,\ 2m-1;\ 8m)$ и $A(4m,\ 2m;\ 8m)$ существуют в том и только том случае, когда существует матрица Адамара H_{4m} или, что равносильно, в том и только том случае, когда существует некоторая симметрическая сбалансированная блок-схема (см. ниже). Неизвестны m, для которых не существует матрицы Адамара H_{4m}, хотя такие матрицы не найдены в некоторых случаях, например $m=29$ и $m=39$. В то время как коды $A(4m-1,\ 2m-1;\ 8m)$ существуют, возможно, для всех m, коды $P(4m-1;\ 2m-1;\ 8m)$ существуют для многих m, но не для всех, например, они не существуют при $m=7,\ 10,\ 13,\ 14$. Вопрос о существовании этих кодов обсуждается в третьем разделе.

Если $n=4m+1$ и $N=8m+4$, то расстояние d любого кода не превышает $2m$. Однако, тогда как коды

$$P(4m-1,\ 2m-1;\ 8m)$$

существуют для многих чисел m, коды

$$P(4m+1,\ 2m;\ 8m+4)$$

не существуют. Ниже будет показано, что коды $P'(4m+1, 2m;\ 8m+2)$ иногда существуют, хотя они не оптимальны. В случае $n\leqslant 220$ они существуют только для $n=5$ и $n=13$. Вопрос о существовании таких кодов обсуждается в четвертом разделе. Упоминаются еще коды $P(4m+1,\ 2m-1;\ 8m+4)$, хотя они также неоптимальны.

Если $n=4m$ и $N=8m+2$, то d не превышает $2m-1$, однако код $P(4m,\ 2m-1;\ 8m+2)$ существует только при $m=1$. Оптимальный код $P'(4m,\ 2m;\ 8m)$ существует при $m=1$. Как упоминалось выше, имеется много кодов $P^*(4m,\ 2m;\ 8m)$. Вопрос о существовании кодов P' обсуждается также в четвертом разделе.

Если $n=4m+2$, $N=8m+6$, то кодовое расстояние не превышает $2m$. Коды $P(4m+2,\ 2m;\ 8m+6)$ существуют почти для всех (если не для всех) значений m; однако они далеки от оптимальных и не рассматриваются здесь.

Таблица I

Циклический перестановочный код $P(7, 3; 16)$

Кодовое слово	← $P^*(8, 4; 16)$ →							
	← $P(7, 3; 16)$ →							
$i=0$	1	1	1	0	1	0	0	0
1	0	1	1	1	0	1	0	0
2	0	0	1	1	1	0	1	0
3	1	0	0	1	1	1	0	0
a_i 4	0	1	0	0	1	1	1	0
5	1	0	1	0	0	1	1	0
6	1	1	0	1	0	0	1	0
0	0	0	0	0	0	0	0	0
$i=0$	0	0	0	1	0	1	1	1
1	1	0	0	0	1	0	1	1
2	1	1	0	0	0	1	0	1
3	0	1	1	0	0	0	1	1
a_i 4	1	0	1	1	0	0	0	1
5	0	1	0	1	1	0	0	1
6	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	1
j	0	1	2	3	4	5	6	7

II. Разностные множества

Для математического обоснования циклически перестановочных кодов желательно ввести понятие разностного множества. Оказывается, что каждое разностное множество приводит к циклически перестановочному коду и и что обратное утверждение имеет место в нескольких интересных случаях.

(v, k, λ)-*разностным множеством* $D_0 : d_1, \ldots, d_k \pmod v$ называется множество k различных целых чисел, такое, что каждое целое число $1, 2, \ldots, v-1$ встречается

среди разностей $d_\alpha - d_\beta \pmod v$ ровно λ раз [1]). Отсюда следует, что каждое множество $D_i : d_1 + i, \ldots, d_k + i \pmod v$ является разностным множеством с теми же самыми v, k и λ, $i = 0,1, \ldots, v-1$. В соответствующей *циклической матрице инцидентности* $A = \|a_{ij}\|$ элемент a_{ij} равен 1, если число j содержится в i-м разностном множестве $d_1 + i, \ldots, d_k + i \pmod v$, $i, j = 0,1, \ldots, v-1$, и равен 0 в противном случае [2]).

Дополнением разностного множества (v, k, λ) называется множество целых чисел по модулю v, не содержащихся в исходном разностном множестве. Дополнение оказывается разностным множеством $(v, v-k, v-2k+\lambda)$. Соответствующая матрица инцидентности является *дополнением* исходной матрицы инцидентности и может быть получена из последней заменой 0 на 1.

Лемма 1. *Если существуют* (v, k, λ)*-разностные множества* $D_i : d_1 + i, d_2 + i, \ldots, d_k + i \pmod v$, $i = 0,1, \ldots, v-1$, *то расстояние между любыми двумя строками циклической матрицы инцидентности* $A = \|a_{ij}\|$ *равно* $2k - 2\lambda$. *Аналогичный результат справедлив для дополнительной матрицы инцидентности* $\overline{A}$.

Доказательство. Разности $l = d_\alpha - d_\beta \pmod v$ в D_0 соответствует совпадение единиц в строках матрицы инцидентности, отвечающих D_0 и D_l по столбцу d_α, так как $d_\alpha = l + d_\beta$ содержится в D_0 и в D_l. В силу того что каждая разность l встречается ровно λ раз в D_0, совпадение единиц в этих строках происходит ровно λ раз. Для каждого i

---

[1]) Исследование разностных множеств и их перечень для $3 \leqslant k \leqslant 50$, $k < \frac{v}{2}$, см. в работе Холла [9].

[2]) Эта матрица является матрицей инцидентности циклического симметрического (v, k, λ)-блокового устройства (Холл и Райзер [10]), в котором v элементов принадлежат v множествам по k элементов в каждом, так что каждый элемент принадлежит k различным множествам и каждая пара элементов принадлежит λ различным множествам. Отсюда следует, что каждое множество имеет ровно λ общих элементов с любым другим множеством. Необходимо выполняется условие $\lambda(v-1) = k(k-1)$ и, следовательно, λ выражается через v и k в виде $k(k-1)/(v-1)$. Отсюда также следует, что каждые два множества имеют ровно λ общих элементов.

ровно λ чисел a_{ij} равны 1. Таким образом, в каждой из рассматриваемых строк ровно $k-\lambda$ единиц не совпадают с единицами другой строки и, следовательно, эти строки матрицы инцидентности различны ровно в $2k-2\lambda$ позициях, т. е. расстояние между ними равно $2k-2\lambda$. Такой же результат должен иметь силу для дополнительной матрицы инцидентности, так как расстояние инвариантно относительно операции дополнения.

Лемма 2. *Расстояние каждой строки в A до своего дополнения равно v, а до любой другой строки в $\overline{A}$ равно $v-(2k-2\lambda)$.*

Доказательство. Очевидно, что расстояние каждой строки до своего дополнения равно v. Следовательно, лемма 2 вытекает из леммы 1.

Каждую строку матрицы инцидентности или ее дополнения можно представлять как кодовое слово $a_i = a_i^0, a_i^1, \ldots, a_i^{v-1}$, если положить для удобства $a_i^j \equiv a_{ij}$. С помощью лемм 1 и 2 кодовое расстояние полученного кода определяется следующей теоремой.

Теорема 1. *Если существует (v, k, λ)-разностное множество, то кодовое расстояние d' соответствующего кода $P'(v, d'; 2v)$ не превышает чисел $2k-2\lambda$ и $v-2k+2\lambda$. Кодовое расстояние d кода $P(v, d; 2v+2)$ является наименьшим из чисел k, $v-k$, $2k-2\lambda$ и $v-2k+2\lambda$.*

В качестве примера рассмотрим разностное множество 0, 1, 2, 4 (mod 7), для которого $v=7$, $k=4$, $\lambda=2$. Каждая разность встречается дважды ($1-0=2-1=1$, $2-0=4-2=2$ и т. д.). Соответствующая циклическая матрица инцидентности для разностных множеств $i, i+1, i+2, i+4$ (mod 7); $i=0,1, \ldots, 6$, является матрицей порядка 7 в левом верхнем углу табл. I и имеет кодовое расстояние $2k-2\lambda=4$. Ввиду того что $v-2k+2\lambda=3$, кодовое слово $a_0=1110100$, соответствующее разностному множеству 0, 1, 2, 4 (mod 7), порождает код $P(7, 3; 16)$ из табл. 1.

Аналогично дополнение 3, 5, 6 (mod 7), $v=7$, $k=3$, $\lambda=1$, данного разностного множества порождает тот же самый код; соответствующим кодовым словом 0001011 является $\bar{a}_0$. Далее, множество $v-d_1, v-d_2, \ldots, v-d_k$ (mod v),

обратное данному (v, k, λ)-разностному множеству $d_1, d_2, \ldots, d_k \pmod v$, является также (v, k, λ)-разностным множеством. С помощью кодовых слов 0010111 или 1101000 можно, рассматривая цифры $j=0, 1, \ldots, 6$ в обратном порядке, построить код $P(7,3; 16)$, обратный коду из табл. I.

III. Циклически перестановочные коды $P(4m-1, 2m-1; 8m)$

Различные классы циклически перестановочных кодов

$$P(4m-1,\ 2m-1;\ 8m)$$

рассматривались в литературе (Плоткин [14], Голомб [7]) или были получены из известных результатов (Брауэр [3], Холл [9]). Однако следующая теорема показывает, что кодовые слова в каждом таком коде соответствуют разностным множествам и, следовательно, эти коды могут быть найдены перебором разностных множеств.

Лемма 3. *Если дана циклическая матрица инцидентности порядка* v, *такая, что каждая строка содержит* k *единиц и расстояние между любыми двумя различными строками равно* 2μ, *то каждой строке соответствует* $(v, k, k-\mu)$*-разностное множество.*

Доказательство. Это утверждение обратно лемме 1. Если любые две строки матрицы инцидентности различаются ровно в 2μ позициях, то они содержат ровно $(2k-2\mu)/2 = k-\mu$ общих единиц. Следовательно, каждая разность $l = d_\alpha - d_\beta$ встречается $k-\mu$ раз в множестве целых чисел, для которых a_i^j есть единица, и это множество является $(v, k, k-\mu)$-разностным множеством, причем $\mu = k(v-k)/(v-1)$.

Теорема 2. *В любом коде* $P(4m-1, 2m-1; 8m)$ *каждому кодовому слову, отличному от* 0 *и* I, *соответствует разностное множество* $(4m-1, 2m, m)$ *или его дополнение* $(4m-1, 2m-1, m-1)$.

Доказательство. Код содержит 0, I, $4m-1$ кодовых слов a_i с $2m$ единицами и $4m-1$ дополнительных кодовых слов $\bar{a}_i$ с $2m-1$ единицами. Расстояние между любыми двумя кодовыми словами a_i одинаково и, следовательно, не меньше числа $2m$. Аналогичное утверждение справедливо для кодовых слов $\bar{a}_i$. Предположим теперь, что расстояние между a_α и a_β равно $2m+2\varepsilon$, где ε — некоторое неотрицательное целое число. Тогда расстояние между a_α и $\bar{a}_\beta$ равно $4m-1-2m-2\varepsilon=2m-1-2\varepsilon$. Отсюда следует, что $\varepsilon=0$, т. е. расстояние между любыми двумя кодовыми словами a_i равно $2m$. Мы оказываемся в условиях леммы 3, и каждому кодовому слову соответствует разностное множество $(4m-1, 2m, m)$ или $(4m-1, 2m-1, m-1)$ в зависимости от того, содержит ли это кодовое слово $2m$ или $2m-1$ единиц. (Между прочим, кодовым словам 0 и I отвечают соответственно разностные множества $(4m-1, 0, 0)$ и $(4m-1, 4m-1, 4m-1)$.) И обратно, если дано разностное множество, такое, что $k=2m$ или $k=2m-1$, то $2k-2\lambda=2m$ в обоих случаях. Число $2m-1$ равно наименьшему из чисел k, $v-k$, $2k-2\lambda$, $v-2k+2\lambda$ и в силу теоремы 1 является кодовым расстоянием.

Все известные результаты, приводящие к циклически перестановочным кодам $P(4m-1, 2m-1; 8m)$ собраны в табл. II. Символом Q в таблице отмечается существование *кода квадратичных вычетов* (Плоткин [14]), порождаемого $(4m-1, 2m-1, m-1)$-разностным множеством квадратичных вычетов по простому модулю $4m-1$. Хопф, Шур, Гильмен, Коксетер, Тодд и Пэлли, упомянутые Брауэром [3], рассматривали отдельные случаи матриц Адамара, причем работы первых двух относятся к 1920 году. Только первые два таких кода ($n=3, 7$) являются групповыми. В качестве примера построения этих кодов заметим, что квадратичные вычеты по модулю 7 образуют разностное множество 1, 2, 4 (mod 7), вновь приводящее нас к коду из табл. 1. Подобным образом нуль и квадратичные вычеты по модулю 7 образуют разностное множество 0, 1, 2, 4 (mod 7), приводящее к другому коду из табл. I.

Знаком R отмечается существование *рекурсивного кода*, порождаемого разностным множеством, соответствующим

максимальной последовательности неприводимого полинома степени r над $GF(2)$. (Голомб [7], Грин и Сан Суси [8]) [1]).

Рекурсивный код $P(2^r-1,\ 2^{r-1}-1;\ 2^{r+1})$ существует для каждого $r \geqslant 2$ ($m = 2^{r-2}$) и является групповым кодом. При $m=1$, 2 ($r=2$, 3) рекурсивные коды изоморфны рассмотренным выше соответствующим кодам квадратичных вычетов. Например, полиномиальному уравнению $x^3+x^2+1=0$ для $r=3$ соответствует максимальная последовательность 1110100 (т. е. $a_0^3+a_0^2+a_0^0 \equiv 0 \pmod 2$ и т. д.), отвечающая также разностному множеству 0, 1, 2, 4 (mod 7) и, следовательно, коду из табл. I.

Между прочим, последовательным умножением разностного множества можно найти все неприводимые полиномы степени r, если известен один из них. Для $r=5$, например, имеется шесть таких полиномов (вообще, имеется $\varphi(2^r-1)/r$ таких полиномов, где $\varphi(n)$ — функция Эйлера; Цирлер [18]). Разностному множеству (31, 16, 8) D_a: 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 16, 17, 19, 24, 25, 28 (mod 31) соответствует полином $x^5+x^4+x^3+x^2+1$. В результате умножения по модулю 31 каждого элемента множества D_a на 3 мы получаем множество D_b, соответствующее полиному x^5+x^2+1, а вторичное умножение на 3 приводит к множеству D_c, которое соответствует полиному $x^5+x^4+x^2+x+1$. Последующие умножения на 3 приводят к обратным трем разностным множествам и, следовательно, дают оставшиеся три неприводимых полинома степени 5 (т. е. $x^5+x^3+x^2+x+1$ и т. д.) [2]).

В случае произвольного r умножение на некоторое подходящее целое число часто дает все $\varphi(2^r-1)/r$ полиномов (как, например, при 2^r-1 простом и $2^r-1=15$ или 63), но в других случаях так можно получить половину

---

[1]) Этим разностным множествам соответствуют гиперплоскости в конечной дезарговой проективной геометрии (Холл [9]).

[2]) Любые $2r$ последовательных элементов (максимальной) последовательности, соответствующей разностному множеству ($4m-1$, $2m$, m), можно использовать для определения полинома. Это определение проще всего в случае, когда $r-1$ последовательных нулей используются в качестве исходной точки; каждый символ последовательно показывает наличие или отсутствие соответствующего члена в полиноме.

Таблица II

Различные циклически перестановочные коды $P(4m-1,\ 2m-1;\ 8m)$

m	$n=4m-1$	$d=2m-1$	N_G	$N=8m$	Циклические коды $P(n, d; N)$
1	3	1	8	8	$Q \leftrightarrow R \leftrightarrow T$
2	7	3	16	16	$Q \leftrightarrow R$
3	11	5	16	24	Q
4	15	7	32	32	$R \leftrightarrow T$
5	19	9	32	40	Q
6	23	11	32	48	Q
7	27	13	32	56	
8	31	15	64	64	Q^* $R \leftrightarrow S$
9	35	17	64	72	T
10	39	19	64	80	
11	43	21	64	88	Q * S
12	47	23	64	96	Q
13	51	25	64	104	
14	55	27	64	112	
15	59	29	64	120	Q
16	63	31	128	128	R^* U
32	127	63	256	256	Q * $R \leftrightarrow S$
64	255	127	512	512	R ($^*U$?)

N_G — число элементов соответствующего наилучшего группового кода $B(n, d; N_G)$; Q обозначает наличие кода квадратичных вычетов, R — рекурсивного кода, S — кода вычетов шестой степени, T — кода, определяемого парой простых чисел-близнецов, и U — наличие другого кода. Звездочкой указывается существование неизоморфных кодов, а стрелка показывает изоморфизм.

из них, а другая половина получается их обращением. (Заметим, что при $r>2$ не существует неприводимых полиномов, обратных самим себе.) Для $r=5$ преобразование $x \longrightarrow x^3$ можно осуществить непосредственно над полиномами. Например, $x^5+x^4+x^3+x^2+1$ переходит в $x^{15}+x^{12}+x^9+x^6+1$, для которого разложение на неприводимые множители дает $(x^5+x^2+1)(x^{10}+x^5+x^4+x^2+1)$. Искомый полином является первым сомножителем. В общем случае, однако, это преобразование

и редукцию к полиномам труднее использовать, чем разностные множества.

Символ S в табл. II означает наличие *кода вычетов шестой степени*, порождаемого разностным множеством вычетов шестой степени (Холл [9]), которое существует для простых $n=v=4l^2+27$. Эти коды не прибавляют ничего нового в нашем случае $n=4m-1$, но они, очевидно, отличаются от кодов квадратичных вычетов для тех же n и изоморфны рекурсивным кодам при $n=2^r-1$.

Символом T отмечается существование кода, порождаемого разностным множеством, для которого $v=pq=p(p+2)$, где p и $q=p+2$ являются простыми числами-близнецами (Брауэр [3]). Эти коды можно построить с помощью квадратичных вычетов по одному простому числу. Между прочим, Стэнтон и Спротт [15] показали существование разностных множеств над заданными парами, где $v=p^aq^b$ и числа p^a, $q^b=p^a+2$ являются степенями простых чисел. Если, однако, $a>1$ или $b>1$, то при этом используются элементы поля Галуа $GF(p^a)$ или $GF(q^b)$, которые обеспечивают сохранение разностей в полученных разностных множествах. Следовательно, не возникает циклической матрицы инцидентности, хотя нециклическую матрицу можно построить.

Символ U в табл. II означает негрупповой код, порождаемый разностным множеством [61, 31, 15]. Холл [9], в 1956 году указал это множество и заметил, что оно не является матрицей инцидентности проективной геометрии. Другие негрупповые коды могут существовать также для последующих 2^r-1, не являющихся простыми, например для 255.

В дополнение к символам Q, R, S, T и U, отмечающим различные коды, в табл. II показывается изоморфизм кодов и его отсутствие соответственно знаками ($\leftrightarrow$) и (*).

Рассмотренные выше случаи исчерпывают известные автору циклически перестановочные коды $P(4m-1, 2m-1; 8m)$. Исследования Холла [9] показали, что при $4m\leqslant 100$ нет других случаев. С помощью теоремы 2.1 Холла [9] понятия умножения разностных множеств и необходимого условия Човла и Райзера (см. Холл [9]) можно показать отсутствие прочих случаев и для $4m\leqslant 120$. Благодаря теореме 2 автора теорему Холла

и условие Човла—Райзера можно использовать для нахождения всех циклически перестановочных кодов $P(4m-1,\ 2m-1;\ 8m)$. Аналогично по каждому разностному множеству $(4m-1,\ 2m,\ m)$ строится оптимальный код $P_1(4m-1,\ 2m,\ 4m)$ из нулевого кодового слова 0 и циклических перестановок кодового слова a_0.

IV. Другие циклически перестановочные коды

Теорема 3. *В коде* $P'(4m+1,\ 2m;\ 8m+2)$ *любому кодовому слову соответствует разностное множество с* $v=4\binom{l+1}{2}+1=4m+1$, $k=l^2$, $\lambda=\binom{l}{2}$ *или его дополнение с* $k=(l+1)^2$, $\lambda=\binom{l+2}{2}$; *так что такой код существует только при* $m=\binom{l+1}{2}$.

Доказательство. Имеем $4m+1$ кодовых слов a_i, содержащих k единиц; причем расстояние между ними не менее чем $2m$; аналогичное утверждение справедливо для дополнительных кодовых слов $\bar{a}_i$, содержащих $4m+1-k$ единиц. Предположим, что расстояние между a_α и a_β равно $2m+2\varepsilon$; тогда расстояние между a_α и $\bar{a}_\beta$ равно $4m+1-2m-2\varepsilon=2m+1-2\varepsilon$. Таким образом, $\varepsilon=0$ и расстояние между любыми двумя кодовыми словами a_i равно $2m$. В силу леммы 3 каждому кодовому слову a_i соответствует разностное множество $(4m+1,\ k,\ k-m)$, а его дополнению $\bar{a}_i$ с $4m+1-k$ единицами соответствует дополнительное разностное множество

$$(4m+1,\ v-k,\ v-k-m).$$

Остается определить вид числа k. В силу леммы 1 имеем $2k-2\lambda=2m$. Из равенства $\lambda=k(k-1)/(v-1)$ следует равенство $k-m=k(k-1)/4m$, равносильное равенству $k^2-(4m+1)k+4m^2=0$. Корни $k_{1,2}=\frac{1}{2}(4m+1\pm\sqrt{8m+1})$ этого уравнения являются целыми числами, если $8m+1$ есть $(2l+1)^2$, т. е. квадрат некоторого целого нечетного числа. Итак,

$$v=4m+1=2l^2+2l+1=4\binom{l+1}{2}+1,$$

откуда

$$m = \binom{l+1}{2}.$$

Два решения приводят к

$$k = l^2,\ \lambda = \frac{l^2-l}{2} = \binom{l}{2}$$

и

$$k = (l+1)^2,\ \lambda = \frac{l^2+3l+2}{2} = \binom{l+2}{2}.$$

Для $l=1$ и $l=2$ разностные множества и коды, рассматриваемые в теореме 3, существуют. Для $l=1$ разностное множество $(v, k, \lambda)=(5, 1, 0)$ приводит к коду $P'(5, 2; 10)$; для $l=2$ разностное множество 0, 1, 3, 9 (mod 13) с $(v, k, \lambda)=(13, 4, 1)$ приводит к коду $P'(13, 6; 26)$. Полученные коды содержат на 6 кодовых слов меньше, чем соответствующие оптимальный код $A(5, 2; 16)$ и наилучший известный код $A(13, 6; 32)$, построенный Стивенсом и Боурициусом [16]. Для $l=3, 4, 5, 6, 7$, т. е. для $4m+1<145$ из результатов Холла [9] следует отсутствие таких разностных множеств, как (25, 9, 3), (41, 16, 6) и т. д. Следовательно, для $m=6, 10, 15, 21, 28$ соответствующие коды $P'(4m+1, 2m; 8m+2)$ не существуют. Для $l=8, 9$, т. е. для $4m+1<221$, автором показано отсутствие разностных множеств. Поэтому коды этого класса рассматриваются реже, чем коды из предыдущего раздела.

Теорема 4. *В коде* $P'(4m, 2m; 8m)$ *любому кодовому слову соответствует разностное множество с* $v=4l^2$, $k=\binom{2l}{2}$, $\lambda=\binom{l}{2}$ *или его дополнение; так что такие коды существуют только для* $m=l^2$.

Доказательство. Аналогично доказательству теоремы 3 разностное множество имеет вид $(4m, k, k-m)$. Из леммы 1 следует равенство $k-m=k(k-1)/(4m-1)$, откуда $k_{1,2}=2m\pm\sqrt{m}$, т. е. $m=l^2$.

Такой код $P'(4, 2; 8)$ существует для $m=l^2=1$, но не существует для $l=2, 3, 4, 5$. Автор не рассматривает случай $l\geqslant 6$, так как уже для $l=6$ неизвестно, существует ли разностное множество (144, 66, 30) и код P'

(144, 72; 288). Заметим, что во всех этих случаях выполняется необходимое условие Човла—Райзера существования разностных множеств (Холл [9]); именно, число $k-\lambda$ является квадратом (в нашем случае это l^2). Конечно, все такие коды оптимальны.

Для полноты можно добавить, что каждому кодовому слову в коде $P'(4m+2,\ 2m;\ 8m+4)$ соответствует разностное множество и показать аналогичным анализом, что $3m+1$ является квадратом целого числа. Если применить необходимое условие Човла—Райзера, то увидим, что m является квадратом биномиального коэффициента $\binom{2l+2}{l}$. В первых нескольких случаях ситуация такова: для $l=0$, $m=1$ существует разностное множество (6, 1, 0), но получается плохой код P' (6, 2; 12); для $l=1$, $m=16$ разностное множество (66, 26, 10) не существует, как показал Холл [9]; случаи $l=2$, $m=225$ и $l=3$, $m=3136$ не исследованы.

С помощью теоремы 1 можно также рассматривать произвольные разностные множества и получаемые из них коды. В некоторых случаях эти коды почти оптимальны. Например, разностное множество (40, 13, 4) приводит к коду P' (40, 18, 80).

Можно также использовать циклические матрицы инцидентности, которые не соответствуют разностным множествам (и циклическим сбалансированным блок-схемам). Рассмотрим, например, квадратичные вычеты по модулю 13 Q_0: 1, 3, 4, 9, 10, 12 ($v=13$, $k=6$). Среди разностей $l=d_\alpha-d_\beta$ ($l\in Q_0$) шесть встречаются по два раза каждая, а шесть оставшихся встречаются по три раза. В соответствующей циклической матрице инцидентности, которая отвечает частично сбалансированной блок-схеме (Боуз и Шимамото [1]), расстояние каждой строки до любой из некоторых шести строк равно 8 ($=2k-2\lambda_1=12-4$), а расстояние до любой из остальных шести строк равно 6 ($=2k-2\lambda_2=12-6$) [1]). Следовательно, наименьшее {рас-

[1]) Из формулы $\lambda=k(k-1)/(v-1)$ получается число $\lambda=2{,}5$, которое хотя и не является целым числом, но показывает природу множества. Соответствующая циклическая блок-схема известна как симметрическая частично сбалансированная блок-схема с двумя ассоциированными классами.

стояние между строкой матрицы и строкой ее дополнения равно

$$5\ (=v-2k+2\lambda_{\min}=13-8)$$

и получается код P' (13, 5; 26). Добавление кодовых слов 0 и I не меняет расстояния и приводит к коду P (13, 5; 28). Этот код, однако, далек от оптимального кода A (13, 5, 64) (Стивенс и Боурициус [16]). Вообще, с помощью квадратичных вычетов по любому простому модулю $4m+1$ можно получать циклические коды $P(4m+1,\ 2m-1;\ 8m+4)$ (Цзянь [4]), но эти коды не оптимальны. Другие аналогичные коды также не оптимальны. Например, для $4m+1=9$ код P (9, 3; 20) получается из множества целых чисел 0, 1, 2, 4 (mod 9), циклическая матрица инцидентности для которых соответствует частично сбалансированной блок-схеме $\left(9,\ 4,\ 1\frac{1}{2}\right)$. Заметим, однако, что этот код не является оптимальным как в смысле числа кодовых слов N, так и в смысле расстояния d; существуют коды A (9, 3; 38) и A (9, 4; 20) благодаря Голею. Между прочим, код P (9, 4; 20) не существует в силу теоремы 3, хотя код P (9, 4; 19) можно построить (Стивенс и Боурициус [16]) из кодового слова 000000000 и циклических перестановок кодовых слов 111010000 и 111100110.

V. Родственные коды

В заключение надо заметить, что результаты теорем 1, 3 и 4 можно распространить на произвольные (нециклические) симметрические блок-схемы с заданными v, k, λ (см., например, Коннор [6]). Однако это обобщение является не таким же сильным, как в рассмотренной выше теореме 1 Боуза и Шрикханде, показывающей одновременное существование кодов A ($4m-1$, $2m-1$; $8m$), A ($4m$, $2m$; $8m$), блок-схем и матриц Адамара.

Если дана симметрическая блок-схема (v, k, λ), то можно получить код D' (v, d'; $2v$), рассматривая строки матрицы инцидентности и их дополнения как кодовые слова. Как и в теореме 1, расстояние d' не превышает каждое из чисел $2k-2\lambda$ и $v-2k+2\lambda$.

По отношению к кодам $A(4m+1, 2m; 8m+2)$ теорема 3 обобщается следующим образом: только симметрические блок-схемы с

$$v=4\binom{l+1}{2}+1,\ k=l^2,\ \lambda=\binom{l}{2}$$

или их дополнения приводят к кодам такого вида, содержащим дополнение каждого кодового слова. Например, для $l=3$ нециклическая блок-схема (25, 9, 3) существует (и принадлежат Бхаттачария; см. Коннор [6]) и порождает код $D'(25, 12; 50)$, кодовыми словами которого служат строки матрицы инцидентности и их дополнения. Однако этот результат не аналогичен теореме Боуза и Штрикандера: хотя код $A(9, 4, 18)$ существует, для него не имеется соответствующей симметрической блок-схемы $(9, k, \lambda)$.

По отношению к кодам $A(4m, 2m; 8m)$ теорема 4 обобщается следующим образом: только симметрические блок-схемы с

$$v=4l^2,\ k=\binom{2l}{l},\ \lambda=2\binom{l}{2}$$

или их дополнения приводят к кодам $D'(4m, 2m; 8m)$, содержащим дополнение каждого кодового слова. Например, для $l=2$ нециклическая блок-схема (16, 6, 2) существует и приводит к коду $A(16, 8; 32)$, в котором каждое кодовое слово содержит 6 или 10 единиц. Конечно, многие коды $A(4m, 2m; 8m)$ можно получить из блок-схемы Адамара с $v=4m-1$, если добавить к ним кодовые слова 0 и I и проверочный разряд.

VI. Заключение

В другой работе автор исследует простоту кодирования и декодирования циклически перестановочными кодами, используя большей частью критерий максимального правдоподобия, а не метод проверки на четность. Простоту циклически перестановочных кодов можно использовать для увеличения эффективности до оптимальной или почти оптимальной при небольших n. Кроме того, Грин и Сан Суси [8] рассмотрели способ кодирования и декодирования методом проверки на четность, который применим ко всем рекурсивным кодам.

Хотя циклические коды рассмотрены здесь в терминах систем передачи информации, их можно применять в устройствах матриц распределения нагрузок переключательных цепей (Цзянь [5]; Нейман, [12]). В этих приложениях свойства расстояния используются так, что с малыми возбуждениями на входе задающего аппарата можно отобрать один выход, в то время как на всех других сетка возбуждения есть эффективный нуль; расстояние кода допускает определенное изменение в задающих возбуждениях. Циклические структуры, рассматриваемые в таком аспекте, должны облегчить конструирование, инспекцию и эксплуатацию матриц переключательных цепей [1]).

Автор хочет поблагодарить Гроули и Тэйга за несколько стимулирующих бесед и полезных идей. Он также благодарен Рутледжу из ИБМ, который указал ему работу Стивенса и Боурициуса [16], приведенную в литературе к этой заметке.

ЛИТЕРАТУРА

1. Bose R. C., Shimamoto T., Classification and analysis of partially balanced incomplete block design with two associate classes, *J. Am. Statist. Assoc.*, **47** (1952), 151—184.
2. Bose R. C., Shrikhande S. S., A note on a result in the theory of code construction, *Inform. and Control*, **2** (1959), 183—194.
3. Brauer A., On a new class of Hadamard determinants, *Math. Z.*, **58** (1953), 219—225.
4. Chien R. T., Orthogonal matrices, error-correcting codes and load-sharing matrix switches (Letter), *IRE Trans. on Electronic Computers*, **EC-8** (1959), 400.
5. Chien R. T., A class of optimal noiseless loud-sharing matrix switches (Letter), *IBM J. Research Develop.*, **4** (1960), 414—417.
6. Connor W. S., Jr., On the structure of balanced incomplete block designs, *Ann. Math. Statist.*, **23** (1952), 57—71.
7. Golomb S. W., Sequences with randomness properties, Glenn L. Martin Co., Baltimore, Maryland (1955), Final Report on Contract No. W36-039 SC-54-36611.
8. Green J. H., Jr., San Soucie R. L., An error-correcting encoder and decoder of high efficiency, *Proc IRE*, **46** (1958), 1741—1744.

[1]) По этим вопросам см. статью: Собельман В. И., О геометрии прошивок ферритовых матриц, сб. „Проблемы кибернетики" № 6, Физматгиз, 1959, стр. 7—38 и библиографию к ней.— *Прим. перев.*

9. Hall M., Jr., A survey of difference sets. *Proc. Am. Math. Soc.*, 7 (1956), 975—986.
10. Hall M., Reser H., Cyclic incidence motrices. *Can. J. Math.*, 3 (1951), 495—502.
11. Neumann P. G., Encoding and decoding for cyclic permutation codes (1962).
12. Neumann P. G., On the logical design of noiseless loud-sharing matrix switches (1962).
13. Peterson W. W., Error-Correcting Codes, M. I. T. Press and Wiley, New York, 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]
14. Plotkin M., Binary codes with specified minimum distance, *IRE Trans. on Inform. Theory*, **IT-6** (1960), 445—450.
15. Stanton R. G., Sprott D. A., A family of difference sets, *Can. J. Math.*, **10** (1958), 73—77.
16. Stevens R. F., Bouricius W. G., The heuristic generation of large error-correcting codes, IBM Research Memorandum RC-123 (1959).
17. Takahasi H., Goto E., Application of error-correcting codes to multi-way switching. First. Intern. Conf. on Inform. Processing, Paris, Oldenbourg, Munich, and Butterworth, London (1959).
18. Zierler N., Several binary sequence generators, M. I. T. Lincoln Laboratory Technical Report 95, Lexington, Massachusetts (1955).

ОБ ОПТИМАЛЬНЫХ КОДАХ, ИСПРАВЛЯЮЩИХ ПАКЕТЫ ОШИБОК [1])

В. Элспас, Р. Шорт

В статье подробно исследован класс систематических двоичных кодов, которые исправляют пакеты ошибок, характерные для некоторых цифровых каналов. Эти коды, являющиеся обобщением кодов, найденных Абрамсоном и Меласом, представляют собой циклические коды, которые исправляют любые одиночные пакеты ошибок длиной до l знаков и любые циклические перестановки таких пакетов ошибок в n-значном кодовом слове. Они являются оптимальными в том смысле, что содержат минимальное количество избыточных знаков, которое теоретически возможно в циклическом коде при данных значениях n и l.

Циклический код полностью определяется порождающим полиномом $g(x)$. Поэтому свойства циклического кода можно установить на основе анализа соответствующего $g(x)$. Сформулированы необходимые и достаточные условия, которым должен удовлетворять $g(x)$ для того, чтобы порождаемый им циклический код был оптимальным в отношении исправления пакетов ошибок длиной до l. Эти условия сформулированы в порядке, который соответствует порядку проверок, позволяющих установить, является ли соответствующий код оптимальным. Такие проверки могут быть проведены (в принципе) для любого $g(x)$. С помощью указанных проверок были найдены все оптимальные коды, исправляющие пакеты ошибок длиной до l для $n < 2^{12}$ и $l < 6$. Порождающие полиномы этих кодов сведены в таблицы, приведенные в статье. Всего в таблицах перечислено 98 кодов (не считая обратных) для $l=3$ и $l=4$. Показано, что для $l=5$ в пределах указанных длин оптимальных кодов не существует. Для $l \geqslant 6$ на практике будут, по-видимому, применяться неоптимальные коды — оптимальные должны иметь слишком большую длину.

Введение

В настоящей статье приведены результаты широкого поиска оптимальных циклических кодов, рассчитанных на исправление пакетов ошибок, длина которых лежит

[1]) Elspas B., Short R. A., A note on optimum bursterror-correcting codes, *IRE Trans. on Information Theory*, **IT-8** (1962), № 1, 39—42.

в заданных пределах. Максимальная длина пакета, которая рассматривалась при описываемом поиске, составляла $l=5$ знаков, а максимальная длина кодового слова $n=4095$ знаков. Однако методы, которые были использованы при этом поиске, применимы и в общем случае. Всего было найдено 98 различных кодов, исправляющих пакеты ошибок длиной до $l=3$ и 4. Показано, что для длин кодового слова, не превышающих 4095 знаков, и $l=5$ оптимальных кодов (т. е. кодов, содержащих не более 16 проверочных знаков) не существует.

Циклический код [1]) полностью определяется порождающим полиномом $g(x)$. Степень этого полинома равна количеству проверочных знаков, а его период—длине n соответствующего кода (речь идет о полной длине; укороченные циклические коды, которые были исследованы Файром [2], Риджером [3] и др., в настоящей работе не рассматриваются).

Было показано [4], что для того, чтобы двоичный $(n, n-r)$ код с r проверочными знаками исправлял любые пакеты ошибок длиной до l, должно выполняться следующее неравенство:

$$n \leqslant 2^{r-l+1}-1. \qquad (1)$$

Отметим, что в настоящей работе, говоря о каком-либо сочетании ошибок, мы имеем в виду все его циклические перестановки. Для того чтобы сочетание ошибок считалось исправляемым, код должен исправлять все n циклических перестановок этого сочетания ошибок.

Коды, для которых в соотношении (1) имеет место знак равенства, обладают наибольшей длиной кодового слова. Следовательно, эти коды имеют максимальное для заданных значений длины пакета l и числа проверочных знаков r количество информационных знаков $k=n-r$. В таких кодах 2^r возможных состояний корректора [2]) используются для исправления пакетов ошибок (с учетом всех их циклических перестановок) наилучшим теорети-

[1]) Общий анализ циклических кодов и относящиеся к ним определения можно найти, например, в (1).

[2]) По терминологии [1]—2^r различных синдромов.— *Прим. перев.*

чески возможным способом. Из 2^r возможных состояний корректора для исправления пакетов не используется только $2^{l-1}-1$ состояний. Все остальные состояния корректора используются для опознания $1+n2^{l-1}$ различных сочетаний (векторов) ошибок [1]). Коды с такими свойствами Абрамсон [4] назвал оптимальными, а Питерсон ([1], стр. 186—187) — кодами с минимальной избыточностью. В настоящей работе используется первое из этих названий. Из сказанного выше следует, что коды, являющиеся оптимальными для исправления пакетов ошибок, в известном смысле аналогичны плотноупакованным кодам для исправления независимых ошибок.

Коды для случаев $l=1$ и 2, которые были подробно исследованы Абрамсоном [5], [6], соответствуют циклическим кодам Хэмминга. Эти коды для исправления пакетов ошибок являются оптимальными. Они существуют для всех достаточно больших значений r ($r>1$ для $l=1$ и $r>3$ для $l=2$). Кроме того, Мэлас [7] и Абрамсон [4] привели несколько отдельных примеров оптимальных кодов, исправляющих пакеты ошибок, для $l=3$. Настоящая работа имеет целью выяснить, насколько широк класс оптимальных кодов, исправляющих пакеты ошибок, для случаев, когда (расчетная) длина пакета равна 3 и больше, а также перечислить по возможности все представляющие практический интерес оптимальные коды, длина которых не превышает некоторого значения. Длины блока, превышающие несколько тысяч двоичных знаков, были признаны слишком большими для практического использования. За максимальное значение l при настоящем поиске было принято значение $l=5$, так как уже при $l=6$ требуется слишком большой объем вычислительной работы.

---

[1]) Используя результаты, приведенные ниже в настоящей статье, можно показать, что $n>2^{l-1}-1$. Отсюда количество неиспользуемых состояний корректора во всех случаях оказывается недостаточным для исправления какого-либо другого сочетания ошибок при всех n возможных циклических размещениях его в кодовом слове. Однако при небольшом усложнении схемы эти состояния корректора могут быть использованы для обнаружения некоторых сочетаний ошибок.

Необходимые условия оптимальности кода, исправляющего пакеты ошибок длиной до l

Свойства циклического кода, порождаемого полиномом $g(x)$, можно связать со свойствами множества циклов [1]) линейного регистра сдвига с обратной связью, характеристическим полиномом которого является $g(x)$. В самом деле, именно такого рода линейные цепи последовательного действия положены в основу одного из способов реализации циклических кодов [1]. Как указано Абрамсоном [4], для оптимальности кода, исправляющего пакеты ошибок длиной до l, необходимо, чтобы множество циклов, соответствующих порождающему полиному $g(x)$, содержало бы 2^{l-1} циклов длины $n=2^{r-l+1}-1$ [2]) и цикл единичной длины (содержащий только одно состояние корректора — когда корректор состоит из одних нулей); остальные $2^{l-1}-1$ из 2^r возможных состояний корректора могут образовывать любые циклы.

Легко видеть [3]), что код, исправляющий любые пакеты ошибок длиной до l, должен содержать не менее $r_1=2l$ проверочных знаков (независимо от того, является ли он оптимальным, или нет). В то же время из (1) следует, что оптимальный код, исправляющий пакеты ошибок длиной до l, должен содержать самое большее $r_2=2\,[\log_2^r(n+1)-1]$ проверочных знаков. Таким образом, для оптимального кода, исправляющего пакеты ошибок длиной до l,

$$2l \leqslant r \leqslant 2\,[\log_2(n+1)-1]. \qquad (2)$$

Кроме того, для того чтобы циклический код, порождаемый полиномом $g(x)$, был оптимальным для исправления пакетов ошибок длиной до l, необходимо выполнение еще следующих трех условий (приводимых здесь без доказательства) [4]).

---

[1]) Общий анализ линейных цепей последовательного действия и регистров сдвига с обратными связями, в том числе анализ множества их циклов, можно найти в [8].

[2]) То есть 2^{l-1} циклов, каждый из которых состоит из n различных состояний корректора. — *Прим. перев.*

[3]) [1], гл. 4, стр. 61, теорема 4.8.

[4]) Доказательство того, что эти условия являются необходимыми, можно найти в [9].

1) $g(x)=g_1(x)\,g_2(x)\,\ldots\,g_t(x)$, где $g_i(x)$ — различные неприводимые полиномы степеней r_i, причем $r=\sum r_i$.
2) Все r_i должны быть делителями $m=r-l+1$.
3) Одно и только одно из чисел r_i должно равняться m, причем соответствующий $g_i(x)$ должен быть примитивным неприводимым полиномом.

Слабую оценку наименьшего возможного значения длины блока оптимального кода, исправляющего пакеты ошибок длиной до l, дает неравенство

$$n_{\text{опт}} \geqslant 2^{l+1}-1, \tag{3}$$

которое является прямым следствием соотношения (1) и неравенства $r \geqslant 2l$.

Приведенные выше условия были использованы для ограничения класса полиномов, подлежащих проверке на оптимальность порождаемых ими кодов. Если выполняются условия 1)—3), то множество циклов, которым обладает $g(x)$, удовлетворяет необходимым условиям оптимальности кода, исправляющего пакеты ошибок длиной до l. Если при этом все 2^{l-1} подлежащих исправлению сочетаний ошибок (длиной до l) порождают различные циклы и все эти циклы имеют длину n, то соответствующий код действительно является оптимальным для исправления пакетов ошибок длиной до l. Таким образом, поиск оптимальных кодов может быть подчинен следующей системе: сначала $g(x)$ разлагают на множители и проверяют выполнение условий 1)—3), а затем для отобранных полиномов проверяют выполнение условия различия циклов [1]).

Подробное исследование возможностей построения оптимальных кодов для $l=3$, 4 и 5 показывает, что необходимым условиям в отношении структуры множества циклов удовлетворяют только полиномы, приведенные в табл. 1.

---

[1]) Из условий 1)—3) и соотношения (2) следует, что если сочетания ошибок имеют длину, меньшую $l+1$, то порождаемые ими циклы имеют длину n (а не меньше).

Таблица I

Длина пакета	Вид $g(x)$	Условия в отношении m (степени f) *
3	$(1+x+x^2)f(x)$	$2 \mid m$ и $m \geqslant 4$
4	A) $(1+x+x^3)f(x)$	$3 \mid m$ и $m \geqslant 6$
	B) $(1+x)(1+x+x^2)f(x)$	$2 \mid m$ и $m \geqslant 6$
5	A) $(1+x)(1+x^4)f(x)$	$4 \mid m$ и $m \geqslant 8$
	B) $(1+x)(1+x+x^3)f(x)$	$3 \mid m$ и $m \geqslant 6$
	C) $(1+x+x^2+x^3+x^4)f(x)$	$4 \mid m$ и $m \geqslant 8$

Во всех случаях $f(x)$ — примитивный неприводимый полином степени $m=r-l+1$, т. е. период $f(x)$ равен $n=2^m-1$.

* Запись $2 \mid m$ следует читать: „2 является делителем m".

Оптимальные коды, исправляющие пакеты ошибок длиной до 3

Как показано Абрамсоном [4], условием того, что 4 сочетания ошибок x, xx, xox и xxx принадлежат четырем различным циклам полинома $g(x)=(1+x+x^2)f(x)$, является справедливость соотношения

$$1+x \equiv x^a \bmod f(x), \text{ где } a \not\equiv 2 \pmod 3. \qquad (4)$$

Нами были составлены таблицы показателей s степеней полиномов, определяемых соотношением

$$p(x) \equiv x^s \bmod \varphi(x), \quad 0 \leqslant s < n, \qquad (5)$$

где n период φ, для всех неприводимых $p(x)$, степень которых меньше 6, и для всех примитивных неприводимых $\varphi(x)$, степень которых меньше 13 (за исключением степени 11). Располагая такими таблицами показателей степеней, можно табулировать все удовлетворяющие условию (4) полиномы $f(x)$ вплоть до 12-й степени. Эти полиномы, а также соответствующие им $g(x)$ приведены в табл. II (полиномы $g(x)$ записаны в этой таблице в виде произведения $g(x)=(1+x+x^2)f(x)$. Для каждого $g(x)$ существует „обратный" полином $g^*(x)=x^r g\left(\frac{1}{x}\right)$,

Таблица II

Порождающие полиномы для оптимальных кодов, исправляющих пакеты ошибок длиной до 3

n	r	$g(x)$ (для каждого сомножителя перечислены степени, коэффициент при которых равен единице)	
15	6	(012) (014)	
63	8	(012) (016)	(012) (01256)
255	10	(012) (01278)	(012) (02358)
		(012) (01358)	(012) (0123678)
		(012) (02348)	
1023	12	(012) (0134,10)	(012) (0158,10)
		(012) (0125,10)	(012) (023458,10)
		(012) (0235,10)	(012) (0168,10)
		(012) (0256,10)	(012) (012468,10)
		(012) (0137,10)	(012) (012478,10)
		(012) (0267,10)	(012) (01245678,10)
		(012) (012467,10)	(012) (0149,10)
		(012) (012567,10)	(012) (012369,10)
		(012) (01234567,10)	(012) (013469,10)
		(012) (0238,10)	(012) (012489,10)
4095	14	(012) (0356,12)	(012) (02346,10,12)
		(012) (013456,12)	(012) (01257,10,12)
		(012) (023456,12)	(012) (02357,10,12)
		(012) (0347,12)	(012) (01467,10,12)
		(012) (0467,12)	(012) (01238,10,12)
		(012) (012348,12)	(012) (02348,10,12)
		(012) (0158,12)	(012) (01478,10,12)
		(012) (013458,12)	(012) (0234678,10,12)
		(012) (013568,12)	(012) (0135678,10,12)
		(012) (014578,12)	(012) (01269,10,12)
		(012) (034678,12)	(012) (0234579,10,12)
		(012) (01234678,12)	(012) (0145789,10,12)
		(012) (012459,12)	(012) (01246,11,12)
		(012) (023459,12)	(012) (01348,11,12)
			(012) (0134568,11,12)
		(012) (01234569,12)	(012) (01378,11,12)

Продолжение табл. II

n	r	$g(x)$ (для каждого сомножителя перечислены степени, коэффициент при которых равен единице)	
4095	14	(012) (034579,12)	(012) (0125678,11,12)
		(012) (012389,12)	(012) (0123489,11,12)
			(012) (0124689,11,12)
		(012) (034589,12)	(012) (0125789 11,12)
		(012) (01234589,12)	(012) (0124,10,11,12)
		(012) (023689,12)	(012) (0125,10,11,12)
		(012) (015789,12)	(012) (012347,10,11,12)
		(012) (012,10,12)	(012) (012367,10,11,12)
		(012) (01236,10,12)	

который также удовлетворяет всем изложенным выше условиям. Однако, поскольку коды, соответствующие $g^*(x)$, отличаются от кодов, соответствующих $g(x)$, только обратным расположением знаков, то из каждой пары взаимно обратных полиномов в таблице приведен только один. Таким образом, в табл. II перечислены порождающие полиномы для всех оптимальных кодов, исправляющих пакеты ошибок длиной до 3, у которых число проверочных знаков не превышает 14, а длина блока не превышает 4095. Весьма вероятно, что аналогичные коды существуют для любой четной степени r, превышающей 4, однако доказать это не удалось [1]). Если требуется иметь код, длина которого превышает 4095, то для его нахождения (если только он существует) может быть применена такая же методика [2]).

---

[1]) Для доказательства необходимо показать, что в любом поле Галуа $GF(2^n)$, где n — четное число, большее 4, существует примитивный элемент x, который удовлетворяет уравнению $1+x=x^a$ при $a \not\equiv 2 \pmod 3$.

[2]) Рецензент обратил наше внимание на то, что аналогичная работа по кодам, исправляющим пакеты ошибок длиной до 3, была проведена в Северокаролинском университете Гроссом, Боузом и Чакраварти.

Оптимальные коды, исправляющие пакеты ошибок длиной до 4

Два вида порождающих полиномов, при которых соответствующий код будет оптимальным для исправления пакетов ошибок длиной до 4, приведены в таблице I (как и в ранее рассмотренном случае, обратные полиномы в этот список не включены). Можно вывести $\binom{8}{2} = 28$ условий, при соблюдении которых все восемь подлежащих исправлению сочетаний ошибок принадлежат различным циклам. В число этих условий входят требования в отношении показателей степеней неприводимых полиномов, соответствующих сочетаниям ошибок *xx*, *xxx*, *xoxx* и *xxox*, по модулю $f(x)$. Таким образом, a, b, c и d определяются однозначно из соотношений

$$\begin{aligned} 1+x &\equiv x^a \bmod f(x), \\ 1+x+x^2 &\equiv x^b \bmod f(x), \\ 1+x^2+x^3 &\equiv x^c \bmod f(x), \\ 1+x+x^3 &\equiv x^d \bmod f(x), \end{aligned} \tag{6}$$

где $0 \leqslant a, b, c, d <$ периода $f(x)$.

В случае 4A) упомянутые 28 условий сводятся к 12 независимым соотношениям по модулю 7, в которые входят a, b и c. Аналогичным образом, в случае 4B) эти условия сводятся к 4 независимым соотношениям по модулю 3, в которые входят a, b, c и d.

Системы соотношений, к которым сводятся упомянутые 28 условий, могут быть использованы для выявления допустимого класса полиномов $f(x)$ с помощью таблицы показателей степеней полиномов, о которой говорилось выше. Оказывается, что для $m=6$ ни один из (шести) примитивных неприводимых полиномов шестой степени не удовлетворяет всем необходимым условиям ни в случае 4A), ни в случае 4B). То же относится и к 16 примитивным неприводимым полиномам восьмой степени. Наименьшим значением m, при котором порождающий полином удовлетворяет всем поставленным требованиям, является $m=9$ для случая

4А). Этот полином

$$g(x) = (1+x+x^3)(1+x+x^2+x^3+x^6+x^7+x^9) = \\ = 1+x^3+x^5+x^8+x^{12}$$

порождает циклический код длины $n=511$ с $k=n-r=499$ информационными знаками, который исправляет все пакеты ошибок длиной до 4. Остальные порождающие полиномы, приведенные в таблице III, соответствуют кодам с $r=13$, $n=1023$ (пять различных кодов для случая 4В)) и с $r=15$, $n=4095$ (четыре различных кода для случая 4А) и 13 различных кодов для случая 4В)).

Проверка того, что 98 оптимальных кодов, которые приведены в таблицах II и III, действительно исправляют любые пакеты ошибок длиной до 3 и 4 соответственно, была проведена на специализированном цифровом моделирующем устройстве. Это устройство было запрограм-

Таблица III

Порождающие полиномы для оптимальных кодов, исправляющих пакеты ошибок длиной до 4

n	r	$g(x)$ (для каждого сомножителя перечислены степени, коэффициент при которых равен единице)	
511	12	(013) (0123679)	
1023	13	(01) (012) (0234, 10)	(01) (012) (012369, 10)
		(01) (012) (012467, 10)	(01) (012) (013469, 10)
		(01) (012) (0123458, 10)	
4095	15	(013) (012389, 12)	(01) (012) (02348, 10, 12)
		(013) (012), 10, 11, 12)	(01) (012) (01478, 10, 12)
		(013) (012347, 10, 11, 12)	(01) (012) (0234678, 10, 12)
		(013) (039, 10, 12)	(01) (012) (01269, 10, 12)
		(01) (012) (0347, 12)	(01) (012) (01246, 11, 12)
		(01) (012) (012459, 12)	(01) (012) (0125789, 11, 12)
		(01) (012) (034589, 12)	(01) (012) (0124, 10, 11, 12)
		(01) (012) (01234589, 12)	(01) (012) (012347, 10, 11, 12)
		(01) (012) (02346, 10, 12)	

мировано таким образом, что оно выполняло вычислительные операции в поле Галуа [1]).

Некоторые из приведенных в таблице III кодов были независимо от авторов найдены Кином из Лаборатории электроники Станфордского университета [2]). Авторы признательны ему за это подтверждение части их работы.

Оптимальные коды, исправляющие пакеты ошибок длиной до 5

Три возможных вида порождающих полиномов, при которых соответствующий код будет оптимальным для исправления пакетов ошибок длиной до 5, приведены в табл. I (случаи (5A)—(5C)). Существует $\binom{16}{2} = 120$ условий, при выполнении которых все 16 сочетаний ошибок, подлежащих исправлению, принадлежат различным циклам. В число этих условий, помимо требований в отношении показателей степеней полиномов a, b, c и d, которые были изложены выше, входят требования в отношении показателей степеней e, f и g, соответствующих сочетаниям ошибок *xxoox*, *xooxx* и *xxxxx*. Эти 120 различных соотношений для случая (5A) могут быть сведены к 64, а для случая (5B)—к 28 независимым соотношениям. Для случая (5C) необходим несколько другой прием. Укажем только, что при проверке отобранных полиномов $f(x)$ оказалось, что не существует ни одного примитивного неприводимого полинома $f(x)$ степени 12 или меньше, который удовлетворял бы всем этим условиям. Отсюда следует, что не существует оптимальных кодов для исправления пакетов ошибок длиной до 5, которые имели бы 16 или менее проверочных знаков, т. е. длина блока которых составляла бы 4095 или менее.

---

[1]) [1], глава 7, стр. 115.

[2]) Сведения об этих кодах приведены в [10].

Коды, исправляющие более длинные пакеты ошибок

Для $l>5$ описанная выше методика поиска становится слишком громоздкой, чтобы ее можно было выполнять путем ручных вычислений. Например, один из видов полиномов, порождающих оптимальный код для исправления пакетов ошибок длиной до $l=6$, должен удовлетворять 332 различным условиям. Некоторые интересные результаты можно получить в том случае, если использовать для поиска соответствующим образом запрограммированную вычислительную машину. Однако авторы полагают, что при $l>5$ длина оптимальных кодов настолько велика, что практического интереса они не представляют[1]). Достаточно сказать, что объем буферной памяти декодирующего устройства должен равняться полной длине кодового слова, т. е. n знакам. Теоретическая нижняя граница длины оптимального кода для исправления пакетов ошибок длиной до l, определяемая соотношением (3), составляет $2^{l+1}-1$. Однако по крайней мере при $l=4$ и 5 наименьшая длина оптимальных кодов значительно превышает эту нижнюю границу. Другим доводом против использования столь длинных кодов является то положение, что с ростом длины кодового слова возрастает вероятность его поражения сразу несколькими пакетами ошибок.

В связи с этим возникает вопрос, насколько хорошим может быть неоптимальный код, исправляющий пакеты ошибок. Коды Файра обеспечивают исправление пакетов ошибок длиной до l при наличии $r=3l-1$ проверочных знаков. При этом их длина может меняться в широких пределах. Стандфордский научно-исследовательский институт в настоящее время исследует вопрос о возможности построения кода, число знаков которого было бы значительно меньше $3l-1$ и который в то же время не был бы чрезмерно длинным. Весьма вероятно, что в

[1]) Один из рецензентов обратил наше внимание на отчет [11], в котором приведены результаты таких поисков с помощью вычислительной машины. В частности, оказалось, что для $l=6$ не существует кодов, длина блока которых составляла бы 4095 или меньше.

большинстве случаев можно построить пригодный для практического использования циклический код, исправляющий пакеты ошибок длиной до l, с числом проверочных знаков, близким к наименьшему теоретически возможному значению $2l$, если выбирать длину блока равной некоторой доле $n_{опт}$. Примеры параметров нескольких таких кодов приведены в таблице IV.

Таблица IV

Параметры нескольких неоптимальных кодов для исправления пакетов ошибок до $l>4$

l	r	n	$k=n-r$
5	11	21	10
5	12	63	51
6	12	21	9
6	13	63	50
7	14	21	7
7	15	63	48
8	16	21	5
8	17	63	46
9	18	21	3
9	19	63	44
10	20	21	1

Д-р Стоун из Станфордского научно-исследовательского института в процессе многих плодотворных обсуждений с авторами настоящей работы внес существенный вклад в отдельные ее разделы. Авторы чрезвычайно признательны ему за помощь. Они благодарны также Боумэну за проведение довольно трудоемкой проверки 98 оптимальных кодов на цифровом моделирующем устройстве.

ЛИТЕРАТУРА

1. Peterson W. W., Error-correcting codes, John Wiley and Sons, New York, 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]
2. Fire P., A class of multiple-error-correcting codes for non-independent errors, Reconnaissance Systems Lab., Mountain View, Calif., Sylvania Rept. No RSL-E-2, March 1959.

3. Reiger S. H., Codes for correction of clustered errors. *IRE Trans. on Inform. Theory*, **IT-6** March 1960, 16—21.
4. Abramson N. M., Error-correcting codes from linear seqential circuits, Information Theory, Fourth London Symposium, London, Butterworth, 1961, 26—37.
5. Abramson N. M., A class of systematic codes for non-independent errors, *IRE Trans. Inform. Theory*, **IT-5**, December (1959), 150—157.
6. Abramson N. M., A note on single-error-correcting binary codes, *IRE Trans. Inform. Theory*, **IT-6**, September (1960), 502—503.
7. Melas C. M., A new group of codes for correction of dependent errors in data transmission, *IBM J. Res. and Dev.*, **4**, July (1960), 58—65.
8. Elspas B., The theory of autonomous linear seqential networks, *IRE Trans. Circuit Theory*, **CT-6**, March (1959), 45—60. [Русский перевод: Элспас Б., Теория автономных линейных последовательных сетей, Кибернетический сб., вып. 7, ИЛ, М., 1963, 90—128.]
9. Document of Stanford Res. Inst., Menlo Park, Calif., Tech. Rep. No 1, for the Rome Air Dev. Ctr., Griffiss AFB N. Y., Contract No AF 30 (602)—2327, October 1961.
10. Basic Electronic Research. Stanford Electronics Labs., Stanford, Calif. Quarterly Status Rept., No 25, October 1—December 31, 1960, 41—43.
11. Foulk C. R., Some properties of maximally-efficient cyclic burst-error-correcting codes and results of a computer search for such codes, Digital Computer Lab., University of Illinois, Urbana, Rept. No 375, June 12, 1961.

ИСПРАВЛЕНИЕ МНОГОКРАТНЫХ ПАКЕТОВ ОШИБОК[1])

Дж. Стоун

Приводятся два результата относительно исправления многократных пакетов ошибок. В разделе II приведена теорема, позволяющая повысить способность кодов, построенных над $GF(2)$, исправлять такие ошибки с помощью построения циклических кодов с заданным весом. В разделе III приведен метод построения квазициклических кодов над $GF(p^k)$, исправляющих многократные пакеты ошибок.

I. Введение

A. Краткое содержание

Исправлению одиночных пакетов ошибок в литературе по корректирующим кодам уделяется достаточно много внимания (см., например, Питерсон [4], гл. 10). Вопросы исправления многократных пакетов ошибок являются более сложными и им соответственно уделяется меньше внимания.

Две теоремы, приведенные в настоящей работе, представляют собой попытку облегчить исправление многократных пакетов за счет создания определенного типа кодов с заданным весом, т. е. за счет сведения исправления пакетов ошибок к исправлению многократных (единичных) ошибок.

Первая теорема показывает, что эффективность циклических кодов с заданным весом, построенных над $GF(2)$, в отношении исправления многократных пакетов почти на 50% выше, чем можно было ожидать.

---

[1]) Stone J., Multiple burst error correction, *Information and Control*, **4** (1961), № 4, 324—331.

Построение таких кодов изучалось довольно подробно (см., например, Боуз, Чоудхури [1], [2], Питерсон [4]).

Вторая теорема показывает, что для исправления m-кратных пакетов ошибок длины p или меньше можно использовать весьма эффективные квазициклические коды с весом $2m+1$, построенные над $GF(p^k)$.

В. Определения

Пусть $V^n(GF(p^k))$ — n-мерное векторное пространство над полем Галуа из p^k элементов. Некоторые элементы $v=(a_0, a_1, \ldots, a_{n-1})$ этого векторного пространства следует считать представляющими сообщение, и это сообщение должно передаваться по каналу в следующем порядке:

$$a_{n-1}, a_{n-2}, \ldots, a_0.$$

Векторы, представляющие сообщение, назовем кодом. Коды, обсуждаемые здесь, суть групповые коды, т. е. кодовые векторы образуют подгруппу группы $V^n(GF(p^k))$. При передаче могут возникнуть ошибки. Предположим, что принят искаженный вектор $v'=(a_0', a_1', \ldots, a_{n-1}')$. Тогда ошибка имеет вид

$$e=v'-v=(a_0'-a_0,\ a_1'-a_1, \ldots, a_{n-1}'-a_{n-1}).$$

Если мы ищем специальный код, исправляющий некоторый класс ошибок P, то возникает вопрос о существовании алгоритма, по которому принятый вектор декодируется правильно (в предположении, что ошибка принадлежит P). Если v_i' представляет собой вектор, в который переходит v_i после сложения с ошибкой e_i, то желаемый алгоритм существует тогда и только тогда, когда из условия

$$v_1'=v_1+e_1=v_2+e_2=v_2', \tag{1}$$

где e_i принадлежит P, v_i принадлежит коду, $i=1, 2$, следует равенство

$$v_1=v_2. \tag{2}$$

Для групповых кодов эквивалентным является условие: если e_1-e_2 принадлежит коду и e_i принадлежит P, то $e_1=e_2$. Введем следующее

Определение. *Групповой код C исправляет класс ошибок P тогда и только тогда, когда из того, что $e_1 - e_2$ принадлежит C и e_1, e_2 принадлежат P, следует, что $e_1 = e_2$.*

Пакет ошибок длины l есть вектор, у которого равны нулю все компоненты, не принадлежащие множеству из l соседних компонент, первая и последняя из которых не равны нулю. (Последняя компонента a_{n-1} считается стоящей рядом с a_0.) В этой статье за P принимается совокупность всех векторов, которые можно представить в виде суммы не более m пакетов ошибок, длина которых не превышает l. Пусть вектор

$$v = (a_0, a_1, \ldots, a_{n-1})$$

принадлежит $V^n(GF(p^k))$. Обозначим через g_v полином над $GF(p^k)$, определенный следующим образом: $g_v(x) = \sum_{i=0}^{n-1} a_i x^i$. Если существует полином g над $GF(p^k)$, такой, что

$$C = \{v : g \mid g_v\}, \tag{3}$$

то C называют *квазициклическим кодом*. Если, кроме того, n — период[1]) g, C называют *циклическим кодом* (легко проверить, что для такого кода из принадлежности к нему вектора $v = (a_0, a_1, \ldots, a_{n-1})$ следует, что этому коду принадлежат циклические перестановки v, например $(a_{n-1}, a_0, a_1, \ldots, a_{n-2})$). Ясно, что эти коды групповые.

Если v принадлежит $V^n(GF(p^k))$, определим вес v, обозначаемый через $\omega(v)$, как число ненулевых компонент v. По определению, *вес кода C* есть минимальный ненулевой вес его элементов. Вес полинома h над $GF(p^k)$, обозначаемый через $\omega(h)$, равен числу ненулевых коэффициентов h. $[x]$ означает наибольшее целое, не превышающее x. *Избыточность* квазициклического кода C есть, по определению, степень g.

---

[1]) *Период элемента* θ в $GF(p^k)$ есть наименьшее целое n, такое, что $\theta^n = 1$. *Период полинома* f над $GF(p^k)$ есть наименьшее целое n такое, что f делит $x^n - 1$ над $GF(p^k)$.

II. Исправление многократных пакетов с помощью циклического кода заданного веса над $GF(2)$

Легко проверить, что циклический код C веса $2mt+1$ исправляет класс ошибок P, содержащий до m пакетов длины t. Следующая ниже теорема и замечания показывают, что C в действительности исправляет m-кратные пакеты длины, большей t. Обычно выигрыш достигает почти 50%.

Теорема 1. *Если* $C \subset V^n(GF(2))$ *— циклический код веса* $2mt+1$ *и* $n > 3mt$, *то он исправляет класс* P *всех сочетаний не более чем* m *пакетов, длина которых не превышает* $l = t + [(t-2)/2 + 3/(4m)]$. *Для* $m=1$ *это дает* $l = t + [(2t-1)/4]$, *а для* $m \geqslant 2$ $l = t + [(t-2)/2]$.

Доказательство. C есть циклический код и поэтому существует полином g над $GF(2)$ с периодом n, такой, что

$$C = \{v : g \mid g_v\}.$$

Положим, что $e_1 - e_2$ принадлежит C, а e_1 и e_2 принадлежат P. Нужно показать, что $e_1 - e_2 = 0$. Из допущения относительно n следует, что $n > 2ml$, и поэтому вектор $e_1 - e_2$ имеет по крайней мере одну нулевую компоненту. Так как C — циклический код, существует такая циклическая перестановка $e_1 - e_2$, что она принадлежит C и имеет 0 последней компонентой.

Если будет доказано, что эта перестановка равна нулю, вектор $e_1 - e_2$ тоже должен быть равен нулю. Таким образом, при доказательстве того, что из $e_1 - e_2 \in C$ следует $e_1 = e_2$, мы можем ограничиться векторами $e_1 - e_2$ с нулем в последней позиции. Больше того, мы можем допустить, что

$$\omega(e_1) + \omega(e_2) \geqslant 2mt + 1, \tag{5}$$

ибо, если это не так, то

$$\omega(e_1 - e_2) \leqslant \omega(e_1) + \omega(e_2) < 2mt + 1, \tag{6}$$

а поскольку C имеет вес $2mt+1$, из соотношения $e_1 - e_2 \in C$ немедленно следует, что $e_1 - e_2 = 0$. Так как

e_j, $j=1, 2$, принадлежат P, мы имеем

$$g_{e_j} = \sum_{i=1}^{m} x^{k_i} E_{ij}, \tag{7}$$

где E_{ij} есть полином над $GF(2)$ со свободным членом и порядком, меньшим $t+s$, $s=[(t-2)/2+3/(4m)]$ (для каждого j E_{ij} представляют неперекрывающиеся пакеты). Учитывая, что $1+1=0$, E_{ij} можно записать в виде

$$E_{ij} = \sum_{a=0}^{t+s-1} x^a + \sum_{k=1}^{a_{ij}} x^{a_{ijk}}, \quad i=1, 2, \ldots, m, \quad j=1, 2, \tag{8}$$
$$0 \leqslant a_{ijk} \leqslant t+s-1,$$

где a_{ijk} указывают коэффициенты E_{ij}, равные 0, а a_{ij} обозначает общее число таких коэффициентов. Очевидно,

$$\omega(E_{ij}) = t+s-a_{ij}. \tag{9}$$

Так как $\omega(g_{e_j}) = \omega(e_j)$, мы можем подставить (9) и (7) в (5) и в результате получим

$$\sum_{i=1}^{m} t+s-a_{i_1} + \sum_{i=1}^{m} t+s-a_{i_2} = 2m(t+s) - \sum_{i,j} a_{ij} \geqslant$$
$$\geqslant 2mt+1, \tag{10}$$

или

$$2ms-1 \geqslant \sum_{i,j} a_{ij}. \tag{11}$$

Заметим, что бином $(1+x)$, умноженный на первую сумму в (8), дает полином $1+x^{t+s}$ веса 2, тогда как взятая $(1+x)$ раз вторая сумма даст полином самое большее веса $2a_{ij}$. Отсюда

$$\omega((1+x)E_{ij}) \leqslant 2+2a_{ij} \tag{12}$$

и потому

$$\omega((1+x)g_{e_1-e_2}) \leqslant \omega((1+x)g_{e_1}) + \omega((1+x)g_{e_2}) \leqslant$$
$$\leqslant \sum_{i=1}^{m} \omega((1+x)E_{i_1}) + \sum_{i=1}^{m} \omega((1+x)E_{i_2}) \leqslant$$
$$\leqslant \sum_{i,j} 2+2a_{ij} = 4m+2\sum_{i,j} a_{ij}. \tag{13}$$

Поскольку мы предположили раньше, что $g_{e_1-e_2}$ имеет порядок, меньший $n-1$, $(1+x)g_{e_1-e_2}$ имеет порядок, не

превышающий $n-1$, и поэтому существует вектор η, такой, что $g_\eta = (1+x)\, g_{e_1-e_2}$. В соответствии с (3) $C = \{v : g \mid g_v\}$ для некоторого g, и так как делители полинома $g_{e_1-e_2}$ будут делить g_η, из того, что $e_1 - e_2$ принадлежит C, следует, что η принадлежит C. Мы покажем, что $\eta = 0$, откуда будет следовать, что $g_\eta = 0$, значит

$$(1+x)\, g_{e_1-e_2} = 0,$$

или $e_1 = e_2$.

Из (13) и (11) получаем

$$\omega(g_\eta) \leqslant 4m + 2\sum_{i,j} a_{ij} \leqslant 4m + 4ms - 2. \qquad (14)$$

Подставляя в (14) $s = [(t-2)/2 + 3/(4m)]$ и замечая, что $(t-2)/2 + 3/(4m)$ не есть целое, получим

$$\omega(g_\eta) \leqslant 4m + 4m\left[\frac{2mt - 4m + 3}{4m}\right] - 2 <$$
$$< 4m + 2mt - 4m + 3 - 2 = 2mt + 1. \qquad (15)$$

Так как η принадлежит коду C, имеющему вес $2mt+1$, $\eta = 0$, откуда, как было отмечено выше, следует, что $e_1 = e_2$. Это и заканчивает доказательство, если не считать замечания о том, что для $m \geqslant 2$ имеет место равенство

$$[(t-2)/2 + 3/(4m)] = (t-2)/2.$$

В качестве примера рассмотрим код с весом 25 и длины, большей 36. Положив $2mt + 1 = 25$ или $mt = 12$, построим следующую таблицу (см. табл. I), которая показывает, что теорема должна оказаться полезной при исправлении единичных, двойных и тройных пакетов (для пакетов 4, 6 и 12 она не дает нетривиальной информации).

Отметим, что условие $n > 3mt$ используется только для того, чтобы гарантировать $n > 2ml$, т. е. чтобы исправлялись пакеты длины l. В общем случае, для циклического кода C с весом $2mt+1$ при

$$l = \min\left(\left[\frac{n-1}{2m}\right], t + \left[\frac{t-2}{2} + \frac{3}{4m}\right]\right)$$

можно установить, используя в сущности доказательство,

приводимое выше, что C исправляет m-значные пакеты длины l.

Если надо исправлять сочетание кратных пакетов и случайных единичных ошибок [1]), можно доказать теорему, аналогичную теореме 1.

Таблица I

$n > 36$

m	t	l	ml
1	12	17	17
2	6	8	16
3	4	5	15
4	3	3	12
6	2	2	12
12	1	1	12

Например, если P состоит из сочетаний не более чем m пакетов длины l и не более чем u единичных ошибок, код с весом $2(mt+u)+1$, где

$$t \geqslant \frac{2}{3} l + 1 + [(2u-3)/6m],$$

будет исправлять P [при $n > 2(mt+u)$].

Очевидно, что поскольку код с весом $2(ml+u)+1$ должен исправлять P, выигрыш зависит от того, насколько $1+[(2u-3)/6m]$ меньше $l/3$. Отсюда следует, что хотя и можно доказать аналогичные теоремы для кодов, исправляющих единичные ошибки и более длинные кратные пакеты, а также некоторые двойные и т. п. ошибки, увеличение эффективности, указываемое этими теоремами, снизится (если только l и m не взяты большими).

---

[1]) Желательность исправления таких ошибок была предложена Вольфом.

III. Исправление многократных пакетов ошибок с использованием дифференцирования над полями $GF(p^k)$

Теорема 2. *Пусть f — полином над $GF(p^k)$ степени d — порождает квазициклический код в $V_n(GF(p^k))$ веса по крайней мере $2m+1$ и пусть $pd<n$. Тогда квазициклический код в $V_n(GF(p^k))$, порожденный полиномом f^p, исправляет m-кратные пакеты, длина которых не превышает p.*

Доказательство[1]). Нужно показать, что если e_j, $j=1, 2$,— ошибки рассматриваемого класса, то из соотношения $e_1-e_2 \in C$ следует, что $e_1=e_2$. Пусть

$$g_{e_1-e_2}(x)=\sum_{j=1}^{2}\sum_{i=1}^{m} x^{k_{ij}}E_{ij}(x), \quad 0\leqslant k_{ij}<n, \tag{16}$$

где E_{ij} — полином над $GF(p^k)$ со свободным членом, имеющий степень ниже p. Если e_1-e_2 принадлежит C, то должен существовать полином h над $GF(p^k)$, такой, что

$$hf^p=g_{e_1-e_2}. \tag{17}$$

Так как $pf^{p-1}=0$, то взяв производные, мы видим, что

$$f^p \mid g^{(q)}_{e_1-e_2}, \quad q=0, 1, 2, \ldots, p-1, \tag{18}$$

где (q) обозначает порядок дифференцирования над $GF(p^k)$.

Пусть q_0, $0\leqslant q_0\leqslant p$,— первое значение q, для которого $q^{(q)}_{e_1-e_2}=0$. Если $q_0\neq 0$, то в силу соотношения (18)

$$f \mid g^{(q_0-1)}_{e_1-e_2}$$

и

$$g^{(q_0-1)}_{e_1-e_2}\neq 0.$$

---

[1]) *Замечание к доказательству.* В письме, опубликованном в *Proc. IRE*, август 1961, Ф. Корром и озаглавленном „Обнаружение многократных пакетов", рассматриваются циклические коды, порожденные полиномом $f(x^l)$, где l есть длина пакетов, подлежащих исправлению или обнаружению. Если $k=1$, теорема дает укороченный вариант специального случая этих кодов, для которых $f(x^l)=f^l(x)$.

Однако $g_{e_1-e_2}^{(q_0-1)}$, очевидно, есть полином от x^p, и равенство (16) показывает, что он не может иметь больше $2m$ ненулевых компонент. Поскольку f не делит такие полиномы, должно быть $q_0=0$. Отсюда

$$g_{e_1-e_2}=0 \text{ или } e_1=e_2,$$

что и заканчивает доказательство.

Для того чтобы эффективно использовать теорему 2, необходимо выбирать полином f желаемого типа таким образом, чтобы он имел минимальный возможный порядок d, так как избыточность кода равна pd. Выигрыш, даваемый уменьшением порядка f, может сделать методы вычисления заслуживающими внимания. Однако, если f строится методом Боуза — Чоудхури (так что он имеет корни $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2m}$), то в общем случае избыточность, получаемая по теореме 2, достаточна для построения кода Боуза — Чоудхури (ассоциированного с полиномом g с корнями $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2pm}$), который исправляет все ошибки кратности pm или меньше. Таким образом, если нельзя построить f более эффективным способом, единственное преимущество, которое дает теорема 2, состоит в том, что построение f^p заключается только в возведении f в p-ю степень (над полем характеристики p), что много проще, чем операции с корнями, участвующие в построении g.

Пример. Полином $f(x)=(x^7+x^3+1)(x^7+x^3+x^2+x+1)(x^7+x^4+x^3+x^2+1)$ имеет период $n=127$ и является кодом Боуза — Чоудхури над $GF(2)$ с корнями $\alpha, \alpha^2, \ldots, \alpha^6$, порождающим код с весом, не меньшим 7. Это дает $m=3$ и так как $p=2$, то по теореме 2 полином $f^2(x)=(x^{14}+x^6+1)(x^{14}+x^6+x^4+x^2+1)(x^{14}+x^8+x^6+x^4+1)$ порождает квазициклический код с $n=127$, исправляющий все тройные пакеты длины самое большее $p=2$.

Соответствующий код Боуза — Чоудхури, исправляющий все ошибки кратности $3\times 2=6$ или меньше, порождается полиномом

$$g(x)=(x^7+x^3+1)(x^7+x^3+x^2+x+1)(x^7+x^4+x^3+x^2+1)\times$$
$$\times(x^7+x^6+x^5+x^4+x^2+x+1)(x^7+x^5+x^4+x^3+x^2+x+1)(x^7+x^6+x^4+x^2+1),$$

причем f^2 и g соответствуют одной скорости передачи информации.

ЛИТЕРАТУРА

1. Bose R. C., Ray-Chaudhuri D. K., On a class of error-correcting binary group codes, *Information and Control*, 3 (1960), 68—79. [Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 2, ИЛ, М., 1961.]
2. Bose R. C., Ray-Chaudhuri D. K., Further results on error-correcting binary group codes, *Information and Control*, 3 (1960), 279—290. [Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Дальнейшие результаты относительно двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 6, ИЛ, М., 1963, 7—12.]
3. Fire R., A class of multiple error-correcting binary codes for non-independent errors, Sylvania Rept., RSL-E-2 (1959), Sylvania Reconnaissance Systems Laboratory, Mountain View, California.
4. Peterson W. W., Error-correcting codes, Mass. Insf. Technol., Press and Wiley, New York, 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]

О КОДАХ, ОБНАРУЖИВАЮЩИХ ОШИБКИ В АСИММЕТРИЧНЫХ КАНАЛАХ [1])

Дж. Бергер

В статье описаны некоторые новые коды, принадлежащие к числу разделимых. Применительно к обнаружению ошибок в полностью асимметричном канале эти коды являются совершенными. Приведены некоторые результаты сравнения одного простого кода, в котором проверочные знаки соответствуют сумме единиц, содержащихся среди информационных знаков, с кодом „4 из 8". Найдено, что в нескольких случаях сравнение оказывается в пользу нового кода. Указаны некоторые более сложные коды того же типа.

I. Введение

Последнее время в некоторых системах связи стали применяться коды с постоянным весом, особенно коды „4 из 8" [2]). Одной из основных причин применения этих кодов являются их преимущества при использовании в каналах связи, являющихся в значительной степени асимметричными. Дело в том, что в случае полностью асимметричного канала код с постоянным весом является совершенным применительно к обнаружению ошибок. Полностью асимметричным является канал, в котором имеет место только один вид ошибок, т. е. возможно либо только преобразование нулей в единицы, либо наоборот только единиц в нули. В двоичном симметричном канале (т. е. в канале, в котором оба вида ошибок равновероятны) коды с постоянным весом обнаруживают все возможные сочетания нечетного числа ошибок; из всех возможных сочетаний четного числа ошибок они не обнаруживают только такие, при которых преобразование нулей в единицы на одних позициях

[1]) Berger J. M., A note on error detection code for asymmetric channels, *Information and Control*, **4** (1961) 1, 68—73.

[2]) То есть код длиной 8, каждое кодовое слово которого состоит из четырех единиц и четырех нулей.— *Прим. перев.*

компенсируется преобразованием единиц в нули на других. Кроме того, при заданной длине кодового слова оптимальный код с постоянным весом имеет обычно больше разрешенных комбинаций, чем разделимый код с эквивалентной обнаруживающей способностью.

Основной недостаток кода с постоянным весом состоит в том, что этот код является неразделимым. Под разделимостью мы понимаем возможность указать, какие двоичные знаки являются носителями передаваемой информации, а какие обеспечивают возможность обнаружения ошибок. В кодах с постоянным весом обнаруживающая способность обусловливается самим видом кодового слова, благодаря чему отделить проверочные знаки без преобразования кода невозможно. В результате такой неразделимости обнаруживающая способность кода оказывается неразрывно связанной с характером информационных знаков. А это в свою очередь приводит к невозможности просто изменить код в существующей системе. При использовании кода с постоянным весом обычно прежде всего определяется основной алфавит системы. Затем выбирается надлежащий код, имеющий достаточное количество кодовых слов. Далее, каждой букве алфавита сопоставляется кодовое слово с постоянным весом. Недостатки такого закрепления кодовых слов особенно проявляются в тех случаях, когда буквы большей частью передаются группами. Поскольку при использовании неразделимых кодов избыточность заложена уже в каждой (закодированной) букве, то повысить экономичность за счет кодирования сразу целой группы букв оказывается невозможным. Таким образом, разделимые коды обладают преимуществом по сравнению с неразделимыми; оно состоит в большей гибкости и возможности повышения экономичности, особенно в тех случаях, когда передаваемая информация кодируется большими блоками. Остается выяснить, могут ли эти преимущества быть сохранены в случае асимметричного канала.

Целью настоящей работы является описание некоторых кодов, которые, являясь разделимыми, в то же время обнаруживают все ошибки, возможные в полностью асимметричном канале. Один простой код будет описан подробно. Будут приведены результаты его сравнения

по некоторым показателям с кодом с постоянным весом. Далее будут указаны некоторые другие коды того же типа, которые обладают некоторыми дополнительными свойствами.

II. Описание кодов с суммированием

Рассмотрим множество, состоящее из $n-r$ информационных и r проверочных двоичных знаков. Образуем двоичное число, выражающее количество единиц, содержащееся среди $n-r$ информационных знаков, и дополним каждый знак этого числа по модулю 2, т. е. заменим все единицы на нули и все нули — на единицы. Полученное в результате описанных действий двоичное число и представляет собой r проверочных знаков рассматриваемого кода[1]). (Возможен и другой способ, состоящий в том, что в качестве r проверочных знаков используют двоичное число, соответствующее количеству нулей в $n-r$ информационных двоичных знаках. Этот способ полностью эквивалентен первому, но его техническая реализация может оказаться проще.) Таким образом, r равно наименьшему целому числу, превышающему $\log_2(n-r)$. В качестве информационных можно выбрать любую последовательность $n-r$ двоичных знаков. Такой последовательностью может быть одна шестизначная буква (при этом $r=3$), шесть восьмизначных букв (при этом $r=6$) и т. д. В качестве конкретного примера рассмотрим случай $n=9$, $r=3$. Одной из разрешенных комбинаций в этом случае является следующее кодовое слово:

r проверочных знаков $(n-r)$ информационных знаков
(100 011010).

Такие коды мы будем обозначать $\Sigma(n-r, r)$.

В полностью асимметричном канале возможно либо только преобразование нулей в единицы, либо наоборот. Рассмотрим сперва случай, когда единицы преобразуются в нули. Тогда при возникновении ошибок количество

[1]) Этот частный код был независимо от автора предложен Смитом младшим. Кроме того, он был предложен Фримэном в частной беседе.

единиц, содержащихся среди информационных знаков, уменьшается, и число, равное количеству принимаемых единиц, которое должно было бы совпадать с числом, представляемым принимаемыми проверочными двоичными знаками, оказывается меньше последнего. Каждый из принимаемых проверочных знаков дополняется по модулю 2 (чем компенсируется дополнение, осуществленное на передающей станции).

Полученное в результате этого двоичное число и используется для сравнения. Если при передаче единицы преобразовались в нули, то r принимаемых проверочных знаков будут содержать больше нулей (чем было передано) и полученное в результате дополнения по модулю 2 двоичное число будет больше, чем исходная проверочная сумма. Таким образом, при рассматриваемом характере ошибок любые из них приводят к тому, что сумма, вычисленная по принимаемым информационным знакам, всегда оказывается меньше числа, представляемого принимаемыми проверочными знаками. При таком положении дел любые ошибки, состоящие в преобразовании единиц в нули, будут обнаружены. Аналогичное положение имеет место и в том случае, когда в канале нули преобразуются в единицы. В табл. I приведен подробный пример всех описанных выше операций. Итак, если для любых $n-r$ информационных знаков выбрать описанным выше способом r проверочных знаков, то полученный код обнаружит все ошибки, которые могут быть внесены полностью асимметричным каналом.

В любом другом канале такой код обнаружит все единичные и большую часть многократных ошибок. Если предположить, что ошибки независимы и что все кодовые слова равновероятны, то легко можно подсчитать ожидаемое количество необнаруженных ошибок для рассматриваемого кода, а также для кода с постоянным весом „4 из 8“. Результаты некоторых вычислений для случая симметричного канала, которые приведены в таблице II, показывают, что рассматриваемый код обладает преимуществами по сравнению с кодом „4 из 8“.

Интересно отметить, что хотя общее количество возможных двойных ошибок для кода Σ (6, 3) больше, чем для кода „4 из 8“ (так как длина первого из этих кодов

равна 9, а второго 8), однако не обнаруживаются они значительно реже. Среднее количество необнаруживаемых двойных ошибок для кода Σ' (6, 3) составляет 10,5, а для кода „4 из 8“ — 16. При этом превосходство кода Σ (6, 3) сохраняется и в случае любой асимметрии. Аналогичный результат мы имеем и для кода Σ (7, 3).

Очевидно, что с ростом n эффективность обнаружения двойных ошибок для кодов с суммированием становится ниже, чем для кодов, образуемых путем объединения нескольких слов кода „4 из 8“ в одно кодовое слово соответствующей длины. При этом при длине слова,

Таблица I

1. Подлежащие передаче информационные знаки	011010
2. Двоичная запись количества единиц	011
3. Количество единиц после дополнения по модулю 2	100
4. Переданное слово	100 011010
5. Слово, принятое с двумя ошибками	000 001010
6. Число, полученное путем подсчета количества единиц среди принятых информационных знаков	010
7. Число, полученное в результате дополнения по модулю 2 принятых проверочных знаков	111

Таблица II

Среднее количество необнаруживаемых сочетаний ошибок, вносимых симметричным каналом

Характер ошибок	Код		
	Σ(6, 3)	Σ(7, 3)	„4 из 8“
Независимые двойные ошибки	10,5	14	16
Независимые тройные ошибки	1,97	—	0
Пакеты * длины 2	3	3,5	4
Пакеты * длины 3	2,68	—	3,4

* Под пакетом длины l подразумевается любое сочетание ошибок, охватывающее l знаков, при условии, что первый и (последний) l-й из этих знаков содержат ошибки.

соответствующей примерно одинаковым избыточностям, эффективность обнаружения ошибок для обоих кодов оказывается примерно одинаковой. Так, в симметричном канале код Σ (12, 4) не обнаруживает 39 сочетаний из двух ошибок, а код, образуемый путем объединения двух слов кода „4 из 8“,— 32 сочетания. Однако при больших n код с суммированием обладает значительно меньшей избыточностью и в то же время остается совершенным в отношении обнаружения ошибок в полностью асимметричном канале.

III. Использование кодов с суммированием

Из сказанного выше следует, что наилучшее использование кода с суммированием может иметь место в двух случаях. Первый из них имеет место, если передача является стартстопной и вообще, если проверку на обнаружение ошибок удобно осуществлять для каждой буквы отдельно. При этом можно использовать код Σ (6, 3) или Σ (7, 3), так как эти коды обладают большей помехоустойчивостью, чем код с постоянным весом, и в то же время, в отличие от последнего, являются разделимыми. Второй случай, когда целесообразно использовать коды с суммированием, имеет место, если, как правило, передаются длинные записи. В этом случае код с суммированием может быть применен ко всей записи в целом. При этом все буквы записи представляются в двоичной форме и запись является последовательностью двоичных букв. Ввиду того, что в этом случае помехоустойчивость при применении кода с суммированием оказывается ниже, чем при объединении нескольких слов кода „4 из 8“ в одно кодовое слово, то для большей надежности можно дополнительно применить циклический код. Поскольку с увеличением количества информационных знаков избыточность кода с суммированием растет по закону только $\log_2$ количества информационных знаков, то при таком кодировании можно обеспечить значительно большую помехоустойчивость, чем при использовании кода „4 из 8“, и в то же время добиться значительно лучшего использования канала. Например, для передачи записи на карте, состоящей примерно из 480 двоичных

знаков, код с суммированием потребует 9 проверочных знаков, а код „4 из 8“ — примерно 160 проверочных знаков. Если в дополнение к коду с суммированием использовать циклический код с 20 или менее проверочными знаками, то помехоустойчивость почти наверняка окажется выше, чем при использовании кода „4 из 8“. Во всех случаях применение кода с суммированием оправдано только при условии, что ожидается значительная асимметрия канала. В противном случае можно ожидать, что чисто циклический код даст лучшие результаты.

IV. Другие коды

По аналогии с кодом с суммированием, который был описан выше, можно построить и другие коды, которые будут обладать некоторыми дополнительными свойствами, оставаясь в то же время совершенными в отношении обнаружения ошибок в полностью асимметричном канале. Рассмотрим, например, код, каждой информационной позиции которого приписаны различные веса, причем ни один из этих весов не является степенью двух. Проверочные знаки этого кода образуются путем суммирования весов, соответствующих тем информационным позициям, на которых расположены единицы. Очевидно, что веса, приписываемые следующим друг за другом информационным позициям, представляют собой последовательность чисел 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17 и т. д. Такой код будет обладать всеми свойствами ранее описанного кода с суммированием, но, кроме этого, он будет обнаруживать все двойные ошибки. Однако проверочных знаков он потребует примерно вдвое больше, чем простой код с суммированием. Его количество проверочных знаков равно наименьшему целому числу, превышающему

$$\log_2 \left\{ \left[(k+\mu) \frac{k+\mu+1}{2} \right] - 2\mu + 1 \right\},$$

где k — количество информационных знаков, а μ определяется из соотношения

$$2^{\mu-1} < k+\mu < 2^{\mu}.$$

В качестве примера рассмотрим информационное слово (0110100001). При использовании приведенной выше последовательности весов сумма весов для такого слова равна 34. Количество проверочных знаков для $n = 10$, $\mu = 4$ составляет 7. Двоичная запись суммы весов имеет вид (0100010). Таким образом, полная последовательность, передаваемая в канал, имеет вид (1011101 0110100001). В полностью асимметричном канале такой код обнаружит любые ошибки по той же причине, что и в ранее рассмотренном случае. Однако, кроме этого, в канале с любой степенью асимметрии, наряду со всеми одиночными ошибками, этот код обнаружит все двойные ошибки. Чтобы доказать это, рассмотрим следующие три случая, которые исчерпывают все варианты двойных ошибок. (а) Обе ошибки расположены в проверочных позициях. Они будут обнаружены, так как поскольку всем позициям проверочной суммы соответствуют веса, являющиеся различными для каждой позиции степенью двойки, то очевидно, что в случае искажения двух проверочных знаков принимаемая проверочная сумма не может иметь правильного значения и, следовательно, не может совпасть с суммой, которая вычислена по принимаемым информационным знакам. (б) Обе ошибки расположены в информационных позициях. Поскольку, как и ранее, все веса различны, то ни при каком расположении двух ошибок их воздействие на сумму, вычисляемую по принимаемым информационным знакам не может скомпенсироваться. (в) Одна ошибка расположена в проверочной, а другая—в информационной позиции. Ошибка в проверочной позиции может привести либо к увеличению, либо к уменьшению проверочной суммы на величину, равную некоторой степени двойки. В то же время ни одна информационная позиция не имеет веса, являющегося степенью двойки. Поэтому и такие две ошибки не могут скомпенсировать друг друга.

Без сомнения аналогичным образом можно построить коды, которые будут обнаруживать не только двойные, но и четверные и т. д. ошибки. Однако сложность построения таких кодов значительно возрастает с ростом кратности обнаруживаемых ошибок. В настоящее время применение таких кодов вряд ли оправдано. Например, для того, чтобы были обнаружены две пары компенсирую-

щих друг друга ошибок, т. е. два перехода нулей в единицы и два перехода единиц в нули, необходимо так подобрать веса, чтобы все разности любых двух весов были различными. По-видимому, простого способа построения последовательности весов с такими свойствами не существует. Кроме того, для защиты всего кодового слова от необнаруживаемых ошибок на последовательность весов должны быть наложены и другие ограничения.

V. Заключение

Коды, описанные в настоящей статье, являются разделимыми и в то же время совершенными в отношении обнаружения ошибок в полностью асимметричном канале. Основная задача состояла в доказательстве того, что разделимые коды могут обладать основным свойством кодов с постоянным весом и что преимущества, даваемые асимметрией канала, могут быть использованы и при сохранении гибкости, которая присуща разделимым кодам. Было показано, что описанные коды с суммированием в нескольких случаях могут успешно конкурировать с кодами с постоянным весом „4 из 8“. Кроме того, указаны возможности обобщения кодов с суммированием, позволяющие повысить их обнаруживающую способность за счет увеличения избыточности.

О КОДАХ С СУММИРОВАНИЕМ, ОБНАРУЖИВАЮЩИХ ПАКЕТЫ ОШИБОК [1])

Дж. Бергер

В статье описан новый код, обнаруживающий пакеты ошибок. Этот код принадлежит к числу кодов с суммированием — его проверочные двоичные знаки определяются путем алгебраического суммирования информационных знаков, которым приписаны надлежащим образом выбранные веса. Описываемый код при наличии примерно $l+\log_2(k/l)$ избыточных двоичных знаков, где k — количество информационных двоичных знаков, обнаруживает в произвольном канале [2]) любой пакет ошибок, длина которого не превышает l. Применительно к обнаружению ошибок в полностью асимметричном канале этот код является совершенным.

Коды с суммированием, описанные в предыдущей статье [1], легко можно видоизменить таким образом, что они будут обнаруживать любой пакет ошибок, длина которого не превышает некоторого наперед заданного значения. При этом сохраняется их свойство обнаруживать все ошибки, которые возможны в полностью асимметричном канале. Такие коды обладают всеми свойствами простого кода с суммированием. Их избыточность значительно меньше избыточности, которая имела бы место в случае совмещения циклического кода с кодом с суммированием. Это означает следующее. Пусть имеется код с суммированием, обнаруживающий все ошибки в полностью асимметричном канале, каждое кодовое слово которого содержит k информационных двоичных знаков, и пусть в дополнение к тому эффекту, который дает применение этого кода, требуется обнаруживать любой пакет ошибок длиной до l. Тогда в случае совмещения двух кодов, при

---

[1]) Berger J. M., A Note on burst detecting sum codes, *Information and Control*, 4 (1961), № 2—3, 297—299.

[2]) То есть в канале с произвольной степенью асимметрии, в том числе и в симметричном канале.— *Прим. ред.*

котором обеспечивалось бы совершенное обнаружение ошибок в полностью асимметричном канале и обнаружение любого пакета ошибок длиной до l в произвольном канале, потребовалось бы примерно $l + \log_2(k)$ проверочных знаков [1]). В настоящей же статье описывается единый простой код, который, как будет показано ниже, для получения тех же результатов требует примерно $l + \log_2(k/l)$ проверочных знаков. Этот код принадлежит к числу кодов с суммированием. Веса, приписываемые его информационным позициям, выбираются с таким расчетом, чтобы обеспечить требуемую обнаруживающую способность в отношении пакета ошибок. Как показано в [1] применительно к ранее описанному коду с суммированием, код, образуемый таким образом, является разделимым (в том смысле, что в нем можно разграничить информационные и проверочные позиции). Проверочные знаки описываемого кода перед передачей заменяются их дополнениями по модулю 2 с тем расчетом, чтобы код обладал свойствами, необходимыми для обнаружения ошибок в полностью асимметричном канале.

Описание кода с суммированием, обнаруживающего пакеты ошибок

Пусть слово W из k двоичных знаков необходимо закодировать таким образом, чтобы при приеме был обнаружен любой пакет ошибок длиной до l, который может внести произвольный канал, а также любые ошибки, которые может внести полностью асимметричный канал. Условимся, что первым передается крайний справа знак слова W и что вслед за этим словом передается проверочная сумма, начиная с младшего значащего разряда. В этом случае

[1]) Обнаружение любых сочетаний ошибок, которые возможны в полностью асимметричном канале (в том числе пакета ошибок, длина которого равна длине кодового слова), обеспечивается и в том случае, когда код с суммированием не обнаруживает ошибки в l проверочных знаках, добавленных для обнаружения пакета ошибок в произвольном канале. Для обеспечения минимальной избыточности и максимальной эффективности проверочные знаки каждого из кодов следует формировать независимо от проверочных знаков второго кода, ориентируясь только на информационные знаки.

первому слева знаку слова W приписывают вес 2^{l-1}, следующему — 2^{l-2} и т. д. вплоть до веса 2^0. Затем повторяют то же чередование весов до тех пор, пока не будут исчерпаны все знаки слова W. Проверочную сумму образуют путем простого сложения весов, соответствующих позициям, на которых расположены знаки 1, с преобразованием полученной суммы в двоичную систему счисления. Перед передачей каждый знак проверочной суммы дополняется по модулю 2, а на приемной станции осуществляется обратная операция с тем, чтобы принимаемую проверочную сумму можно было сопоставить с суммой весов, вычисленной по принимаемым информационным знакам. Как можно видеть из работы [1], структура кодов с суммированием такова, что если для взвешивания информационных знаков применяется любое множество ненулевых весов и если передаваемые проверочные знаки представляют собой дополнения (по модулю 2) знаков суммы этих весов, то всегда можно добиться совершенного обнаружения ошибок в асимметричном канале. Поэтому остается только показать, что описанный выше способ выбора весов обеспечивает обнаружение любого пакета ошибок длиной до l.

Прежде всего заметим, что поскольку проверочные знаки являются двоичным представлением образованной суммы, то они сами оказываются соответствующим образом взвешенными. Таким образом последние l проверочных знаков имеют веса, соответствующие первому из повторяющихся сочетаний весов информационных знаков. Далее можно заметить, что любой пакет ошибок, длина которого не превышает l, охватывает двоичные знаки, веса которых являются различными степенями числа 2. Таким образом, ни при каких пакетах ошибок длиной до l изменения суммы за счет ошибок в различных знаках не могут скомпенсировать друг друга, и такие ошибки не могут остаться необнаруженными.

Приведенные выше рассуждения можно пояснить следующим простым примером. Пусть количество информационных знаков $k = 12$. Обозначим эти знаки $i_1, i_2, \ldots, i_{12}$. Пусть необходимо обнаруживать любой пакет ошибок длиной до 3. Тогда следует использовать веса 4, 2 и 1. При этом потребуется 5 проверочных знаков; обозначим

их c_1, c_2, ..., c_5, причем c_5 будет младшим значащим разрядом. Принимаемое слово после замены каждого принимаемого проверочного знака его двоичным дополнением по модулю 2 будет иметь вид

$$\overset{4\ 2\ 1\ 4\ 2}{c_1c_2c_3c_4c_5}\quad \overset{\}1}{i_1i_2i_3i_4i_5\ \ldots\ i_{12}}$$

(цифры над информационными знаками указывают их веса). Очевидно, что любой пакет ошибок, длина которого не превышает 3 и который охватывает только информационные знаки, изменит значение суммы, образуемой по приведенным выше правилам. В результате этого проверочная сумма, вычисленная по принимаемым информационным знакам, будет отличаться от принимаемой проверочной суммы. Например, если знаками $i_2i_3i_4$ являлись соотвественно знаки 100 и если пакет ошибок преобразовал их в знаки 011, то сумма увеличится на 3. Аналогичные рассуждения можно провести и для случая, когда пакет ошибок охватывает только проверочные знаки. (Заметим, что разрядам c_1 и c_2 соответствуют веса 16 и 8.) Действительно, любой пакет ошибок, длина которого не превышает полного количества проверочных знаков и который охватывает только проверочные знаки, будет обнаружен. Наконец, если пакет ошибок длиной до 3 охватывает часть проверочных и часть информационных знаков, то изменятся обе суммы, но эти изменения не могут быть одинаковыми. Например, если пакет ошибок охватывает знаки c_4, c_5 и i_1, то поскольку вес c_4 равен 2, а вес c_5 равен 1, то принимаемая проверочная сумма может измениться (по сравнению с переданной) на ± 1, ± 2 или ± 3, в то время как сумма, вычисленная по принимаемым информационным знакам, может измениться только на ± 4.

Рассматриваемый код, как и циклические, позволяет обнаружить также многие пакеты ошибок, длина которых превышает l. Так, в приведенном выше примере из 224 возможных сочетаний четырех смежных двоичных знаков только 40 могут быть изменены пакетом ошибок длиной 4 без обнаружения этих изменений. Аналогичным образом, рассматриваемый код обеспечивает большую вероятность обнаружения независимых многократных пакетов ошибок,

чем в случае применения простых кодов с суммированием. Это объясняется введением взвешивания.

Очевидно, что количество избыточных двоичных знаков, которое должен иметь рассматриваемый код, равно наименьшему из целых чисел, превышающих $\log_2$ суммы всех возможных весов информационных знаков. Выражая соотношение между количеством информационных знаков k и максимальной длиной пакета ошибок l с помощью алгоритма деления

$$k = ql + g, \quad 0 \leqslant g < l,$$

легко видеть, что необходимое количество избыточных знаков r равно наименьшему из целых чисел, превышающих

$$\log_2 [q(2^l - 1) + 2^l (1 - 2^{-g})].$$

В большинстве случаев достаточно точную оценку дает приближенная формула

$$r \approx [l + \log_2 (q + 1)],$$

где квадратные скобки обозначают целую часть. Таким образом, для совершенного обнаружения ошибок в полностью асимметричном канале достаточно к тем l проверочным знакам, которые необходимо иметь в случае циклического кода, добавить только относительно небольшое количество проверочных знаков, равное примерно $\log_2 (k/l)$.

ЛИТЕРАТУРА

1. Berger J. M., A note on error detection codes for asymmetric channels. *Information and control*, **4** (1961), № 1, 68. См. русский перевод в настоящем сборнике, стр. 107—115.

О ПОСТРОЕНИИ ГРУППОВЫХ КОДОВ [1])

Р. Б. Банерджи

Задача построения систематических кодов, исправляющих ошибки, ставится следующим образом: построить такой групповой код, чтобы каждое слово, представляющее возможный набор ошибок, который исправляется кодом, принадлежало к различным классам смежности.

Описывается вычислительный метод, с помощью которого создаются коды любой заданной длины, исправляющие любые заданные наборы ошибок. Метод, предложенный Саксом [8], оказывается частным случаем описываемого здесь способа, когда множество исправляемых ошибок является множеством всех возможных ошибок кратности до e.

Построены коды, содержащие до 10 проверочных символов и исправляющие двойные и тройные ошибки, а также пакеты ошибок длиной до трех символов. Для вычисления каждого кода на вычислительной машине типа LGP-30 требуется от 3 до 4 часов. Полученные коды сравниваются с другими известными кодами.

I. Введение

В предыдущем кратком сообщении (см. Банерджи [2]) автор наметил в общих чертах способ построения систематических кодов с проверкой на четность, которые исправляют все произвольно выбранные ошибки.

В настоящем сообщении разъясняется метод и дается его теоретическое обоснование. Описаны также некоторые коды, полученные с помощью вычислительной машины LGP-30. Эти коды исправляют двойные и тройные ошибки и пакеты ошибок длиной до 3 символов. Рассматриваются коды различной длины, от наиболее коротких до таких, которые могут быть построены с использованием до 10 проверочных символов. Последнее ограничение обусловлено

---

[1]) Banerji R. B., On constructing group codes, *Information and Conirol*, 4 (1961), № 1, 1—14.

методом поиска, который мы использовали для ускорения выполнения программы.

Мы рассмотрим связь нашего метода построения кодов с методами некоторых других авторов. Кроме того, мы сравним полученные коды с кодами той же длины, найденные другими способами.

II. Теория групповых кодов

В этом разделе мы рассмотрим основы теории систематических кодов и ее применение для построения кодов. Чтобы сделать изложение более полным, мы процитируем ранее полученные результаты автора.

Систематический код длины n определяется как множество двоичных слов длины n, в которых различные символы могут быть разделены на два класса: информационные символы (числом k) и проверочные. Во всех словах множества информационным и проверочным символам соответствуют одни и те же позиции. Код, в котором информационные символы принимают значения 1 или 0, содержит 2^k слов. Значения проверочных символов определяются как суммы по модулю два некоторых фиксированных информационных символов. Правила, по которым определяются проверочные символы, называются правилами проверки на четность.

Каждому слову длины n мы можем поставить в соответствие слово длины $n-k$, называемое проверочной последовательностью [1]). Это делается следующим образом.

Для всех j мы образуем сумму (по mod 2) j-го проверочного символа и символа, образованного из информационных символов j-м проверочным правилом. Сумма помещается в j-й позиции слова. Получаемое в результате двоичное слово длиной $n-k$ (в котором на каждое проверочное правило приходится один символ) является проверочной последовательностью.

Слепян [9] показал, что члены любого систематического кода образуют группу по сложению по модулю 2. Кроме того, он показал, что если группа всех слов длины n разложена на систематический код и его классы смеж-

[1]) Проверочную последовательность часто называют синдромом.— *Прим. ред.*

ности, то все слова, находящиеся в одном классе смежности приводят к одной и той же проверочной последовательности.

Если любой элемент некоторого класса смежности сложить (по mod 2) с элементом систематического кода, то в результате получится другой элемент того же класса смежности. Пусть имеется система связи, которая передает только элементы систематического кода. Если во время передачи появляется ошибка, то такая ошибка может быть представлена словом (называемым набором ошибок) длины n, в котором в тех позициях, где имеется ошибка, стоят единицы, а в остальных — нули. Тогда принятая последовательность является суммой передаваемого слова и слова, представляющего ошибку. Образовав для принятого слова проверочную последовательность, мы можем найти класс смежности, в котором находится набор ошибок. Однако действительная ошибка не может быть выделена с помощью проверок на четность, если мы не будем уверены в том, что только один из членов класса смежности может представлять возможный набор ошибок. При такой уверенности мы можем установить взаимно однозначное соответствие между проверочными последовательностями и ожидаемыми ошибками.

Поэтому для того чтобы групповой код исправлял все ошибки из некоторого их множества, необходимо и достаточно, чтобы отсутствовали классы смежности, содержащие больше одного элемента из множества возможных ошибок. Таким образом, код для исправления единичной ошибки должен иметь в любом классе смежности не больше одного слова, содержащего единственную единицу. Аналогично код, исправляющий двойные ошибки, не должен иметь в любом классе смежности более одного слова, содержащего две или меньше единицы. Если исправлению подлежат пакеты ошибок, длина которых не превышает трех разрядов, то в каждом классе смежности должно иметься не более одного слова, содержащего две единицы, разделенных не более чем одним нулем[1]).

[1]) Точнее, в каждом классе смежности не должно содержаться более одного слова, состоящего из одной единицы, из двух единиц, стоящих рядом или разделенных одним нулем, и из трех стоящих рядом единиц.— *Прим. ред.*

До сих пор не имеется описанного в литературе метода (за исключением „разумного истощения"), который позволил бы образовать групповой код для исправления любого множества ошибок[1]). Тем не менее такой метод может быть разработан на основе взаимно однозначного соответствия между проверочными последовательностями и классами смежности кода. Рассматривая это соответствие, Слепян отметил, что проверочная последовательность суммы (по mod 2) двух слов является суммой (по mod 2) их проверочных последовательностей. Аналогичное утверждение имеется у Файра [5]. Этот факт устанавливает изоморфизм между группой из 2^{n-k} проверочных слов и факторгруппой группы всех слов длины n по подгруппе кодовых комбинаций.

Наше требование к групповому коду, изложенное выше, теперь может быть сформулировано в следующей форме, не зависящей от самого группового кода. Проверочные последовательности, получаемые в соответствии с правилами проверки на четность для систематического кода, должны удовлетворять следующему условию: разные ошибки должны иметь различные проверочные последовательности. Для инженеров это утверждение, по-видимому, является тривиальным. Следующий пример покажет, однако, что оно является не только постановкой проблемы, но также и частичным ее решением. Возьмем, в качестве примера код Хэмминга [7] длины 7, исправляющий единичную ошибку. Используя вышеприведенное утверждение, мы можем установить соответствие между возможными единичными ошибками и семью возможными проверочными последовательностями (см. табл. I).

Получив проверочные последовательности для всех ошибок, можно легко установить правила проверок; это может быть сделано способом, предложенным Хэммингом. Отметим, что так как ошибки в 1-й, 3-й, 5-й, 7-й позициях изменяют первый проверочный символ, то эти позиции должны быть включены в первое проверочное уравнение. Применяя аналогичные соображения ко второму и третьему проверочным символам, мы получим проверочные уравнения Хэмминга.

[1]) Последняя статья Цзяня [4] близко примыкает к этой проблеме.

Естественно, что случаи, когда исправляются многократные ошибки, уже не так просты. Поскольку многократная ошибка является суммой (по mod 2) единичных ошибок, выбор проверочных последовательностей для единичных ошибок автоматически определяет проверочные последовательности для многократных ошибок.

Например, выбор 001, 010 и 011, который мы сделали в приведенном выше примере для проверочных последовательностей, получаемых от ошибок в первой, второй и третьей позициях слова из семи символов, не может иметь места, если мы применяем код, исправляющий двойную ошибку, так как проверочная последовательность 011 в этом случае получалась бы от ошибок в первой и третьей позициях. Следовательно, выбор этой последовательности для представления ошибки в третьем символе приведет к тому, что ошибки вида 100 и 011 будут находиться в одном классе смежности.

Рассмотрим задачу более формально.

Допустимыми мы будем называть такие наборы ошибок, которые могут быть исправлены, т. е. наборы, которые находятся в различных классах смежности.

Таблица I

Проверочные последовательности для кода, исправляющего единичную ошибку

Значение ошибки	Проверочная последовательность
0000001	001
0000010	010
0000100	011
0001000	100
0010000	101
0100000	110
1000000	111

Множество ошибок будет называться „независимым“, если ни один из его членов не может быть получен суммированием элементов некоторого подмножества оставшегося множества.

Говорят, что множество наборов ошибок „покрывает" допустимые ошибки, если это множество независимо и если допустимые наборы ошибок могут быть получены суммированием элементов покрывающего множества. (Элементы покрывающего множества не обязательно должны быть допустимыми наборами ошибок.)

Пусть A — множество всех допустимых наборов ошибок $(X_1, X_2, \ldots, X_\alpha)$, которое покрыто множеством $(Y_1, Y_2, \ldots, Y_\beta)$, $(\beta \leqslant \alpha)$. Пусть X_i получается из покрывающего множества с помощью уравнений

$$X_i = A_{i1}Y_1 \oplus A_{i2}Y_2 \oplus \ldots A_{i\beta}Y_\beta, \quad 1 \leqslant i \leqslant \alpha, \qquad (1)$$

где каждое A_{ij} есть либо 1, либо 0.

Говорят, что допустимый набор ошибок X_i „порождается" множеством $(Y_1, Y_2, \ldots, Y_k)$ покрывающих наборов $(k \leqslant \beta)$, если $A_{ij} = 0$ для всех $j > k$.

Рассмотрим множество $(X_{i_1}^{(j)}, X_{i_2}^{(j)}, \ldots, X_{i_m}^{(j)})$ допустимых наборов ошибок, образованных множеством $(Y_1, \ldots, Y_k)$ покрывающих наборов, таких, что имеет место равенство $A_{i_p j} = 1$ для каждого $p\,(1 \leqslant p \leqslant m)$ в уравнении (1). Множество $\alpha_{i_p}^{(j)} = X_{i_p}^{(j)} \oplus Y_j$ называется множеством наборов ошибок, „относящихся к", Y_j. Эти $\alpha_{i_p}^{(j)}$ не обязательно должны быть допустимыми наборами ошибок.

Основной нашей задачей является нахождение множества $(Z_1, Z_2, \ldots, Z_\alpha)$ проверочных последовательностей, соответствующих множеству $(X_1, X_2, \ldots X_\alpha)$ допустимых ошибок. Мы предлагаем делать это посредством нахождения множества $(W_1, W_2, \ldots, W_\beta)$ проверочных последовательностей, соответствующих членам покрывающего множества $(Y_1, Y_2, \ldots, Y_\beta)$.

Пусть множество последовательностей $(W_1, W_2, \ldots, W_k)$ $(k < \beta)$ соответствует множеству $(Y_1, Y_2, \ldots, Y_k)$ покрывающих (наборов) ошибок.

Обозначим через $(W_{i_1}, W_{i_2}, \ldots, W_{i_m})$ проверочные последовательности, соответствующие ошибкам $(X_{i_1}, X_{i_2} \ldots, X_{i_m})$ допустимых наборов ошибок, порожденных $(Y_1, Y_2, \ldots, Y_k)$, а через $(W_{j_1}^{(k+1)}, W_{j_2}^{(k+1)}, \ldots, W_{j_n}^{(k+1)})$ — проверочные последовательности, соответствующие множеству

наборов ошибок, относящихся к следующему элементу Y_{k+1} покрывающего множества. Тогда, если W_{k+1} является проверочной последовательностью, соответствующей Y_{k+1}, мы имеем

$$W_{k+1} \neq W_{i_p} \oplus W_{j_q}^{(k+1)} \quad \text{для каждого } (p, q). \tag{2}$$

Предположим противное, т. е. пусть

$$W_{k+1} = W_{i_p} \oplus W_{i_q}^{(k+1)},$$

или

$$W_{k+1} \oplus W_{j_q}^{(k+1)} = W_{i_p}, \tag{3}$$

Последовательность $W_{j_q}^{(k+1)}$ соответствует образцу ошибки вида $\alpha_{j_q}^{(k+1)} = X_{j_q}^{(k+1)} \oplus Y_{k+1}$, где $X_{j_q}^{(k+1)}$ является допустимой ошибкой, образованной множеством $(Y_1, Y_2, \ldots, Y_{k+1})$. Отсюда $W_{j_q}^{(k+1)} \oplus W_{k+1}$ в уравнении (3) соответствует $X_{j_q}^{(k+1)}$, набору ошибок, образованному $(Y_1, Y_2, \ldots, Y_{k+1})$ и включающему Y_{k+1}. Аналогично W_{i_p} соответствует допустимому набору ошибок, образованному исключительно с помощью $(Y_1, Y_2, \ldots, Y_k)$. Так как эти два набора ошибок различны, из уравнения (3) следовало бы тождество проверочных последовательностей для двух допустимых ошибок, что не должно иметь места. Поэтому должно быть верным уравнение (2).

Достаточность условия доказывается следующим образом.

Пусть имеются два набора ошибок X и X^1, проверочные последовательности которых различны. Пусть X имеет вид $\alpha_{j_q}^{(k+1)} + Y_{k+1}$, где $\alpha_{j_q}^{(k)}$ относится к Y_k. Пусть $W_{i_q}^{(k+1)}$ — проверочная последовательность, соответствующая $\alpha_{j_q}^{(k+1)}$, и W_{i_p} соответствует X^1. Тогда должно быть $W_{i_q}^{(k+1)} + W_{i_p} \neq W_{k+1}$, откуда следует уравнение (2).

С помощью уравнения (2) можно обосновать следующий метод.

1. Выберем множество Z линейно независимых наборов ошибок, которое покрывает множество допустимых наборов ошибок. Выберем две произвольные проверочные последовательности, соответствующие двум наборам ошибок в Z.

Обозначим это множество проверочных последовательностей через B и множество произвольно выбранных элементов из Z — через A.

2. Путем суммирования элементов множества B по mod 2 образуем множество проверочных последовательностей, соответствующих всем допустимым наборам ошибок, которые покрываются множеством A. Прибавим это множество к B.

3. Выберем из Z другой элемент и образуем проверочные последовательности наборов ошибок, которые покрываются множеством A и связаны с этим элементом. Обозначим это множество через C.

4. Образуем множество C^* суммированием по mod 2 элементов B и C.

5. В качестве представителя следующего элемента, выбираемого из Z, возьмем последовательность, не принадлежащую B и C^*. Добавим эту последовательность к множеству B, а новый элемент из Z — к A.

6. Уничтожим множества C и C^*. Будем повторять последовательно этапы 2—6 до тех пор, пока все члены Z не попадут в A. На этом этапе B должно содержать проверочные последовательности, соответствующие каждой допустимой ошибке.

III. Упрощение метода в специальных практически важных случаях

Согласно правилам, изложенным в конце предыдущего раздела, множество Z может быть выбрано как множество всех единичных ошибок. В том случае, когда все допустимые ошибки являются многократными ошибками, множество C образует подмножество B.

В коде, исправляющем e ошибок, B является множеством проверочных последовательностей всех ошибок, имеющих e или меньше единиц, в то время как C является множеством таких последовательностей для ошибок, имеющих не больше $e-1$ единиц. Отсюда исключению подлежит множество всех проверочных последовательностей, соответствующих ошибкам, имеющим $2e-1$ или меньше единиц. Следует заметить, что это совпадает с критерием Сакса [8], при котором в коде, исправляющем

e ошибок, каждое множество из $2e$ характеристик (проверочных последовательностей) является линейно независимым. Следовательно, выполнение этого критерия автоматически удовлетворяет теореме 1 Боуза и Чоудхури [3].

Однако способ, которым Сакс использовал этот критерий, существенно отличается от нашего и в приведенном им примере получен код, в некотором смысле менее эффективный, нежели тот, который получен при использовании нашего метода.

В то время как метод Сакса не дает результатов, если ошибки имеют произвольную структуру, наш метод легко применим для исправления пакетов ошибок, описанных Абрамсоном [1], Файром [5] и Хагельбергером [6]. В случае исправления пакетов ошибок длины l множество B, рассматриваемое в предыдущем разделе будет соответствовать пакетам ошибок длины l (т. е. таким, у которых не имеется двух единиц, разделенных более чем $l-1$ символами); множество C будет в этом случае соответствовать всем ошибкам, имеющим меньше e единиц, обладающих тем свойством, что не имеется ошибок на расстоянии, большем чем $l-1$ от следующего члена Z, который включается в A. Развивая метод в этом направлении, можно получить коды, несколько более эффективные, чем те, которые получены Файром.

Другое преимущество этого метода состоит в том, что, когда образуются коды длины n, можно получить все аналогичные более короткие коды. Очень удобный способ для этого состоит в выборе первых двух членов A как наборов ошибок в первой и второй позициях. Последующие члены, которые поступают в A, выбираются как ошибки в последующих символах. Тогда на каждом этапе вычисления A содержит единичные ошибки в последовательных позициях слова. Если мы остановимся на K-м этапе, мы получим код длины K.

В следующем разделе мы подробно опишем процесс построения кодов, исправляющих пакеты ошибок длины 3. Затем мы опишем результаты вычисления с помощью машины и сравним полученный код с кодом Файра. Наконец, мы опишем полученные нашим методом коды, исправляющие двойные и тройные ошибки, и сравним их с кодами Слепяна, Сакса и Боуза—Чоудхури.

IV. Некоторые примеры

Метод, описанный в разд. III, следующим образом используется для образования проверочных последовательностей для пакетов ошибок длины 3 или меньше. Сначала выбираем 00...01 и 00...10 в качестве проверочных последовательностей для единичной ошибки в первом и втором символах. Пакеты ошибок, покрывающиеся этими двумя векторами, включают единичные ошибки, а также двойную ошибку в первых двух позициях; при этом двойная ошибка имеет проверочную последовательность 00 ... 011. Ошибка в третьем символе имеет проверочную последовательность 00 ... 0100. Аналогичные рассуждения показывают, что ошибке в четвертом символе соответствует проверочная последовательность 0 ... 01000, так как все последовательности с единицей в первых трех позициях являются проверочными последовательностями для допустимых ошибок, т. е. принадлежат множеству B.

Разыскивая проверочную последовательность для ошибки в пятой позиции, мы должны построить множество C, которое в этом частном случае состоит из проверочной последовательности для единичных ошибок в третьей или четвертой позициях или для двойной ошибки в обеих этих позициях. Суммируя множества B и C, образуем C^*, которое совместно с B исчерпывает все наборы из четырех символов. Таким образом для единичной ошибки в пятом символе мы вынуждены взять проверочную последовательность 0 ... 010000.

Исследуя проверочную последовательность для единичной ошибки в шестом символе, мы включим в B проверочные последовательности пакетов ошибок во 2-й, 3-й и 4-й позициях, а так же в 1-й, 2-й, 3-й позициях. Будут созданы новые C и C^*. Этим исчерпываются все пятизначные наборы, и проверочная последовательность для ошибки в шестой позиции имеет вид 0 ... 0100000.

Множества B, C и C^* для седьмой позиции представлены в табл. II, в которой пропущены повторяющиеся и приводящие к повторениям последовательности.

Следует заметить, что многие последовательности между 0 ... 0 и 1 ... 1 не входят в B и C^*. В качестве

проверочной последовательности для ошибки в седьмой позиции мы выберем последовательность, которая отсутствует в B и C^* и представляет собой наименьшее возможное двоичное число; любой другой подходящий выбор приведет нас к эквивалентному коду. Такой последовательностью является 0 ... 01001.

На этом этапе важно заметить, что до сих пор каждая проверочная последовательность, которую мы строили, имела единственную единицу и, следовательно, для каждого сообщения длиной 6 необходимо было использовать 6 проверочных символов. Теперь снова имеется проверочная последовательность, которая состоит из 6 символов. Поэтому отпадает необходимость в дополнительном проверочном символе для проверки седьмого элемента и таким образом в код вводится одна лишь информационная единица. Следовательно, мы имеем обычный семиразрядный код 0000000 и 1111111. Поэтому здесь нет разницы между корректирующим кодом для тройной ошибки и пакета ошибок.

Этот процесс может быть продолжен. На любом этапе построения мы получаем код, исправляющий пакеты ошибок. Нами получены проверочные последовательности для 30-значного кода, имеющего 8 проверочных символов. Эти последовательности приведены в табл. III. Интересно заметить, что получение приведенных 30 проверочных последовательностей потребовало 4 часа работы вычислительной машины LGP-30 (время сложения 1,7 *мсек*).

Файр получил 16-значный код, который имеет 9 информационных символов и исправляет все пакеты ошибок длиной 3. Наш код обладает теми же возможностями. Однако наш метод показывает, что при 7 проверочных символах можно повысить число информационных символов в коде при сохранении его корректирующих свойств. Процедура для построения группового кода и правил проверки на четность та же, что и у Файра, однако нужно заметить, что вместо того, чтобы все проверочные символы собрать вместе, мы расположили их между информационными; при этом для определения проверочных правил не требуется дополнительных алгебраических преобразований.

Таблица II

Типичное положение при построении кодов, исправляющих пакеты ошибок

C	*B*	*C**
0 ... 0100000	0 01	0 ... 0100001
0 ... 0010000	0 10	0 ... 0100010
0 ... 0110000	0 100	0 ... 0100100
	0 1000	0 ... 0101000
	0 011	0 ... 0100011
	0 0110	0 ... 0100110
	0 101	0 ... 0100101
	0 1100	0 ... 0101100
	0 1010	0 ... 0101010
	0 0111	0 ... 0010001
	0 ... 01110	0 ... 0010010
		0 ... 0010100
		0 ... 0011000
		0 ... 0010011
		0 ... 0010110
		0 ... 0010101
		0 ... 0011100
		0 ... 0011010
		0 ... 0110001
		0 ... 0110010
		0 ... 0110100
		0 ... 0111000
		0 ... 0110011
		0 ... 0110110
		0 ... 0110101
		0 ... 0111100
		0 ... 0111010
		0 ... 0100111
		0 ... 0101110
		0 ... 0111110

В качестве примера рассмотрим (16,9)-код, который получается при усечении табл. III на шестнадцатом сим-

Таблица III

Код, исправляющий пакеты ошибок

Позиция ошибки	Проверочная последовательность
1	00000001
2	00000010
3	00000100
4	00001000
5	00010000
6	00100000
7	00001001
8	00010010
9	00100100
10	01000000
11	00001011
12	00010001
13	01000001
14	00001111
15	00100011
16	01000010
17	00001101
18	01000111
19	01010011
20	10000000
21	00010101
22	00100001
23	01001000
24	10000001
25	00011101
26	01000100
27	10000011
28	00110001
29	00010111
30	10000100

воле. Если мы первые шесть и 10-й символы возьмем в качестве проверочных символов P_i (они характеризуются тем, что их проверочные последовательности содержат

по одной единице, а остальные — в качестве информационных символов D_i, мы будем иметь следующие проверочные правила:

$$P_1 = D_1 + D_4 + D_5 + D_6 + D_7 + D_8,$$
$$P_2 = D_2 + D_4 + D_7 + D_8 + D_9,$$
$$P_3 = D_3 + D_7,$$
$$P_4 = D_1 + D_4 + D_7,$$
$$P_5 = D_2 + D_5,$$
$$P_6 = D_3 + D_8,$$
$$P_7 = D_6 + D_9.$$

Для получения этих правил необходимо считывать только столбцы таблицы проверочных последовательностей.

В табл. IV приведены проверочные последовательности для кода, исправляющего двойные ошибки, а в табл. V — для кода, исправляющего тройные ошибки. Мы сравним коды, полученные нами, с кодами, полученными ранее другими авторами.

Одним из наиболее интересных фактов является то, что 29-значный код для исправления двойных ошибок имеет 19 информационных символов. Сакс, используя почти тот же способ образования кода, получил при том же числе информационных символов 32-значный код. Причина расхождения, по-видимому, кроется в различии способов последовательного построения проверочных последовательностей. В своем примере Сакс исходил из пессимистического предположения о необходимом числе проверочных символов вместо того, чтобы строить коды постепенно с выбором минимально возможного двоичного числа для представления проверочных последовательностей.

Сравним теперь результаты Слепяна и Боуза — Чоудхури с нашими результатами для кодов, исправляющих двойные и тройные ошибки.

(5,1)-код, исправляющий двойную ошибку, получается усечением таблицы IV на пятой строке и является тривиальным. Как показал Слепян (см. [9], табл. II), следующим полезным кодом (кроме (6,1)-кода и (7,1)-кода) является (8,2)-код, который получается усечением нашей таблицы IV на 8-й строке.

Следующие коды [(10,3) и (11,4)] также совпадают со слепяновскими. Однако наш метод дает и другие коды, исправляющие двойные ошибки.

Таблица IV

Код для исправления двойной ошибки

Позиция ошибки	Проверочная последовательность
1	0000000001
2	0000000010
3	0000000100
4	0000001000
5	0000001111
6	0000010000
7	0000100000
8	0000110011
9	0001000000
10	0001010101
11	0001101010
12	0010000000
13	0010010110
14	0010110101
15	0011011011
16	0011101101
17	0011110111
18	0100000000
19	0100010111
20	0100101001
21	0110111101
22	1000000000
23	1000011001
24	1000101101
25	1001010010
26	1010000011
27	1100100011
28	1101011111
29	1111100110

Таблица V

Код для исправления тройной ошибки

Позиция ошибки	Проверочная последовательность
1	0000000001
2	0000000010
3	0000000100
4	0000001000
5	0000010000
6	0000100000
7	0000111111
8	0001000000
9	0010000000
10	0100000000
11	0110111101
12	1000000000
13	1011011001
14	1101101010
15	1110110100

Интересно заметить, что хотя наш метод дает коды, эквивалентные тем, которые даны в таблице Слепяна, все же даже поверхностное сравнение некоторых из них показывает существенное различие.

Рассмотрим теперь коды для исправления тройных ошибок из табл. V и сопоставим их с результатами Слепяна. Кроме тривиального (7,1)-кода, мы получаем нашим методом коды (11, 2), (13, 3), (14, 4), (15, 5). Первый из них приведен Слепяном. Последний совпадает с (15,5)-кодом, данным Боузом и Чоудхури. Коды (13, 3) и (14,4) могут быть получены из последнего вычеркиванием столбцов. Этот процесс, если его продолжить еще на один шаг, приводит к (12,2)-коду, который менее эффективен, чем известные.

V. Некоторые замечания о программах для вычислительной машины

Все программы для получения кодов, описанных в предыдущем разделе, были составлены для вычислительной машины LGP-30. К сведению тех, кто будет использовать ту же самую машину, мы считаем систему автоматического программирования Act-I весьма удобной для наших целей.

Так как LGP-30 не имеет команды суммирования по mod 2, эта операция была специально запрограммирована. Чтобы получить сумму по mod 2 двух чисел A и B, мы использовали формулы

$$A+B\equiv\overline{(\overline{A}\,\&\,\overline{B})\,\&\,\overline{AB}}$$

и

$$A+B\equiv\overline{AB}A+\overline{AB}B,$$

которые, конечно, логически эквивалентны. В машине LGP-30 операция конъюнкции совпадает с командой „выделения“. Операция „отрицания“ лучше всего моделируется с помощью изменения знака (путем взятия дополнения) с последующим циклическим переносом из младшего разряда в старший.

Очень важным условием успешного использования программы является ее связь с методом хранения и поиска информации. Так как мы интересовались нахождением наименьшего числа, не содержащегося в таблице, мы сочли удобным сначала оставить незаполненными два блока памяти и хранить числа множеств B и C^* в той последовательности, как они появляются. Другими словами, если обозначить i-ю ячейку блока B через B_i, то двоичное число i направляется в B_i. Когда начинается поиск наименьшего элемента, мы просматриваем весь блок от начала до конца; найдя наименьшее число k, очищаем ячейку B_k.

Мы были ограничены памятью машины (4000 слов), в которой можно было хранить 1024 проверочные последовательности, т. е. 10 проверочных символов. Мы пытались обойти это затруднение путем последовательной записи в память появляющихся чисел без повторений,

и всякий раз, когда появлялась таблица, содержащая двоичные числа от 1 до k, мы вычеркивали все числа, кроме k, которое оставляли в качестве представителя вычеркнутых чисел. Таким образом, мы смогли составить программу для случая девяти проверочных цифр с помощью только пятидесяти ячеек. Однако при этом бо́льшая часть времени тратилась на вычеркивание и проверку и время работы программы увеличивалось приблизительно в 4 раза. А так как для работы обычной „быстрой" программы требовалось 3—4 часа, мы сочли такой способ экономически неприемлемым.

ЛИТЕРАТУРА

1. Abramson N. M., A class of systematic codes for non-independent errors, *IRE Trans. Inform. Theory*, **IT-2** (1959), 150.
2. Banerji R. B., A systematic method for the construction of error-correcting groups codes, *Nature*, **186** (1960), 627.
3. Bose R. C., Ray-Chaudhuri D. K., On a class of error-correcting binary group codes, *Inform. and Control*, **3** (1960), 68. [Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 2, ИЛ, М., 1961, 83.]
4. Chien R. T., Group codes for prescribed error patterns, *Proc. IRE, Convention*, March (1960).
5. Fire P., A class of multiple-error-correcting binary codes for non-independent errors, Rept. 55, SEL, Stanford Univ., 1959.
6. Hagelberger D. W., Recurrent codes easily mechanized. Burst correcting and binary codes, *Bell. System Tech. J.*, **38** (1959), 969.
7. Hamming R. W., Error-detecting and error-correcting codes. *Bell. System Tech. J.*, **29** (1950), 147. [Русский перевод: Хэмминг Р., Коды с обнаружением и исправлением ошибок, сб., ИЛ, М., 1956.]
8. Sack's G. E., Multiple error correction by means of parity checks. *IRE Trans. Inform. Theory*, **IT-4** (1958), 145.
9. Slepian D., A class of binary signalling alphabets, *Bell. System Tech.*, **35** (1956), 203—234. [Русский перевод: Слепян Д., Класс двоичных сигнальных алфавитов, в сб. Теория передачи сообщений, ИЛ, М., 1957, 82—113.]

КОДЫ С МАЛОЙ ПЛОТНОСТЬЮ ПРОВЕРОК НА ЧЕТНОСТЬ[1])

Р. Г. Галлагер

Краткое содержание. Код с малой плотностью проверок на четность определяется проверочной матрицей, удовлетворяющей следующим условиям: в каждом столбце матрицы содержится лишь небольшое фиксированное число единиц $j \geqslant 3$; также небольшое число единиц k, $k > j$ содержится и в каждой строке. При заданном j и постоянной скорости передачи кодовое расстояние для большинства таких кодов растет линейно с увеличением длины блоков. Если этими кодами пользоваться для передачи по каналу с двоичным входом, уровень шумов в котором не слишком высок, и производить декодирование по методу максимума правдоподобия, то при фиксированном j можно добиться экспоненциального убывания вероятности ошибки с увеличением длины блоков.

В статье описана простая, хотя и не оптимальная, схема декодирования, основанная на непосредственном использовании апостериорных вероятностей на выходе канала. Сложность декодирующего устройства и его быстродействие (в двоичных единицах за секунду) возрастают приблизительно линейно с увеличением длины блоков.

Далее в работе показано, что если передача по двоичному симметричному каналу ведется с достаточно малой скоростью, то вероятность ошибки при таком способе декодирования убывает экспоненциально относительно квадратного корня из длины блоков.

Результаты экспериментальной проверки показывают, что в действительности вероятность ошибки при декодировании намного меньше, чем эта теоретическая оценка.

Кодирование для передачи двоичных сообщений

Помехоустойчивое кодирование является одним из средств для повышения надежности работы систем связи. Теорема кодирования для каналов с шумами, доказываемая в работах по теории информации [1, 6] для широкого класса каналов, утверждает, что если соответствующим образом закодированную информацию передавать со скоростью, меньшей пропускной способности

[1]) Gallager R. G., Low-density parity-check codes, *IRE Transactions on Information Theory*, IT-8 (1962), № 1, 21—28.

канала, то, увеличивая длину кода, можно добиться экспоненциального убывания вероятности ошибки. Однако в теореме нет никаких указаний на зависимость между длиной кода, с одной стороны, и объемом вычислений или стоимостью оборудования, необходимого для достижения столь малой вероятности ошибки, с другой стороны.

В этой работе описывается класс кодирующих и декодирующих схем, которые удобны для работы с кодами большой длины; благодаря этому достигается малая вероятность ошибки, в то время как для реализации этих схем не требуется ни слишком сложного оборудования, ни выполнения большого объема вычислительных операций.

Коды, которые нам предстоит рассмотреть, представляют собой специальный тип кодов с проверками на четность [1]). Кодовые слова в коде с проверками на четность образуются путем присоединения блока проверочных символов к блоку информационных символов. Каждый проверочный символ равен сумме по модулю два [2]) некоторых выделенных информационных символов.

Таблица I

Пример проверочной матрицы

Информационные символы	Проверочные символы	
x_1 x_2 x_3	x_4 x_5 x_6 x_7	
1 1 1	0 1 0 0	$x_5 = x_1 \oplus x_2 \oplus x_3$
1 1 0	1 0 1 0	$x_6 = x_1 \oplus x_2 \oplus x_4$
1 0 1	1 0 0 1	$x_7 = x_1 \oplus x_3 \oplus x_4$

Для удобства эти правила образования проверочных символов можно представить в виде проверочной матрицы, изображенной на табл. I. Эта матрица задает систему линейных однородных уравнений, и кодовые слова являются просто решениями этой системы. Совокупность

[1]) Более подробно о кодах с проверками на четность см. у Слепяна [2].

[2]) Сумма по mod 2 равна 1, если число слагаемых единиц нечетно, и равна нулю, если число слагаемых единиц четно.

символов, определяющих проверочное уравнение, назовем проверочным множеством. Так, например, символы, стоящие в 1, 2, 3 и 5 позициях, образуют первое проверочное множество.

Использование кодов с проверками на четность позволяет осуществлять кодирование (в отличие от декодирования) довольно простыми методами. К тому же, как показал Элайес [3], если случайно выбранный код с проверками на четность использовать для передачи по двоичному симметричному каналу со скоростью, лежащей между критической скоростью и пропускной способностью канала, то вероятность ошибки при декодировании будет почти наверняка такой же, как у лучшего из возможных кодов с той же длиной блоков и с той же скоростью передачи.

К сожалению, из простоты структуры кодов с проверками на четность никоим образом не следует существование простых способов декодирования. Это и заставляет нас обратиться к специальному классу этих кодов, который описывается ниже и для которого существуют практически приемлемые способы декодирования.

Коды с малой плотностью проверок на четность

Коды с малой плотностью проверок определяются проверочной матрицей, состоящей в основном из нулей и содержащей лишь небольшое число единиц. В частности, (n, j, k)-код с малой плотностью проверок может иметь проверочную матрицу, подобную изображенной на табл. II; в этой матрице каждый столбец содержит небольшое фиксированное число j единиц и каждая строка тоже содержит небольшое фиксированное число единиц k. Заметим, что матрицы такого типа не обязаны иметь диагональ, соответствующую проверочным символам, как это имеет место в табл. I.

Тем не менее системы уравнений, определяемые этими матрицами, всегда могут быть решены при кодировании, в результате чего проверочные символы будут выражены в виде суммы информационных символов.

Эти коды не являются оптимальными (в смысле минимума вероятности ошибки) для кодов данной длины;

Таблица II

Пример проверочной матрицы кода с малой плотностью проверок; $n=20$, $j=3$, $k=4$

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

более того, можно показать, что максимальная скорость передачи, при которой можно употреблять эти коды, лежит ниже пропускной способности канала. Однако для кодов с малой плотностью проверок существуют весьма простые способы декодирования, и это обстоятельство компенсирует недостатки, связанные с их неоптимальным поведением.

Анализировать каждый отдельно взятый код с малой плотностью проверок большой длины было бы трудно ввиду чрезвычайно большого числа кодовых слов. Гораздо проще исследовать целый ансамбль таких кодов; зная статистические свойства ансамбля, можно изучать средние значения тех величин, которые невозможно вычислить для каждого отдельного кода. Исходя из свойств всего ансамбля кодов, можно сделать некоторые статистические заключения о свойствах отдельных кодов. Кроме того, код со свойствами, типичными для всего ансамбля, может быть найден с большой вероятностью путем случайного выбора из ансамбля.

Чтобы определить ансамбль (n, j, k)-кодов с малой плотностью проверок, обратимся снова к табл. II. Заме-

тим, что матрица состоит из j подматриц, содержащих по одной единице в каждом столбце. В первой из этих подматриц единицы расположены нисходящими ступенями: единицы i-й строки стоят в столбцах с номерами от $(i-1)k$ до ik. Остальные подматрицы образуются из первой простой перестановкой столбцов. Рассмотрим ансамбль (n, j, k)-кодов, полученных случайной перестановкой столбцов в каждой из $j-1$ нижних подматриц

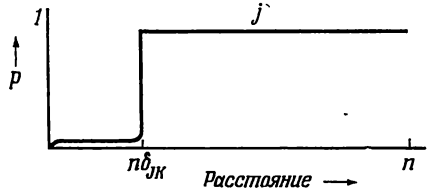


Рис. 1 График оценки функции распределения минимального расстояния.

матрицы, изображенной на табл. II; все перестановки объявим равновероятными [1]). Для этого ансамбля могут быть доказаны два интересных утверждения; первое — относительно минимального расстояния в отдельных кодах и второе — о вероятности ошибки декодирования.

Минимальным расстоянием в коде называется число позиций, в которых два ближайших кодовых слова имеют различные символы. Минимальное расстояние является случайной величиной на ансамбле кодов и ее функция распределения, как показано в работе [4], оценивается сверху функцией, представленной на рис. 1. С увеличением длины блоков при фиксированных $j \geqslant 3$ и $k > j$ изменение этой функции принимает скачкообразный вид с величиной скачка, равной единице в точке, отстоящей от нуля на расстоянии $n\delta_{jk}$. Таким образом, при больших n практически все коды ансамбля имеют минимальное расстояние, равное по меньшей мере $n\delta_{jk}$.

[1]) Может оказаться, что строки в таких матрицах будут линейно зависимы. Это означает, что в действительности существуют коды с несколько более высокой скоростью передачи, чем у кодов, соответствующих этим матрицам.

В табл. III представлено типичное для ансамбля кодов отношение минимального расстояния к длине блока, которое сравнивается с тем же отношением для случайно выбираемых кодов с проверками на четность, т. е. для кодов, у которых 0 и 1 в проверочной матрице появляются с равными вероятностями.

Следует отметить, что все неслучайные способы построения кодов, известные в настоящее время, характеризуются тем, что отношение минимального расстояния к длине блока с увеличением длины блока стремится к нулю.

Вероятность ошибки при декодировании по методу максимума правдоподобия при использовании кодов с малой плотностью проверок, очевидно, зависит от типа используемого канала. В частности, простые результаты получаются для двоичного симметричного канала (ДСК), т. е. канала без памяти с двоичным входом, двоичным выходом и с фиксированной вероятностью ошибки, т. е. перехода одного символа в другой. На примере этого канала можно показать [4], что для некоторой области значений переходных вероятностей код с малой плотностью проверок обладает вероятностью ошибки декоди-

Таблица III

Сравнение δ_{jk}, (отношения минимального расстояния в среднем (n, j, k)-коде к длине блока) с тем же отношением δ в типичном коде с проверкой на четность при той же скорости передачи

j	k	Скорость передачи	δ_{JK}	δ
5	6	0,167	0,255	0,263
4	5	0,2	0,210	0,241
3	4	0,25	0,122	0,214
4	6	0,333	0,129	0,173
3	5	0,4	0,044	0,145
3	6	0,5	0,023	0,11

рования, которая убывает экспоненциально с увеличением длины блока, и что экспонента при этом такая же, как у оптимального кода с несколько более высокой скоростью передачи (см. табл. IV).

Таблица IV

Понижение скорости передачи в коде с малой плотностью проверок

j	k	Скорость передачи	Скорость передачи для эквивалентного оптимального кода
3	6	0,5	0,555
3	5	0,4	0,43
4	6	0,333	0,343
3	4	0,25	0,266

Хотя приведенные результаты для ДСК указывают на то, что коды с малой плотностью проверок обладают свойствами, близкими к оптимальным, они предназначаются главным образом не для этого канала. ДСК является приблизительной моделью физически реального канала лишь в том случае, если имеется приемник, который производит посимвольное восстановление переданного сигнала. Описанный ниже способ декодирования может быть в действительности использован в каналах, позволяющих вычислять апостериорную вероятность, а поскольку при посимвольном восстановлении теряется много полезной информации о сигнале, то в конечном счете мы должны интересоваться вероятностью ошибки декодирования в каналах с двоичным входом и непрерывном выходом. Если входные символы подвергаются одинаковому воздействию шумов, то вероятность ошибки в этом случае, так же как и в случае ДСК, ограничена сверху функцией, экспоненциально убывающей с увеличением длины блоков, однако зависимость экспоненты от параметров канала и кода оказывается более сложной. Подобные результаты, по-видимому, имеют место и для широкого класса каналов с памятью, хотя никаких аналитических результатов здесь еще не получено. Для

каналов с памятью представляется целесообразным модифицировать ансамбль кодов, произведя, например, перестановки в первой подматрице и изменив распределение вероятностей на перестановках.

Декодирование

Введение

Далее будут описаны два способа декодирования, с помощью которых достигается разумный компромисс между сложностью декодирования и вероятностью ошибки. Первый способ чрезвычайно прост, однако может применяться только в ДСК при скоростях передачи, значительно меньших пропускной способности канала. Второй способ, основанный на непосредственном использовании апостериорных вероятностей на выходе канала, обладает бо́льшими возможностями, однако для его понимания необходимо знакомство с первым способом.

По первому способу декодирующее устройство производит вычисления согласно проверочным уравнениям и затем изменяет какой-нибудь символ, который входит в нарушенные соотношения свыше некоторого фиксированного числа раз. Эти новые значения символов используются для нового вычисления проверочных соотношений и этот процесс продолжается до тех пор, пока не будут удовлетворяться все проверочные соотношения.

Если проверочные множества малы, то такой способ декодирования оправдывает себя, так как большинство проверочных множеств либо будет содержать по одной ошибке, либо вообще не будет содержать ошибок. Следовательно, если большинство соотношений, контролирующих какой-нибудь символ, оказываются нарушенными, то это определенно указывает на ошибку в этом символе. Предположим, например, что в ходе передачи искаженным оказался первый символ кода, представленного в табл. II. Тогда будут нарушены 1, 6 и 11 соотношения, т. е. будут нарушены все три проверочных уравнения, контролирующих первый символ. Из проверочных уравнений, контролирующих другие символы в блоке, может быть нарушено не более чем одно уравнение.

Чтобы понять, как может быть исправлен некоторый символ d в том случае, когда контролирующее его проверочное множество содержит более одной ошибки, рассмотрим древовидную схему, изображенную на рис. 2.

Изобразим символ d в виде точки в основании дерева, и пусть каждая линия, выходящая из этой точки соответствует одному из проверочных множеств, в которые входит символ d. Остальные символы в этих проверочных множествах изображаются точками, расположенными

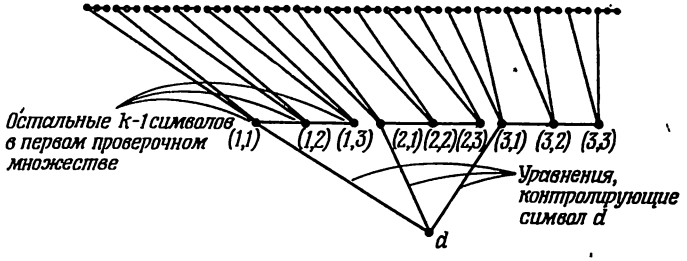


Рис. 2. Дерево, образованное проверочными множествами.

в первом ярусе дерева. Линии, идущие от первого яруса дерева ко второму, соответствуют другим проверочным множествам, в которые входят символы первого яруса, а точки второго яруса изображают остальные символы, входящие в эти проверочные множества. Следует отметить, что в ходе дальнейшего построения дерева один и тот же символ будет появляться в разных частях дерева. Это обстоятельство будет обсуждаться ниже.

Предположим теперь, что искажения при передаче произошли не только в символе d, но и в некоторых символах первого яруса. Тогда на первом этапе декодирования делается попытка исправить ошибки в первом ярусе с помощью проверочных соотношений второго яруса. В свою очередь, это даст возможность исправить символ d на втором этапе декодирования. Таким образом, символы и проверочные уравнения могут оказать помощь при декодировании символа, с которым они, на первый взгляд, не связаны. В схеме вероятностного декодирования эти дополнительные символы и дополнительные проверочные соотношения удается использовать более систематически.

Вероятностное декодирование

Предположим, что кодовые слова из (n, j, k)-кода с равными вероятностями передаются по некоторому каналу с двоичным входом. Пользуясь обозначениями, указанными на рис. 2, мы построим итерационный процесс, который позволяет на m-й итерации для любого символа d вычислить вероятность того, что этот символ равен единице при условии, что принятые символы вплоть до m-го яруса имеют фиксированные значения.

При выполнении первой итерации мы можем рассматривать символ d вместе с символами первого яруса как подкод, в котором любая совокупность символов, удовлетворяющих j проверочным уравнениям дерева, имеет одну и ту же вероятность передачи [1]).

Рассмотрим вероятностный ансамбль, в котором передаваемый символ на позиции d и символы первого яруса суть независимые двоичные единицы, а вероятности получаемых символов определяются вероятностями перехода $P_x(y)$ канала. В этом ансамбле любое сообщение имеет ту же условную вероятность, что и в определенном выше подкоде, при условии, что передаваемые символы удовлетворяют j проверочным уравнениям.

Таким образом, наша задача состоит в том, чтобы относительно этого ансамбля определить вероятность того, что символ в позиции d равен 1 при условии, что принятые символы имеют значения $\{y\}$ и что выполнено требование S, согласно которому переданные символы удовлетворяют j проверочным уравнениям, контролирующим символ d. Эту вероятность запишем в виде

$$P[x_d = 1 \mid \{y\}, S].$$

Пользуясь этим ансамблем и введенным обозначением, мы можем доказать следующую теорему.

Теорема 1. *Пусть P_d означает вероятность того, что переданный символ на позиции d равен* 1, *при условии,*

[1]) Исключение будут составлять случаи, когда некоторые из проверочных уравнений, не содержащих d, образуют проверочные множества, состоящие только из символов первого яруса. В дальнейшем эта возможность будет рассмотрена, хотя она и не представляет серьезного ограничения.

что принятый символ на позиции d имеет фиксированное значение, и пусть P_{il} означает ту же самую вероятность для l-го символа в i-м проверочном множестве первого яруса (рис. 2). Предположим, что символы взаимно независимы и что событие S состоит в том, что передаваемые символы удовлетворяют j проверочным уравнениям, контролирующим символ d. Тогда

$$\frac{P[x_d=0 \mid \{y\},\ S]}{P[x_d=1 \mid \{y\},\ S]} = \frac{1-P_d}{P_d} \prod_{i=1}^{J} \left[\frac{1+\prod_{l=1}^{k-1}(1-2P_{il})}{1-\prod_{l=1}^{k-1}(1-2P_{il})} \right]. \qquad (1)$$

Чтобы доказать эту теорему, нам потребуется следующая лемма.

Лемма 1. *Рассмотрим последовательность m независимых двоичных символов, и пусть вероятность того, что l-й символ есть единица, равна P_l. Тогда вероятность того, что число единиц в последовательности четно, равна*

$$\frac{1+\prod_{l=1}^{m}(1-2P_l)}{2}.$$

Доказательство леммы. Рассмотрим функцию

$$\prod_{l=1}^{m}(1-P_l+P_l t).$$

Заметим, что при разложении этого выражения по степеням t коэффициент при t^i равен вероятности появления i единиц. Функция $\prod_{l=1}^{m}(1-P_l-P_l t)$ обладает аналогичными свойствами, за исключением того, что коэффициенты при нечетных степенях t отрицательны. При суммировании этих двух функций коэффициенты при четных степенях удваиваются, а нечетные степени взаимно уничтожаются. Полагая $t=1$ и деля сумму на 2, получим вероятность четного числа единиц.

Так как

$$\frac{\prod_{l=1}^{m}(1-P_l+P_l)+\prod_{l=1}^{m}(1-P_l-P_l)}{2}=\frac{1+\prod_{l=1}^{m}(1-2P_l)}{2},$$

то лемма доказана.

Доказательство теоремы. По определению условных вероятностей,

$$\frac{P[x_d=0\,|\,\{y\},\ S]}{P[x_d=1\,|\,\{y\},\ S]}=\left(\frac{1-P_d}{P_d}\right)\left(\frac{P[S\,|\,x_d=0,\ \{y\}]}{P[S\,|\,x_d=1,\ \{y\}]}\right). \qquad (2)$$

Если $x_d=0$, то уравнение, в которое входит символ d, удовлетворяется лишь в том случае, когда остальные $k-1$ позиций в проверочном множестве содержат четное число единиц. Так как все символы в ансамбле статистически независимы, то вероятность того, что выполнены все j проверочных соотношений, равна произведению вероятностей выполнения каждого соотношения.

Ввиду леммы 1

$$P[S\,|\,x_d=0,\ \{y\}]=\prod_{i=1}^{j}\left[\frac{1+\prod_{l=1}^{k-1}(1-2P_{il})}{2}\right]; \qquad (3)$$

аналогично

$$P[S\,|\,x_d=1,\ \{y\}]=\prod_{i=1}^{j}\left[\frac{1-\prod_{l=1}^{k-1}(1-2P_{il})}{2}\right]. \qquad (4)$$

Подставляя выражения (3) и (4) в формулу (2), мы приходим к результату, указанному в теореме.

Ввиду сложности этого результата кажется трудным вычислять условные вероятности для переданного символа относительно символов, полученных в двух или более ярусах дерева, изображенного на рис. 2. Тем не менее случай с несколькими ярусами может быть сведен к случаю с одним ярусом применением простой итерации.

Рассмотрим сначала случай с двумя ярусами. Согласно теореме 1, мы можем определить вероятность того, что каждый символ в первом ярусе равен 1, условную отно-

сительно символов, полученных во втором ярусе. Единственным изменением в теореме будет то, что в первом произведении надо брать лишь $j-1$ сомножителей, так как проверочные множества, в которые входит символ d, в расчет не принимаются. Полученные вероятности затем могут быть использованы в формуле (1) для вычисления вероятности того, что переданный символ в позиции d равен 1. Справедливость этого процесса следует непосредственно из независимости новых значений P_{it} по ансамблю, относительно которого была доказана теорема 1. По индукции этот итерационный процесс может быть применен для нахождения вероятности того, что символ d равен 1, условной относительно любого числа ярусов с различными символами дерева.

Теперь можно полностью сформировать процесс декодирования для всего кода в целом. Для каждого символа и каждой комбинации из $j-1$ проверочных множеств, содержащих этот символ, по формуле (1) вычисляется вероятность того, что переданный символ равен единице, условная относительно символов, полученных в $j-1$ проверочных множествах.

Таким образом, для каждого символа получается j различных вероятностей, каждая из которых соответствует одному исключенному проверочному множеству. Далее, эти вероятности используются для вычисления по формуле (1) набора вероятностей второго порядка. Вероятность, связанная с одним символом и используемая при вычислении вероятности другого символа d, является вероятностью, найденной в ходе первой итерации без участия проверочного множества, содержащего символ d. При благоприятном развитии этого процесса вероятности, связанные с каждым символом, при увеличении числа итераций стремятся либо к 0, либо к 1 (в зависимости от значения передаваемого символа). Этот процесс действителен лишь для такого числа итерации, при котором еще выполняется предположение о независимости, сформулированное в теореме 1. Это предположение отпадает, если ветви дерева смыкаются между собой. Так как число точек в каждом следующем ярусе дерева в $(j-1)(k-1)$ раз больше, чем в предыдущем, то предположение будет нарушаться даже при малых значениях m в любых

кодах с умеренной длиной блоков. Можно, однако, пренебречь этим нарушением независимости, если допустить, что зависимости проявляются относительно слабо и до некоторой степени взаимно компенсируются. К тому же, если даже зависимость проявляется на m-й итерации, то первые $m-1$ итераций все-таки уменьшают недостоверность принятых символов. Тогда мы можем считать, что вероятности, полученные после $m-1$ итераций, описывают некоторую новую последовательность принятых символов, декодировать которую будет проще, чем первоначально полученную последовательность.

Наиболее важная особенность этой схемы декодирования состоит в том, что количество вычислительных операций, приходящихся в каждой итерации на один символ, не зависит от длины блоков. Кроме того, можно показать, что среднее число итераций при декодировании ограничено величиной, пропорциональной логарифму от логарифма длины блоков.

Для практического расчета вероятностей, согласно теореме 1, оказывается удобным представить формулу (1) в виде логарифма отношения правдоподобия. Положим

$$\ln\frac{1-P_d}{P_d}=\alpha_d\beta_d, \quad \ln\frac{1-P_{il}}{P_{il}}=\alpha_{il}\beta_{il},$$

$$\ln\left[\frac{P[x_d=0\mid\{y\},\ S]}{P[x_d=1\mid\{y\},\ S]}\right]=\alpha_d'\beta_{d'},$$

где α означает $+1$ или -1, а β означает абсолютную величину логарифма отношения правдоподобия. После некоторых преобразований формула (1) принимает вид

$$\alpha_d'\beta_d'=\alpha_d\beta_d+\sum_{i=1}^{j}\left\{\left(\prod_{l=1}^{k-1}\alpha_{il}\right)f\left[\sum_{l=1}^{k-1}f(\beta_{il})\right]\right\}, \qquad (6)$$

где

$$f(\beta)=\ln\frac{l^{\beta}+1}{l^{\beta}-1}.$$

Вычисление логарифма отношения правдоподобия по формуле (6) может выполняться для каждого символа либо последовательными сериями операций, либо путем параллельных вычислений. Вычисление сериями можно

запрограммировать для выполнения на универсальной вычислительной машине; именно таким способом получены экспериментальные результаты, приведенные в конце этой статьи. Для быстрого выполнения декодирования предпочтительнее способ параллельного вычисления. На рис. 3 представлена упрощенная блок-схема реализации этого способа.

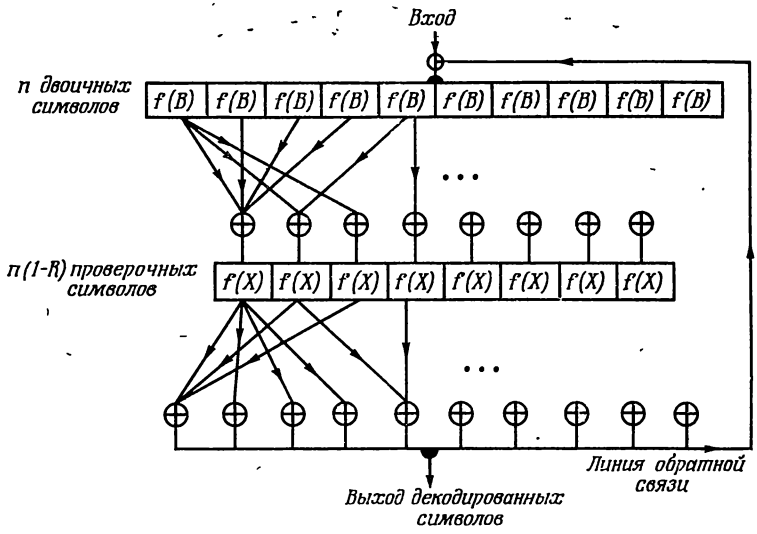


Рис. 3. Блок-схема декодирующего устройства.

На вход декодирующего устройства подается логарифм отношения правдоподобия, после чего верхний ряд блоков вычисляет величины $f(\beta)$ для каждого символа, т. е. выполняет первую операцию в формуле (6), если ее читать справа налево. На выходе сумматоров во втором ряду блок-схемы получается величина $\sum_{l=1}^{k-1} f(\beta_{il})$, что соответствует выполнению второй операции в формуле (6). Точно так же следующие ряды блоков на рис. 3 соответствуют последовательному выполнению остальных операций по формуле (6) справа налево.

На рис. 3, разумеется, опущены некоторые подробности схемы, такие, как определение знака логарифма отношения

правдоподобия и устройство, сопоставляющее каждому символу j различных значений логарифма отношения правдоподобия. Однако отсутствие этих деталей не создает трудностей для понимания всей схемы в целом.

Из приведенной блок-схемы видно, что вычислительное устройство параллельного действия обладает простой конструкцией и для его изготовления требуются аналоговые сумматоры, сумматоры по модулю 2, усилители и нелинейные цепи, аппроксимирующие функцию $f(\beta)$; число этих деталей, вообще говоря, будет пропорционально n. Правда, предстоит еще выяснить, насколько точно следует аппроксимировать функцию $f(\beta)$, однако есть основания считать, что эта проблема не является критической [1]).

Вероятность ошибки при вероятностном декодировании

Математические расчеты, связанные с вероятностным декодированием, сложны, однако нетрудно получить одну слабую оценку для вероятности ошибки.

Рассмотрим ДСК с вероятностью перехода p_0 и воспользуемся сначала (n, j, k)-кодом, в котором $j = 3$, т. е. каждый символ участвует в трех проверочных соотношениях. Рассмотрим дерево, которое образуют проверочные множества (рис. 2); оно состоит из m ярусов, которые будут нумероваться сверху вниз, причем верхний ярус — нулевой, а декодируемый символ образует m-й ярус.

Применим упрощенный способ декодирования: если нарушены оба проверочных соотношения, контролирующих данный символ первого яруса, то значение этого символа меняется на обратное; после этого символы первого яруса используются для выполнения той же операции во втором ярусе, и так далее вплоть до символа d.

Вероятность ошибки при декодировании символа d согласно этой схеме служит оценкой сверху для вероят-

[1]) Если все вычисления выполняются в двоичной форме, то согласно одной ранее проделанной экспериментальной работе можно задать $f(\beta)$ с помощью 6 двоичных единиц, причем погрешность не будет заметным образом влиять на вероятность ошибки декодирования.

ности принять неправильное решение после m-й итерации в описанной выше схеме вероятностного декодирования. Как в той, так и в другой схемах, решение принимается на основе полученных символов m-ярусного дерева, однако при вероятностном декодировании можно вывести более достоверное заключение из этой информации.

Перейдем теперь к определению вероятности того, что описанный модифицированный способ декодирования приводит к неправильному значению символа в первом ярусе. Если полученное значение символа неверно (вероятность этого p_0), то соотношение, контролирующее этот символ, будет нарушено в том и только в том случае, если в других $k-1$ символах произошло четное (включая нуль) число ошибок. По лемме 1 вероятность четного числа ошибок среди $k-1$ символов равна

$$\frac{1+(1-2p_0)^{k-1}}{2}. \tag{7}$$

Поскольку значение символа изменяется на обратное лишь в случае, если нарушены обе проверки, то вероятность того, что данный символ первого яруса получен с ошибкой и затем исправлен, равна

$$p_0\left[\frac{1+(1-2p_0)^{k-1}}{2}\right]^2. \tag{8}$$

Аналогичные рассуждения приводят к формуле

$$(1-p_0)\left[\frac{1-(1-2p_0)^{k-1}}{2}\right]^2, \tag{9}$$

определяющей вероятность того, что данный символ первого яруса был принят правильно, но затем был изменен из-за нарушения проверочных соотношений.

Из формул (8) и (9) следует, что вероятность ошибки в символе первого яруса после такого процесса декодирования равна

$$p_1 = p_0\left\{1-\left[\frac{1+(1-2p_0)^{k-1}}{2}\right]^2\right\}+ \\ +(1-p_0)\left[\frac{1-(1-2p_0)^{k-1}}{2}\right]^2. \tag{10}$$

Отсюда по индукции следует, что если вероятность ошибки при исправлении символа в i-м ярусе равна p_i, то

$$p_{i+1} = p_0 - p_0\left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^2 + \\ + (1-p_0)\left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^2. \quad (11)$$

Покажем теперь, что при достаточно малых значениях p_0 последовательность $[p_i]$ сходится к нулю. На рис. 4 представлен график величины p_{i+1} как функции p_i.

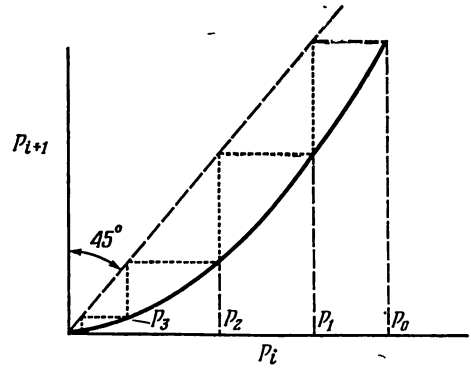


Рис. 4.

Абсцисса при данном значении i является ординатой для $i+1$-го значения, поэтому ломаная пунктирная линия изображает примерный ход графического определения p_i для последовательных значений i. Судя по графику, можно заключить, что если

$$\begin{aligned} 0 < p_{i+1} < p_i \quad & \text{для } 0 < p_i \leqslant p_0, \\ p_{i+1} = p_i \quad & \text{для } p_i = 0, \end{aligned} \quad (12)$$

то последовательность $[p_i]$ сходится к нулю. Из формулы (11) видно, что если p_0 достаточно мало, то неравенство (12) выполняется.

В таблице V приведены максимальные значения p_0 для некоторых значений k.

Скорость стремления последовательности $[p_i]$ к нулю может быть определена из формулы (11), согласно которой при малых значениях p_i

$$p_{i+1} \approx p_i 2(k-1) p_0.$$

Отсюда легко заключить, что для достаточно больших i

$$p_i \approx c[2(k-1) p_0]^i,$$

где c — константа, не зависящая от i. Так как число независимых ярусов в дереве логарифмически растет с увеличением длины блоков, то полученная оценка для вероятности ошибки при декодировании стремится к нулю с ростом длины блоков по степенному закону с небольшой отрицательной степенью.

По-видимому, столь медленная сходимость к нулю объясняется требованием строгой независимости и применением упрощенной схемы декодирования вместо полной схемы вероятностного декодирования.

Рассуждения, подобные приведенным, применимы и в случае кодов, у которых на один символ приходится более трех проверочных соотношений. Чтобы прийти к наиболее строгим результатам, проверяемый символ следует менять всякий раз, когда нарушены b или более соотношений, контролирующих этот символ,

Таблица V

Максимальные значения p_0, при которых сходится упрощенный процесс декодирования

j	K	Скорость передачи	p_0
3	6	0,5	0,04
3	5	0,4	0,061
4	6	0,333	0,075
3	4	0,25	0,106

где b целое число, подлежащее определению. Применяя этот критерий и рассуждая так же, как при выводе

формул (11), мы получаем

$$p_{i+1}=p_0-p_0\sum_{l=b}^{j-1}\binom{j-1}{l}\left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^l\left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^{j-1-l}+$$

$$+(1-p_0)\sum_{l=b}^{j-1}\binom{j-1}{l}\left[\frac{1-(1-2p_i)^{k-1}}{2}\right]^l\left[\frac{1+(1-2p_i)^{k-1}}{2}\right]^{j-1-l}. \quad (13)$$

Теперь целое число b можно выбрать так, чтобы минимизировать p_{i+1}. Решением будет наименьшее целое b, удовлетворяющее неравенству

$$\frac{1-p_0}{p_0}\leqslant\left[\frac{1+(1-2p_i)^{k-1}}{1-(1-2p_i)^{k-1}}\right]^{2b-j+1}. \quad (14)$$

Из этого условия следует, что если p_i убывает, то b тоже убывает. На рис. 5 величина p_{i+1} представлена

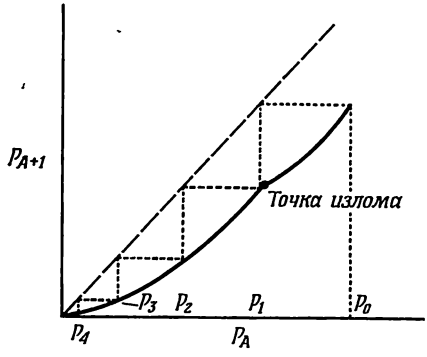


Рис. 5. Ход итераций декодирования для $j>3$.

как функция p_i при условии, что b выбирается в соответствии с неравенством (14). Точка излома на графике соответствует изменению величины b.

Доказательство того, что при достаточно малой вероятности искажения в канале вероятность ошибки декодирования стремится к нулю с увеличением числа итераций, проводится так же, как в предыдущем случае. Однако асимптотическое поведение последовательности $[p_i]$ при стремлении к нулю оказывается другим. Из

неравенства (14) следует, что при малых значениях p_i величина b равна $j/2$ для четных j и $(j+1)/2$ для нечетных j. Подставим эти значения в формулу (13), разложив ее предварительно в ряд по степеням p_i:

$$p_{i+1} = p_0 \binom{j-1}{\frac{j-1}{2}} (k-1)^{(j-1)/2} p_i^{(j-1)/2} +$$

$+$ (слагаемые высшего порядка), j нечетное;

$$p_{i+1} = p_0 \binom{j-1}{\frac{j}{2}} (k-1)^{j/2} p_i^{j/2} + \qquad (15)$$

$+$ (слагаемые высшего порядка), j четное.

С помощью этих выражений удается показать, что можно выбрать такие константы c_{jk} и столь большое i, что

$$\begin{aligned} p_i &\leqslant \exp\left[-c_{jk}\left(\frac{j-1}{2}\right)^i\right], && j \text{ нечетное;} \\ p_i &\leqslant \exp\left[-c_{jk}\left(\frac{j}{2}\right)^i\right], && j \text{ четное.} \end{aligned} \qquad (16)$$

Было бы интересно найти зависимость этих оценок от длины блоков кода. Поскольку в m-м ярусе дерева имеется $(j-1)^m (k-1)^m$ символов, то n должно по меньшей мере равняться этому числу; эта оценка соответствует левой части неравенства (17). С другой стороны, можно указать специальный процесс построения таких кодов, которые удовлетворяют правой части неравенства

$$\frac{\ln(n)}{\ln(j-1)(k-1)} \geqslant m \geqslant \frac{\ln\left(\frac{n}{2k} - \frac{n}{2j(k-1)}\right)}{2\ln(k-1)(j-1)}. \qquad (17)$$

Из неравенств (16) и (17) следует, что вероятность ошибки декодирования для кода, который удовлетворяет условию (17), ограничена сверху следующим образом:

$$P_m \leqslant \exp\left\{-c_{jk}\left[\frac{n}{2k} - \frac{n}{2j(k-1)}\right]^{\frac{\ln((j-1)/2)}{2\ln(j-1)(k-1)}}\right\},\ j \text{ нечетное;}$$

$$P_m \leqslant \exp\left\{-c_{jk}\left[\frac{n}{2k} - \frac{n}{2j(k-1)}\right]^{\frac{\ln j/2}{2\ln(j-1)(k-1)}}\right\},\ j \text{ четное.}$$

Для $j>3$ эта оценка убывает как экспонента от квадратного корня из n. Заметим, что если бы число независимых итераций m было в $2\ln(j-k)(k-1)/\ln(j/2)$ раз больше, то вероятность ошибки убывала бы экспоненциально от n. Возможно, что если пользоваться полной схемой вероятностного декодирования и если продолжать выполнять итерации даже после появления зависимости, то это экспоненциальное поведение будет достигнуто.

Другой путь для исследования схемы вероятностного декодирования состоит в том, чтобы вычислять распределение вероятностей логарифма отношения правдоподобия для большого числа итераций. Такой подход позволяет выяснить, можно ли при заданных j и k с помощью кодирования добиться произвольно малой вероятности ошибки при передаче по данному каналу. Путем расчетов на IBM 709 было найдено, что код с параметрами $j=3$, $k=6$ годится для исправления ошибок, происходящих с вероятностями вплоть до 0,07, и код с параметрами $j=3$, $k=4$ вплоть до 0,144.

Эти цифры интересны еще с той точки зрения, что они опровергают распространившееся мнение, будто пороговое значение скорости передачи, характерное для схемы последовательного декодирования [7], ограничивает интервал скоростей, при которых возможны сколько-нибудь простые способы декодирования.

Экспериментальные результаты

Вероятность неправильного декодирования $P(e)$, связанная с конкретной кодирующей и декодирующей схемой, можно непосредственно измерить, моделируя эту схему вместе с интересующим нас каналом на вычислительной машине. К сожалению, эксперимент должен повторяться до тех пор, пока не накопится достаточно много случаев неправильного декодирования, что необходимо для более или менее точного вычисления $P(e)$; для этого необходимо повторить эксперимент более чем $1/P(e)$ раз. При длине блоков в 500 символов вычислительная машина IBM 7090 затрачивает около 0,1 секунды на выполнение одной итерации в схеме вероятностного

декодирования. Отсюда следует, что даже если $P(e)$ имеет величину порядка 10^{-4}, расчеты потребуют многих часов машинного времени.

Ввиду ограниченности последнего, все эксперименты, которые здесь будут описаны, относились к случаям, когда $P(e)$ велико. Конечно, более интересно было бы знать результаты для малых значений $P(e)$. Однако даже те результаты, которые здесь представлены, вполне убедительно доказывают необходимость дальнейших исследований.

Первые два кода из тех, которые будут рассмотрены, были использованы для передачи по ДСК, а последний код использовался для передачи по гауссовскому каналу.

Повышенный интерес к ДСК был проявлен в ходе исследований по следующим причинам: во-первых, в случае ДСК можно свести к нулю дисперсию числа ошибок, если только работу канала контролировать не вероятностью искажения символа, а фиксированным числом искажений внутри каждого блока; во-вторых, ДСК удобен для сравнения между собой различных кодирующих и декодирующих схем; и наконец, работа схемы декодирования в случае одного канала является, по-видимому, типичной при использовании ее для других каналов.

Код с параметрами (504, 3, 6) для двоичного симметричного канала

С помощью псевдослучайных чисел на вычислительной машине IBM 704 был построен блоковый код длиною 504, в котором каждый символ участвует в трех проверочных соотношениях и все проверочные множества содержат по шести символов. Единственным ограничением при выборе кода было требование, чтобы никакие два проверочных множества не содержали более одного общего символа. Это ограничение, с одной стороны, гарантирует справедливость итераций первого порядка, а с другой стороны, исключает имеющуюся небольшую опасность выбрать код с минимальным расстоянием 2.

На рис. 6 показана относительная частота неудачных попыток декодирования в зависимости от числа

искаженных символов. Цифры в скобках около каждой точки означают число испытаний, проведенных с данным количеством искажений. Ни в одном испытании декодирующее устройство ни разу не выдало неправильного кодового слова; в худшем случае результат декодирования оставался неопределенным. Если при передаче можно

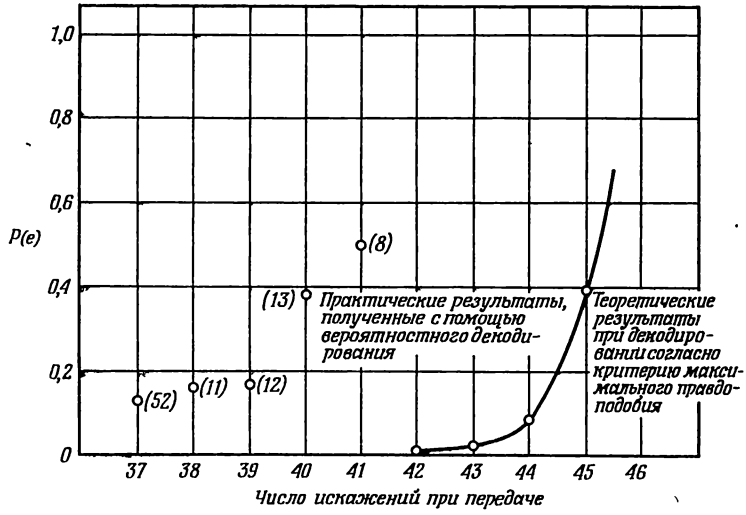


Рис. 6. Зависимость вероятности ошибки от числа искажений при передаче с помощью (504, 3, 6)-кода.

пользоваться каналом обратной связи, то эта неспособность системы исправить отдельные наиболее сложные типы ошибок не вызовет серьезных затруднений, поскольку всегда можно потребовать повторения передачи.

Среди всех шумовых выборок, которые удалось исправить при декодировании, для 86% потребовалось от 9 до 19 итераций. Исправление остальных закончилось между 20-й и 40-й итерациями.

При увеличении числа искажений с 37 до 41 имело место некоторое увеличение числа итераций, необходимых для декодирования, однако не настолько большое, чтобы считать его статистически закономерным.

Кривая на рис. 6 изображает теоретическую оценку вероятности ошибки при декодировании согласно критерию максимума правдоподобия.

В следующем эксперименте (504, 3, 6) код был воспроизведен на вычислительной машине IBM 7090; декодированию были подвергнуты 1000 последовательностей, содержащих по 32 ошибки в каждой. В 26 случаях декодирование не состоялось, а остальные 974 последовательности были декодированы правильно.

Этот результат является весьма обнадеживающим, если учесть, что ни одна из известных схем кодирования и декодирования, рассчитанных на такую скорость передачи [1]), не в состоянии исправлять столь большое количество ошибок, если не прибегать при этом к слишком длительным вычислениям. И все же наиболее интересным следует считать вопрос о том, как работает эта декодирующая схема при малом числе искажений. Скорость, с которой согласно эксперименту убывает вероятность ошибки (при уменьшении числа искажений), слишком мала, однако нет никаких данных для того, чтобы экстраполировать поведение экспериментальной кривой в область с очень малым числом искажений. Исследования в этой области потребуют либо большого количества дополнительных экспериментальных данных, либо поиска новых теоретических подходов.

Код с параметрами (500, 3, 4) для двоичного симметричного канала

Код с параметрами (500, 3, 4), обладающий скоростью передачи 1/4 был воспроизведен на IBM 704 тем же способом, что и (504, 3, 6)-код, описанный в предыдущем разделе. Последовательности, подвергнутые декодированию, содержали от 20 до 77 искаженных символов. Для каждого числа в пределах от 65 до 69 и от 72 до 77 было взято по две последовательности с таким количеством искажений, и по одной последовательности для

[1]) Для кодов, рассматриваемых автором, скорость передачи определяется по формуле $R = 1 - j/k$, поэтому в данном случае $R = 1/2$. — *Прим. перев.*

остальных чисел. Декодирование закончилось успешно для всех последовательностей, за исключением одной с 73 искажениями, одной с 75 и обеих последовательностей с 77 искажениями. Теоретически нарушение корректирующих свойств в ансамбле (500, 3, 4)-кодов начинается со 103 искажений, а в ансамбле всех кодов со скоростью передачи 1/4 — со 108 искажений.

Код с параметрами (500, 3, 5) для канала с белым гауссовским шумом

Рассмотрим канал, на вход которого подается величина, равная плюс единице или минус единице, а величина на выходе образуется прибавлением ко входному значению гауссовской случайной величины со средним значением 0 и дисперсией 1. Логарифм отношения правдоподобия для входной величины относительно выходной равен просто удвоенному значению принятого сигнала. Пропускная способность этого канала подсчитана в работе [5] и равна 0,5 двоичных единиц на символ. Однако, если приемник принимает решение по каждому символу в отдельности и не учитывает в дальнейшем апостериорных вероятностей, то этот канал сводится к ДСК с вероятностью искажения 0,16; при этом пропускная способность уменьшается до 0,37 двоичных единиц на символ.

В этом эксперименте на IBM 704 вместе с описанным каналом был моделирован (500, 3, 5)-код со скоростью передачи 0,4 двоичных единиц на символ.

Для вероятностного декодирования использовалось отношение правдоподобия на выходе канала в логарифмической форме. В 13 попытках декодирующая схема 11 раз декодировала правильно, и два раза декодирование осталось безрезультатным.

Этот эксперимент наводит на мысль, что понижение скорости передачи, неизбежное ввиду неоптимальности схем кодирования и декодирования, предложенных в этой работе, с избытком компенсируется возможностью использовать апостериорные вероятности на выходе канала.

ЛИТЕРАТУРА

1. Shannon C. E., Certain results in coding theory for noisy channels, *Information and Control*, 1, September (1957), 6—25. [Русский перевод: Шеннон К., Работы по теории информации и кибернетике, ИЛ, М., 1963, стр. 497—508.]
2. Slepian D., A class of binary signalling alphabets, *Bell. Sys. Tech. J.*, 35, January, 1956, 203—234. [Русский перевод: Слепян Д., Класс двоичных сигнальных алфавитов, в сб. Теория передачи сообщений, ИЛ, М., 1957, 82—113.]
3. Elias P., Coding for two noisy channels, in Information Theory Cherry C., Ed., 3rd London Symp., September 1955; Butterworths Scientific Publications, London, Eng., 1956.
4. Gallager R. G,, Low density parity check codes, Sc. D. thesis, Mass. Inst. Tech., Cambridge, September 1960.
5. Bloom F. J. et al., Improvement of binary transmission by null-zone reception, *Proc. IRE*, 45, July (1957), 953—975.
6. Fano R. M., The transmission of information, The technology Press, Cambridge, Mass., 1961. [Готовится русский перевод.]
7. Wozencraft J. M., Reiffen B., Sequential decoding, The Technology Press, Cambridge, Mass., 1961. [Русский перевод: Возенкрафт Дж., Рейффен Б., Последовательное декодирование, ИЛ, М., 1963.]

ЭВРИСТИЧЕСКОЕ ОБСУЖДЕНИЕ ВЕРОЯТНОСТНОГО ДЕКОДИРОВАНИЯ [1])

Роберт М. Фано

Цель настоящей статьи — наглядное рассмотрение вероятностного декодирования дискретных сообщений после их передачи по каналу со случайными помехами. Прилагательное „вероятностное" используется для того, чтобы отличить описанный здесь процесс декодирования от алгебраических процессов [1], основанных на специальных структурных свойствах множества кодовых слов, используемых для передачи.

Вначале для того, чтобы обсудить вероятностное декодирование с относящимися к нему работами, опишем основные моменты более общих проблем передачи дискретной информации по каналу со случайными помехами и произведем краткий обзор некоторых ключевых понятий и результатов, имеющих к нему отношение [2]. Впервые эти ключевые понятия и результаты были выдвинуты в 1948 году Шенноном [3]; позднее они разрабатывались и расширялись Шенноном и другими. Первый процесс вероятностного декодирования, имевший практический смысл, был предложен Возенкрафтом в 1957 году [4] и вскоре после этого был обобщен Рэйффеном.

Аппаратура, реализующая этот процесс, была построена в Линкольновской лаборатории [6] и в настоящее время испытывается на телефонной линии.

I. Операция кодирования

Для простоты будем предполагать, что подлежащая передаче информация является последовательностью равновероятных и статистически независимых двоичных цифр. Будем называть эти цифры информационными цифрами, а их скорость R, измеренную числом цифр в секунду, — скоростью передачи информации.

Под каналом связи будет пониматься комплекс имеющихся в распоряжении средств связи. Будем предполагать, что сигналом на входе канала может быть любая функция времени со спектром, лежащим в некоторой

---

[1]) Fano R., A heuristic discussion of probabilistic decoding, *IEEE Transactions on Information Theory*, **IT-9** (1963), № 2, 64—74.

заданной полосе частот, и такая, что ее среднеквадратическое уклонение и (или) пиковое значение удовлетворяют некоторым заданным ограничениям.

Информационные цифры должны быть преобразованы в выбранные сигналы на входе канала и должны быть извлечены из сигнала на выходе канала по возможности с меньшей вероятностью ошибки. Будем называть устройство, которое преобразует информационные цифры в сигналы на входе канала, кодирующим, а устройство, которое извлекает их из сигнала на выходе канала — декодирующим.

Без ограничения общности кодирующее устройство можно рассматривать как автомат с конечным числом состояний, состояние которого в данный момент зависит от ν последних информационных цифр, поступивших на его вход. Это не означает, что состояние автомата однозначно определяется последними ν цифрами. Оно может зависеть также и от времени при условии, что эта зависимость от времени устанавливается заранее и вносится в декодирующее устройство так же как и в кодирующее устройство. Выход кодирующего устройства однозначно определяется текущим состоянием и, следовательно, является функцией ν последних информационных цифр. Как будет видно из дальнейшего, целое число ν, представляющее собой число цифр, от которых зависит выход кодирующего устройства в данный момент, является критическим параметром процесса передачи.

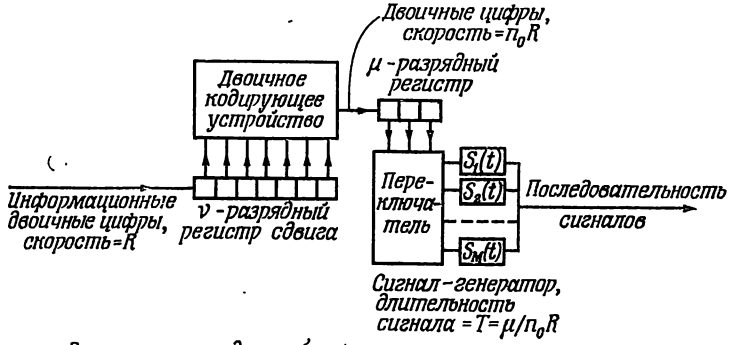
Кодирующее устройство может работать различным образом в зависимости от того, с какой частотой к нему поступают новые цифры. Цифры могут поступать каждый раз по одной в каждые $1/R$ секунд, или по две вместе в каждые $2/R$ секунд и т. д. Предельный случай, в котором информационные цифры поступают на кодирующее устройство блоками, по ν цифр в каждые ν/R секунд, представляет особый интерес и соответствует режиму работы, называемому блочным кодированием. Если каждый последовательный блок из ν цифр поступает на кодирующее устройство в течение времени, которое мало по сравнению с $1/R$, то фактически выход кодирующего устройства зависит только от цифр последнего блока и вовсе не зависит от цифр предшествующих блоков. Таким образом, выход

кодирующего устройства в течение каждого временно́го интервала длительности ν/R, который соответствует передаче одного частного блока цифр, совсем не зависит от выхода в течение временны́х интервалов, соответствующих предшествующим блокам цифр. Другими словами, каждый блок из ν цифр передается независимо от всех предшествующих блоков.

Совсем иное положение возникает в случае, когда информационные цифры поступают на кодирующее устройство блоками длины $\nu_0 < \nu$. При этом выход кодирующего устройства зависит не только от цифр последнего поступившего на кодирующее устройство блока, но также и от $\nu - \nu_0$ цифр предшествующих блоков. Следовательно, он не является независимым от выхода в течение временно́го интервала, соответствующего предшествующим блокам. Как легко увидеть, зависимость на выходе декодирующего устройства от его собственной предыстории фактически простирается безгранично, несмотря на то, что зависимость, вносимая во входные цифры, ограничена последующими ν цифрами. По этой причине режим работы, соответствующий $\nu_0 < \nu$, известен как последовательное кодирование. Различие между блочным кодированием и последовательным кодированием составляет основу для нашего рассмотрения вероятностного декодирования.

Операция кодирования, блочного или последовательного типа наилучшим образом выполняется в два этапа так, как это проиллюстрировано на рис. 1. Первый этап выполняется двоичным кодирующим устройством, которое генерирует n_0 двоичных цифр на одну информационную цифру на входе. Целое число n_0 является параметром устройства и выбирается с учетом остальных операций кодирующего устройства и характеристик канала. Двоичное кодирующее устройство является автоматом с конечным числом состояний, состояние которого зависит от ν последних поступивших на него информационных цифр и, возможно, от времени, так, как это упоминалось ранее. Зависимость состояния от информационных цифр показана на рис. 1 в виде ν информационных цифр, записанных в регистре сдвига с последовательным входом и параллельными выходами. Можно показать, что слож-

ность операций, выполняемых кодирующим устройством с конечным числом состояний, не более чем сложность свертки по модулю два цифр на входе с периодической последовательностью двоичных цифр, обладающей периодом, равным $n_0\nu$. Подходящая периодическая последовательность может быть легко построена, если выбрать равновероятно и независимо $n_0\nu$ цифр из шума. Таким образом, сложность двоичного кодирующего устройства



R – скорость передачи в бит/сек

ν, μ, n_0 – положительные целые числа

$S(t)$ – функция времени длительности T

M – число различных $S(t)$, меньшее и равное 2^{μ}

Рис. 1. Операция кодирования.

растет линейно с ν и его конструкция зависит от канала связи только посредством выбора чисел n_0 и ν.

Второй этап операции кодирования состоит в прямом преобразовании последовательности двоичных цифр, произведенных двоичным кодирующим устройством, в некоторую функцию времени, которая принята для канала. В силу того, что кодирующее устройство представляет собой автомат с конечным числом состояний, результирующая функция времени обязательно должна быть последовательностью элементарных функций времени, выбранных из конечного множества. Элементарные функции времени на рис. 1 обозначены через $S_1(t)$, $S_2(t)$, . . . , $S_M(t)$ где M — число различных элементарных функций времени,

а T — длительность каждой из них. Генерирование этих элементарных функций времени можно выполнить с помощью переключателя, положения которого поочередно устанавливаются цифрами, накопленными регистром μ-разрядного двоичного числа. Цифры, производимые двоичным кодирующим устройством, поступают на этот регистр по μ за один раз, так что каждая последовательная группа из μ цифр преобразуется в один элементарный сигнал. Число различных элементарных сигналов M не может превосходить 2^{μ}, но может быть меньше. Величина M, существенно меньшая, чем 2^{μ}, используется тогда, когда одни элементарные сигналы должны быть употреблены чаще, чем другие. Например, при $\mu = 2$ и $M = 2$ можно сделать частоту одного из двух элементарных сигналов втрое большей частоты другого, если соединить три положения переключателя с одним сигналом и оставшееся одно — с другим.

В то время как представленный на рис. 1 характер преобразования двоичных цифр в сигналы имеет весьма общий вид, существует ряд включенных в эту схему параметров, ограниченных практическими соображениями. Число различных элементарных сигналов M должно быть сравнительно малым; тоже можно сказать и относительно числа n_0. Числа M и n_0, так же как и формы элементарных сигналов, должны быть очень тщательно выбраны с учетом характеристик канала связи. Фактически их выбор заключается в определении класса функций времени, которые можно подавать в канал, и, следовательно, по существу в определении канала [7]. Таким образом, здесь мы сталкиваемся с компромиссом между сложностью аппаратуры и ухудшением характеристик канала.

На рис. 2 изображены два выбора параметров и элементарных сигналов, которые в общем приемлемы, если отсутствует ограничение на ширину полосы частот сигнала и если возмущающий тепловой шум присутствует лишь в канале. В случае а) каждая цифра, произведенная двоичным кодирующим устройством, преобразуется в двоичный импульс, а в случае б) каждый последовательный блок из четырех цифр преобразуется в синусоидальный импульс с длительностью той же, что и эти четыре цифры

и частотой, пропорциональной двоичному числу, образованному группой четырех цифр. Пример, проиллюстрированный на рис. 3, имеет отношение уже к случаю, в котором ширина полосы сигнала ограничена таким

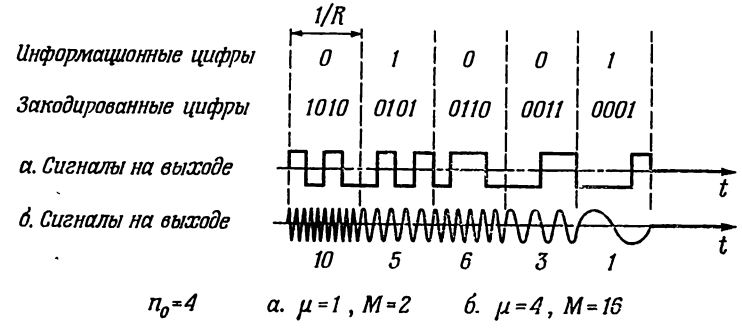


Рис. 2. Примеры кодирования для канала с неограниченной шириной полосы частот.

образом, что допустимая наименьшая длительность импульса равна временно́му интервалу, соответствующему передаче двух информационных цифр. В этом случае

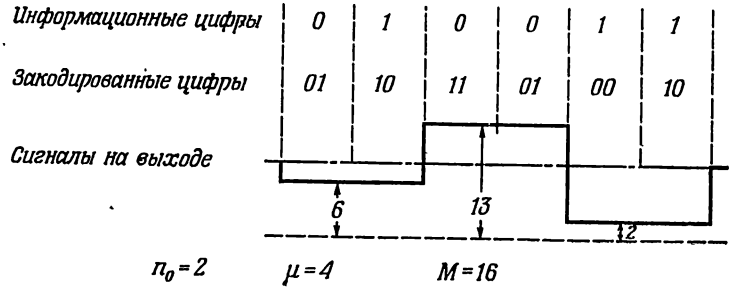


Рис. 3. Примеры кодирования для канала с ограниченной шириной полосы частот.

элементарные сигналы являются импульсами с наименьшей допустимой длительностью и с шестнадцатью различными амплитудами.

На этих примерах становится ясным, что процесс кодирования, проиллюстрированный рис. 1, включает в себя

как частные случаи традиционные виды модуляции, используемые при дискретной передаче. Отличие рассмотренных здесь видов кодирования от традиционных видов модуляции заключается в величине числа ν. В традиционных видах модуляции величина ν очень мала; она часто равна 1 и очень редко превышает 5. Вместо этого здесь вводятся в рассмотрение величины ν, равные 50, и большие. В дальнейшем станет очевидной причина использования таких больших величин ν.

II. Квантование в канале

Предположим, что в операции кодирования заданы длительность и форма элементарных сигналов. Рассмотрим далее, как представить в канале эффект воздействия помех на эти сигналы. Ограничим здесь наше рассмотрение каналами без памяти, так как они представляют собой единственный класс детально исследованных каналов. Канал без памяти применительно к нашим целям можно определить как канал, выход которого в течение каждого интервала длительности T, соответствующего передаче элементарного сигнала, не зависит от того, что было на входе и выходе канала в предыдущие интервалы. Это значит, что действие канала может быть описано внутри любого такого интервала без обращения к предыстории. Будем предполагать также, что канал является стационарным в том смысле, что его свойства не меняются со временем.

Пусть элементарные сигналы передаются с вероятностями $P(S_1)$, $P(S_2)$, ..., $P(S_M)$.

Обозначим через $S'(t)$ сигнал на выходе канала на интервале времени, соответствующем передаче некоторого частного сигнала. Наблюдение сигнала на выходе $S'(t)$ изменяет распределение вероятностей на множестве элементарных сигналов от априорного распределения $P(S)$ к апостериорному условному распределению $P(S/S')$. Последнее распределение, по крайней мере принципиально, может быть найдено по априорному распределению и статистическим характеристикам помех в канале. Точнее, сигнал на выходе $S'(t)$ можно рассматривать как точку в непрерывном пространстве соответствующей размер-

ности. Тогда, если $p(S'/S_k)$—условная плотность вероятности (в предположении, что она существует) сигнала на выходе S' при фиксированном сигнале на входе S_k и

$$p(S') = \sum_{k=1}^{M} P(S_k)\, p(S'/S_k), \tag{1}$$

плотность вероятности S' по всем сигналам на выходе, то

$$P(S/S') = \frac{P(S)\, p(S'/S)}{P(S')}. \tag{2}$$

Для наших целей знание апостериорного распределения вероятности $P(S/S')$ равносильно знанию сигнала на выходе S'. В свою очередь эта вероятность зависит от S' только посредством отношений M плотностей вероятности $p(S/S')$. Далее, эти плотности вероятности практически не могут быть определены с абсолютной точностью. Таким образом, явно или неявно необходимо принять решение о допуске, с которым будут определяться отношения этих плотностей вероятности.

Эффект от введения такого допуска состоит в отождествлении сигналов на выходе S', для которых отношения плотностей вероятностей находятся в пределах установленного допуска. Таким образом, можно разделить S'-пространство на области, в которых отношения плотностей остаются в пределах установленного допуска, и фиксировать только ту частную область, к которой принадлежит сигнал на выходе.

Такое квантование пространства сигнала на выходе S' регулируется соображениями, подобными тем, которыми регулируется выбор элементарных сигналов на входе, а именно, сложность аппаратуры и ухудшение канала.

Этот вопрос больше не будет рассматриваться, хотя в дальнейшем снова будет подчеркиваться, что такое квантование неизбежно на практике. Результатом квантования является то, что следует заменить исходный канал связи новым каналом с дискретными множествами возможных сигналов на входе и выходе и соответствующим образом приведенными возможностями передачи по нему [7].

III. Пропускная способность канала

Здесь удобно изменить общеупотребительную терминологию, используемую при рассмотрении дискретных каналов. Будем называть множество элементарных сигналов на входе входным алфавитом, а каждый отдельный сигнал — входным символом. Подобно этому, множество областей, на которое разделено пространство сигнала на выходе канала, будем называть выходным алфавитом, а каждую отдельную область — выходным символом. Входной и выходной алфавиты будут обозначаться соответственно через X и Y; отдельные принадлежащие им символы будут обозначаться соответственно через x и y. Канал связи, таким образом, полностью описывается алфавитами X и Y и множеством условных распределений вероятностей $P(y|x)$.

Как можно было видеть, при приеме символа y эффект квантования состоит в замене априорного распределения вероятностей $P(x)$ на апостериорное распределение вероятностей

$$P(x|y) = \frac{P(x)\,P(y|x)}{P(y)} = \frac{P(x, y)}{P(y)}, \tag{3}$$

где $P(x, y)$ — совместное распределение вероятностей выходного и входного символов. Таким образом, информация, которая содержится в выходном символе y относительно некоторого входного символа x, определяется как

$$I(x, y) = \log \frac{P(x|y)}{P(x)} = \log \frac{P(y|x)}{P(y)} = \log \frac{P(x, y)}{P(x)\,P(y)}. \tag{4}$$

Как будет видно, эта информационная мера и ее среднее значение, отнесенное к входному и (или) выходному алфавитам, играют центральную роль в рассматриваемой проблеме.

Интересно заметить, что $I(x, y)$ — симметричная функция x и y, т. е. информация, которая содержится в некотором y относительно некоторого x, в точности равна информации, содержащейся в x относительно y. Для того чтобы подчеркнуть это свойство симметрии, $I(x, y) = I(y, x)$ часто называют взаимной информацией, содержащейся

между x и y. В противоположность этому

$$I(x) = \log \frac{1}{P(x)} \tag{5}$$

называют собственной информацией, содержащейся в x. Это название оправдывается тем, что для некоторой частной пары символов $x = x_k$, $y = y_i$ значение $I(x_k, y_i)$ становится равным $I(x_k)$, когда $P(x_k|y_i) = 1$, т. е. когда выходной символ y_i однозначно определяет входной символ x_k. Таким образом, $I(x_k)$ является количеством информации, которое должно быть сообщено относительно x_k для того, чтобы однозначно его установить, и поэтому $I(x_k)$ является верхней гранью для $I(x_k, y)$.

В частном случае алфавита с L равновероятными символами собственная информация каждого символа равна $\log L$. Информация измеряется в битах тогда, когда в приведенных выше выражениях основание логарифма равно 2. Таким образом, собственная информация символов алфавита, имеющего два равновероятных символа, равна 1.

Пусть входной символ выбирается из алфавита X с вероятностью $P(x)$. Среднее, или математическое ожидание взаимной информации, содержащейся между входным и выходным символами, будет

$$I(X, Y) = \sum_{XY} P(x, y)\, I(x, y). \tag{6}$$

Это количество зависит от распределения вероятностей на входе $P(x)$ и от характеристик канала, представленных условными распределениями вероятностей $P(y|x)$. Таким образом, значение $I(X, Y)$ для данного канала зависит лишь от распределения вероятности $P(x)$.

Пропускная способность канала определяется как максимальное значение $I(X, Y)$ по $P(x)$, т. е.

$$C = \max_{P(x)} I(X, Y). \tag{7}$$

Можно показать [2], что если источник, производящий последовательности, состоящие из символов x, соединен со входом канала, то среднее количество информации на

символ, содержащееся на выходе канала относительно его входа, не превосходит C при любых статистических характеристиках источника.

IV. Вероятность ошибки при блочном кодировании

Рассмотрим теперь случай блочного кодирования и предположим, что блок из ν информационных цифр кодирующее устройство превращает в последовательность N элементарных сигналов, т. е. в последовательность N входных символов. Так как, по предположению, информационные цифры равновероятны и независимы друг от друга, количество информации, необходимое для установления каждой из них, равно $\log 2$ (1 *бит*). Таким образом, скорость передачи информации, отнесенная к символу канала, дается равенством

$$R = \frac{\nu}{N} \log 2. \tag{8}$$

(Заметим, что одна и та же буква используется для обозначения скорости передачи информации, отнесенной как к символу канала, так и к единице времени.)

Максимальное количество информации, отнесенное к символу, которое содержится на выходе канала относительно его входа, равно C, пропускной способности канала. Отсюда следует, что нельзя надеяться на возможность передачи информационных цифр с какой-либо разумной степенью точности при любых скоростях $R > C$. Более того, основная теорема Шеннона утверждает, что при любой $R < C$ вероятность ошибочного декодирования блока из ν цифр может быть сделана сколь угодно малой, если использовать достаточно большие значения ν и соответственно большие значения N. Точнее, возможно [2] достичь вероятности ошибки на блок, ограниченной неравенством

$$P_e < 2^{-\nu(\alpha/R)+1}, \tag{9}$$

где α не зависит от ν и изменяется вместе с R так, как это показано на рис. 4. Таким образом, для любого $R < C$ вероятность ошибки падает экспоненциально с увеличением ν.

Из (9) ясно, что вероятность ошибки обусловливается в первую очередь произведением ν на α/R; при этом последний член является для данного канала функцией только R. Таким образом, та же самая вероятность ошибки может быть получена при малых значениях ν и относительно малых значениях R, либо же при значениях R, близких к C, и соответственно больших значениях ν. В первом случае, который соответствует традиционным видам модуляции, в силу малости значения ν кодирующее и декодирующее устройства относительно

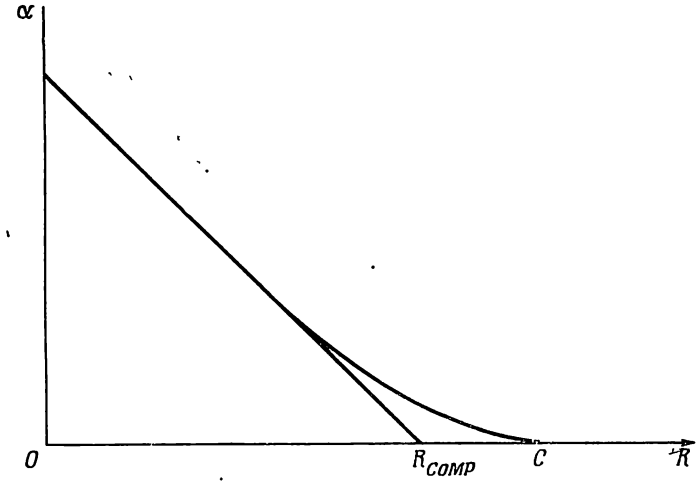


Рис. 4. Связь коэффициента в показателе экспоненты α со скоростью передачи информации R в оценке (9).

просты, но канал используется неэффективно. В противоположность этому во втором случае канал используется эффективно, однако сравнительно большое значение ν означает, что устройства на концах линии связи должны быть существенно более сложными. Таким образом, мы сталкиваемся с компромиссом между эффективностью использования канала и сложностью устройств, расположенных на концах линии связи.

Как было указано в разд. I, работа двоичного кодирующего устройства относительно несложна; она состоит

в применении операции свертки к информационным цифрам на входе и периодической последовательности двоичных цифр, период которой равен $n_0 \nu$. Таким образом, грубо говоря, сложность кодирующего устройства растет линейно с ν. В то же время, операция декодирования является существенно более сложной как в смысле ее схемы, так и в смысле аппаратуры, требуемой для ее воплощения. Ей будет посвящена оставшаяся часть этой работы.

V. Вероятностное блочное декодирование

Как было показано, в процессе блочного кодирования кодирующее устройство превращает каждую последовательность ν информационных цифр в некоторую последовательность N входных символов канала. Будем называть любую такую последовательность входных символов кодовым словом и будем обозначать через u_k кодовое слово, соответствующее последовательности информационных цифр, которое в двоичной системе исчисления означает число k. Последовательность N выходных символов, возникающих из входного кодового слова, будет обозначаться через v.

Вероятность того, что какое-либо слово u произведет некоторую выходную последовательность v, дается равенством

$$P(v/u) = \prod_{j=1}^{N} [P(y/x)]_j, \tag{10}$$

в котором подстрочный индекс j означает, что значение условной вероятности вычислено для входного и выходного символов, появляющихся в j-й позиции слов u и v. Кроме того, так как все последовательности информационных цифр передаются с равными вероятностями, апостериорная вероятность любого частного кодового слова u после приема некоторой выходной последовательности v будет

$$P(u/v) = \frac{P(v/u)\,P(u)}{\sum_U P(v/u)\,P(u)} = 2^{-\nu}\frac{P(v/u)}{P(v)}. \tag{11}$$

Таким образом, кодовое слово, имеющее наибольшую апостериорную вероятность для некоторого выходного v, является тем словом, которое максимизирует условную вероятность $P(v|u)$, представленную равенством (10). Можно заключить, что для того, чтобы минимизировать вероятность ошибки, декодирующее устройство должно выбирать кодовое слово, которое производит последовательности v на выходе канала с наибольшей вероятностью $P(v|u)$.

В то время как установление процесса оптимального декодирования является столь простым, его воплощение представляет весьма серьезные трудности при любых значительных ν. В действительности не существует общего процесса определения кодового слова, соответствующего наибольшему значению $P(v|u)$, такого, который бы избегал вычисления этих вероятностей для большинства из 2^{ν} возможных кодовых слов. Ясно, что необходимое количество вычислений растет экспоненциально вместе с ростом ν и очень быстро становится недопустимо большим. Однако, если не настаивать на минимизации вероятности ошибки, то можно извлечь выгоду из того факта, что при очень малой вероятности ошибки кодовые слова, имеющие наибольшую апостериорную вероятность, должны быть почти всегда существенно более вероятными, чем все другие кодовые слова. Таким образом, может оказаться достаточным найти кодовое слово с вероятностью $P(v|u)$ большей, чем некоторое установленное пороговое значение, допустив возможность того, что существуют другие кодовые слова, вероятности $P(v|u)$ для которых превышают пороговое значение, а также то, что $P(v|u)$ для правильного кодового слова может оказаться меньше порогового значения.

Пусть выбрано некоторое пороговое значение. Предположим, что для данной принятой последовательности v существует кодовое слово u_k, для которого

$$P(u_k|v) \geqslant \sum_{i \neq k} P(u_i|v), \tag{12}$$

где суммирование распространяется по всем остальным $2^{\nu} - 1$ кодовым словам. Тогда кодовое слово u_k должно иметь наибольшую апостериорную вероятность. Условие,

выраженное формулой (12), может быть с помощью (11) представлено в виде

$$P(v/u_k) \geqslant \sum_{i \neq k} P(v/u_i). \tag{13}$$

Значение $P(v/u_k)$ можно легко вычислить с помощью (10). Однако остается нерешенной проблема вычисления тех же условных вероятностей для всех других кодовых слов. Эту трудность можно обойти, если использовать аппроксимацию, связанную с процессом случайного кодирования, с помощью которого была установлена оценка (9).

В процессе случайного кодирования каждое кодовое слово составляется при помощи случайного независимого выбора его символов в соответствии с установленным распределением вероятности $P_0(x)$. Правая часть неравенства (9) в действительности представляет собой среднее значение вероятности ошибки, взятое по ансамблю множества кодов, которые образованы из слов, составленных таким образом. Это означает, между прочим, что удовлетворительные кодовые слова практически можно получить, следуя такому процессу случайного выбора.

Пусть рассматриваемые кодовые слова составлены с помощью случайного и независимого выбора символов в соответствии с некоторым принятым распределением вероятности $P_0(x)$. Тогда представляется правильным подставить вместо правой части неравенства (13) ее среднее значение, взятое по ансамблю множества кодов, которые образованы из слов, составленных таким случайным образом. В таком ансамбле множества кодов вероятность $P_0(u)$ того, что любая частная входная последовательность u выбрана в качестве кодового слова, равна

$$P_0(u) = \prod_{j=1}^{N} [P_0(x)]_j, \tag{14}$$

где подстрочный индекс j означает, что $P_0(x)$ вычислено для j-го символа последовательности u. Таким образом, среднее значение правой части равенства (13), ввиду (10),

будет

$$(2^{\nu}-1)\sum_{U} P_0(u)\,P(v/u) = (2^{\nu}-1)\prod_{j=1}^{N}[P_0(y)]_j, \qquad (15)$$

где U — множество всех возможных входных последовательностей, а

$$P_0(y) = \sum_{X} P_0(x)\,P(y/x) \qquad (16)$$

означает распределение вероятности выходных символов в случае, когда входные символы передаются независимо друг от друга с вероятностью $P_0(x)$. Заменяя затем правую часть (13) правой частью (15) и представляя $P(v/u_k)$, так, как это сделано в (10), получаем

$$\prod_{j=1}^{N}\left[\frac{P(y/x)}{P_0(y)}\right]_j \geqslant 2^{\nu}-1. \qquad (17)$$

Аппроксимируя, наконец, $2^{\nu}-1$ с помощью 2^{ν} и производя логарифмирование обеих частей (17), получаем

$$\sum_{j=1}^{N}\left[\log\frac{P(y/x)}{P_0(y)}\right]_j \geqslant NR, \qquad (18)$$

где R — определенная согласно (8) скорость передачи, отнесенная к символу канала.

Пороговому условию, выраженному неравенством (18), можно дать очень интересную интерпретацию. Слагаемое с индексом j в сумме является взаимной информацией, содержащейся между j-м выходным символом и j-м входным символом; при этом предполагается, что входные символы появляются с вероятностью $P_0(x)$. Если входные символы статистически независимы друг от друга, то сумма этих взаимных информаций будет равна взаимной информации между выходной последовательностью и входной последовательностью. Таким образом, неравенство (18) утверждает, что выход канала может быть удовлетворительно декодирован в некоторое кодовое слово, если взаимная информация на выходе относительно этого кодового слова, вычисленная в предположении, что N входных символов были выбраны независимо друг

от друга с вероятностью $P_0(x)$, превышает количество информации, переданное на одно кодовое слово.

Оказывается, что пороговое значение правой части (18) не только является удовлетворительным, как это показывает наше наглядное обоснование, но является таким значением, которое минимизирует среднюю вероятность ошибки для порогового декодирования, взятую по ансамблю множества кодов. Это было показано Шенноном в его неопубликованной работе. Оценка для вероятности ошибки, полученная Шенноном, имеет тот же вид, что и (9), однако значение α несколько меньше, чем то, которое получено для оптимального декодирования. Шеннон в своей работе предполагает, что ошибка возникает всякий раз, когда неравенство (18) либо удовлетворяется для любого кодового слова, отличного от правильного, либо не удовлетворяется для правильного кодового слова.

Хотя вероятность ошибки при пороговом декодировании больше, чем при оптимальном декодировании, для нее остается справедливой оценка, аналогичная оценке (9). Это обстоятельство наталкивает на мысль попытаться найти процесс, который быстро отвергал бы любое кодовое слово, для которого неравенство (17) не удовлетворяется, и, таким образом, относительно быстро сходился бы для действительно переданного кодового слова. Однако, даже если отвергнуть неправильное кодовое слово после его оценки по (17), выполненной для некоторой малой, но конечной доли всех N символов, то количество вычислений все еще будет возрастать экспоненциально вместе с ростом ν. Для того чтобы избежать экспоненциального роста, нужно позаботиться о том, чтобы иметь возможность устранять большие подмножества кодовых слов с помощью вычисления левой части неравенства (17) для некоторой доли символов одного-единственного кодового слова. Это значит, что кодовые слова должны обладать свойствами древовидной структуры, которая является следствием рассматриваемого в следующей части последовательного кодирования.

Именно эти обстоятельства привели в 1957 г. Возенкрафта к выработке процедуры последовательного декодирования. С тех пор были развиты и другие процедуры декодирования (алгебраическая [1] и вероятностная [8]),

которые имеют практическое значение в некоторых специальных случаях. Однако последовательное декодирование остается единственной известной процедурой, применимой ко всем каналам без памяти.

Существуют, кроме того, основания [9] верить, что некоторая модифицированная форма последовательного декодирования может привести к удовлетворительным результатам в применении к более широкому классу каналов.

VI. Последовательное декодирование

Оставшиеся части этой работы посвящены эвристическому рассмотрению некоторой процедуры последовательного декодирования, которая недавно была предложена автором. Эта процедура во многих отношениях подобна процедуре Возенкрафта [4—6], она более проста для восприятия и, следовательно, ее обоснование и оценка могут быть выполнены более просто.

В лаборатории Линкольна Массачузетского технологического института (Лексингтон, Массачузетс) производится экспериментальное сравнение обеих процедур. Детальное исследование новой процедуры будет произведено в работе, которая скоро будет опубликована.

Рассмотрим более подробно структуру выхода кодирующего устройства в случае последовательного кодирования, т. е. тогда, когда информационные цифры поступают на кодирующее устройство блоками длины ν_0 (на практике ν_0 редко превышает 3 или 4). Выход кодирующего устройства на интервале времени, соответствующем некоторому блоку, выбирается с помощью цифр блока из некоторого множества 2^{ν_0} различных последовательностей входных символов канала. Это частное множество последовательностей, из которого выбирается выход кодирующего устройства, определяется в свою очередь $\nu - \nu_0$ информационными цифрами, предшествующими рассматриваемому блоку. Таким образом, множество возможных последовательностей на выходе кодирующего устройства может быть представлено с помощью дерева с 2^{ν_0} ветвями, исходящими из каждого узла. Каждый последовательный блок из ν_0 информационных

цифр вызывает переход кодирующего устройства от одного узла к другому по ветви, которая точно определяется цифрами блока.

Два дерева, изображенные на рис. 5, соответствуют двум примерам, проиллюстрированным на рис. 2, *б* и 3. Первый пример дает двоичное дерево ($\nu_0 = 1$), в то время как второй пример — четверичное дерево ($\nu_0 = 4$).

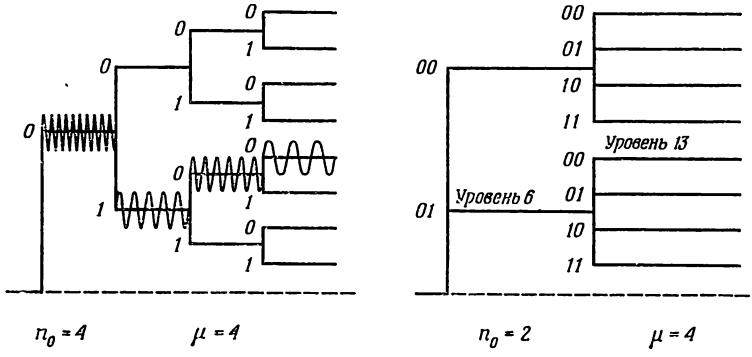


Рис. 5. Кодовые деревья, соответствующие примерам, изображенным на рис. 2, *б* и 3.

Итак, операция кодирования может быть описана с помощью дерева, в каждом узле которого ветвь для дальнейшего следования выбирается с помощью информационных цифр. Путь по дереву, возникающий в результате таких последовательных выборов, определяет выход кодирующего устройства. То же самое можно выразить так: каждый поступающий на кодирующее устройство блок из ν_0 цифр представляется для передачи последовательностью символов, выбранной из некоторого множества 2^{ν_0} различных последовательностей, которое зависит от $\nu - \nu_0$ предыдущих информационных цифр. Таким образом, на выходе канала в течение интервала времени, соответствующего некоторому блоку из ν_0 информационных цифр, возникает информация не только относительно этих цифр, но также и относительно $\nu - \nu_0$ предыдущих цифр.

Операцию декодирования можно рассматривать как процесс, который на основании того, что имеется на вы-

ходе канала, определяет путь на дереве, по которому следовало кодирующее устройство. Предположим сначала, что декодирующее устройство на основе того, что появляется на выходе канала в течение интервала времени, соответствующего передаче некоторой ветви, выбирает в каждом узле тот путь, который имеет наибольшую апостериорную вероятность. Если помехи в канале таковы, что действительно переданная ветвь не оказалась наиболее вероятной ветвью, то декодирующее устройство сделает ошибку и в результате достигнет узла, не лежащего на пути, по которому следует кодирующее устройство. Таким образом, ни одна из ветвей, исходящих из достигнутого узла, не будет наиболее вероятной ветвью на входе канала. Если же какая-либо ветвь случайно оказывается наиболее вероятной входной ветвью, то в узле, в котором она кончается, возникает та же самая ситуация по отношению к ветвям, исходящим из него и т. д. Это грубое описание может быть уточнено.

Предположим, что ветви дерева составляются, так же как и в случае блочного кодирования, с помощью случайного и независимого выбора символов в соответствии с некоторым распределением вероятности $P_0(x)$. Практически это можно выполнить, если $n_0\nu$ двоичных цифр, образующих периодическую последовательность, с которой свертывается последовательность информационных цифр, выбрать случайно и с равными вероятностями и если соответствующим образом произвести соединения положений переключателя, представленного на рис. 1, с элементарными сигналами. При этом так же, как и в случае порогового блочного декодирования, декодирующее устройство, следуя по пути на дереве, вычисляет значение

$$I_N = \sum_{j=1}^{N\nu} \left[\log \frac{P(y|x)}{P_0(y)}\right]_j, \tag{19}$$

где y в j-м слагаемом является j-м символом на выходе канала, а x в том же слагаемом — j-м символом на пути, по которому следует декодирующее устройство.

Тогда если путь, по которому следует декодирующее устройство, совпадает с тем, по которому движется

кодирующее устройство, то можно ожидать, что I_N остается большим NR (где R — скорость передачи информации, отнесенная к одному символу, — здесь она все еще остается равной отношению числа информационных цифр к числу соответствующих символов канала, но уже не представляется формулой (8)). Однако как только декодирующее устройство сделает ошибку и прибудет поэтому в узел, который не лежит на пути следования кодирующего устройства, значение I_N, которое соответствует ветвям, находящимся за этим узлом, с большой вероятностью станет меньшим NR. Таким образом, в конце концов значение I_N становится меньшим NR и это является показателем того, что произошла ошибка в каком-то предыдущем узле.

В такой ситуации для того, чтобы вновь вернуться на правильный путь, декодирующее устройство, очевидно, должно попытаться найти то место, где произошла ошибка. Желательно поэтому для каждого узла вычислить связанную с ним вероятность того, что в этом месте произойдет ошибка.

VII. Вероятность ошибки для некоторого пути

Обозначим через N порядковый номер символа, предшествующего некоторому частному узлу и через N_0 — порядковый номер последнего выходного символа. Так как a priori все пути на дереве являются равновероятными, их апостериорные вероятности пропорциональны условным вероятностям $P(v|u)$, в которых u — последовательность символов, соответствующая некоторому частному пути, а v — результирующая последовательность выходных символов. Эта условная вероятность может быть представлена в виде

$$P(v|u) = \prod_{j=1}^{N} [P(y|x)]_j \prod_{j=N+1}^{N_0} [P(y|x)]_j. \qquad (20)$$

Первый сомножитель в правой части равенства (20) имеет одно и то же значение для всех путей, которые совпадают в первых N символах. Число таких путей, кото-

рые отличаются в некоторых оставшихся $N_0 - N$ символах, равно

$$m = 2^{(N_0 - N) R / \log 2}. \tag{21}$$

Так же, как и в случае блочного декодирования, практически нереально вычислить второй сомножитель в правой части равенства (20) для каждого из этих путей. Эта трудность вновь будет обойдена с помощью усреднения по ансамблю случайно составленных деревьев. По аналогии со случаем порогового блочного декодирования имеем

$$P_0(v/u) = \prod_{j=1}^{N} [P(y/x)]_j \prod_{j=N+1}^{N_0} [P_0(y)]_j, \tag{22}$$

где $P_0(y)$ дается формулой (16).

Пусть P_N — вероятность того, что путь, по которому следует кодирующее устройство, является одним из $m-1$ путей, совпадающих с тем, по которому следует декодирующее устройство на протяжении первых N символов, и отличающихся от него в дальнейшем. Аппроксимируя $m-1$ с помощью m, получаем

$$P_N = K_1 2^{(N_0 - N) R / \log 2} \prod_{j=1}^{N} [P(y/x)]_j \prod_{j=N+1}^{N_0} [P_0(y)]_j =$$

$$= K_2 2^{-NR/\log 2} \prod_{j=1}^{N} \left[\frac{P(y/x)}{P_0(y)}\right]_j, \tag{23}$$

где K_1 и K_2 — постоянные множители. Логарифмируя обе части (23), окончательно получаем

$$\log P_N = \log K_2 + \sum_{j=1}^{N} \left[\log \frac{P(y/x)}{P_0(x)} - R\right]_j. \tag{24}$$

Смысл формулы (24) лучше обсудить, если выразить входящие в нее члены через порядковые номера узлов, лежащих на пути, по которому следует декодирующее устройство. Обозначим через N_b число символов канала на одну ветвь (предполагая, для простоты, что оно одно и то же для всех ветвей), а через n — порядковый номер узла, следующего за N-м символом. Тогда формула (24)

может быть представлена в виде

$$\log P_n = \log K_2 + \sum_{k=1}^{n} \lambda_k, \tag{25}$$

где

$$\lambda_k = \sum_{j=(k-1)N_b+1}^{kN_b} \left[\log \frac{P(y|x)}{P_0(y)} - R\right]_j \tag{26}$$

является той частью суммы, входящей в (24), которая зависит от k-й рассмотренной декодирующим устройством

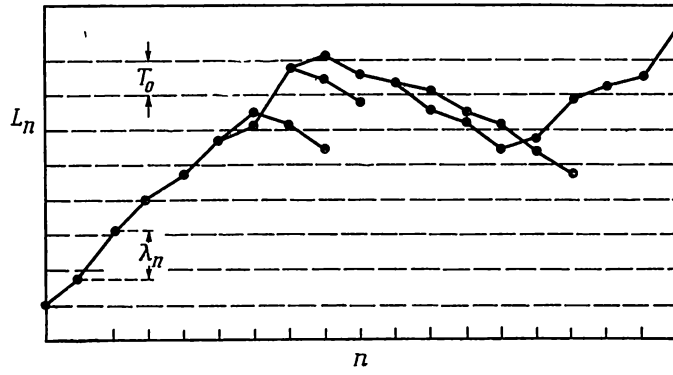


Рис. 6. Поведение функции правдоподобия L_n на различных путях по дереву. Необрывающаяся кривая соответствует правильному пути.

ветви. В итоге можно опустить постоянную в (25) и сосредоточить внимание на сумме

$$L_n = \sum_{k=1}^{n} \lambda_k,$$

которая монотонно возрастает вместе с ростом вероятности P_n.

Типичное поведение L_n как функции от n изображено на рис. 6. В обычных случаях, в которых вероятность того, что ошибка произошла в некотором узле, больше вероятности того, что ошибка произошла в предшествующем узле, значение λ_k положительно. Предположим, что декодирующее устройство достигло n-го узла и значение λ_{n+1}, соответствующее апостериори наиболее вероятной

ветви, исходящей из этого узла, положительно. В этом случае декодирующее устройство должно приступить к рассмотрению ветвей, исходящих из следующего узла, в предположении, что путь до него является правильным. В противном случае, когда значение λ_{n+1} отрицательно, декодирующее устройство должно предположить, что произошла ошибка, и рассмотреть другие ветви, исходящие из предыдущих узлов, в порядке, соответствующем их вероятностям.

VIII. Особенности процедуры декодирования

Оказывается, что процесс поиска других ветвей может быть значительно упрощен, если при поиске не придерживаться точного порядка, предписанного вероятностью. Ниже описывается процедура, в которой декодирующее устройство движется от узла к узлу в прямом или обратном направлении в зависимости от того, является ли в рассматриваемом узле значение L большим или меньшим некоторого порогового значения T. Значение T увеличивается или уменьшается скачком некоторой установленной величины T_0, согласно следующему правилу. Предположим, что декодирующее устройство находится в узле с порядковым номером n и что оно намерено двинуться вперед, выбрав наиболее вероятную ветвь среди еще не проверенных ветвей. Если результирующее значение L_{n+1} превышает пороговое значение T, то ветвь принимается, а T устанавливается равным наибольшему возможному числу, не превосходящему L_{n+1}. Если, наоборот, L_{n+1} оказывается меньшим T, то декодирующее устройство отвергает ветвь и движется назад к узлу с порядковым номером $n-1$. Если $L_{n-1} \geqslant T$, то декодирующее устройство предпринимает попытку вновь двинуться вперед с помощью выбора наиболее вероятной ветви среди тех, которые еще остались не проверенными, или, если все ветви, исходящие из этого узла, были уже проверены, оно движется назад к узлу с порядковым номером $n-2$. Подобным образом декодирующее устройство движется вперед и назад до тех пор, пока вновь не придет к узлу, для которого значение L меньше, чем текущее пороговое значение T.

Пусть декодирующее устройство возвратилось назад к узлу, для которого L меньше, чем текущее пороговое значение. Все пути, исходящие из этого узла, должны содержать по крайней мере один узел, для которого L спускается ниже порога. Эта ситуация может возникнуть из-за ошибки в этом узле или в каком-то предыдущем узле, что проиллюстрировано на рис. 6 с помощью первой кривой, ответвляющейся выше правильной кривой. Это также может явиться результатом того, что из-за необычного усиления помех в канале значение L на правильном пути достигнет максимума и затем в промежутке до последующего возрастания будет иметь минимум (это проиллюстрировано основной кривой рис. 6). В каждом из этих случаев порог должен быть снижен на T_0 для того, чтобы разрешить декодирующему устройству действовать.

После того как порог будет снижен, декодирующее устройство предпринимает попытку вновь двинуться вперед с помощью выбора наиболее вероятной ветви в точности так же, как если бы оно никогда не заходило за узел, в котором порог был снижен. Это вынуждает декодирующее устройство вновь проследить все ранее рассмотренные пути с тем, чтобы установить, остается ли L выше нового порога на одном из этих путей. Значение T, конечно, не увеличивается, когда декодирующее устройство вновь прослеживает какой-либо из этих путей, до тех пор, пока он не достигнет предварительно неизученной ветви. В противном случае, декодирующее устройство снова и снова будет прослеживать тот же самый путь.

Если на правильном пути L остается выше нового порога, то декодирующее устройство может продолжать работу за местом, в которое оно до этого было возвращено, а порог может быть увеличен вновь, как это обсуждалось ранее. Если же вместо этого в некотором узле правильного пути L опустилось ниже уже сниженного порога или если в некотором предшествующем узле, для которого L меньше, чем сниженный порог, возникла ошибка, то порог должен быть снова снижен на T_0. Этот процесс продолжается до тех пор, пока значение порога сделается меньшим, чем наименьшее значение L на правильном

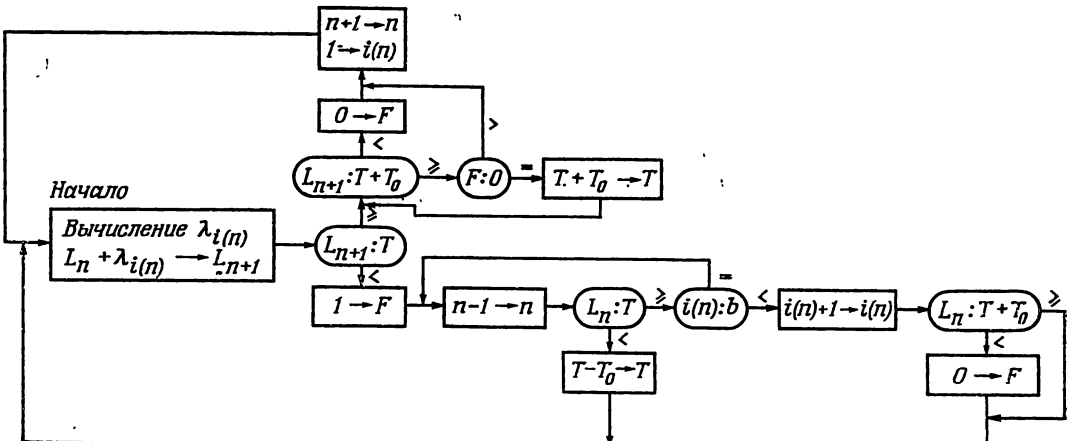
пути, или сделается меньшим, чем значение L в узле, в котором имела место ошибка.

Схема операций, изображенная на рис. 7, описывает процедуру точнее, чем это было сделано на словах. Предположим, что декодирующее устройство находится в некотором узле с порядковым номером n. Крайний левый блок на схеме рассматривает ветви, исходящие из этого узла, и выбирает одну, которая занимает i-е место в ряду уменьшающихся апостериорных вероятностей. (Значение λ для этой ветви обозначено на рис. 7 подстрочным индексом $i(n)$.) Предполагается, что номер $i(n)$ будет запоминаться при каждом n для использования его в будущем. Число ветвей равно $b = 2^{\nu_0}$. Таким образом, $1 \leqslant i(n) \leqslant b$.) Далее суммируются L_n и $\lambda_{i(n)}$ и тем самым вычисляется значение L_{n+1}.

Значение L_n может понадобиться в дальнейшем, если декодирующее устройство вынуждено будет возвратиться назад к n-ому узлу, и поэтому оно либо запоминается, либо снова вычисляется в случае необходимости. Для простоты в схеме принято, что L_n запоминается для каждого значения n.

После этого блока пояснения относительно работы схемы приводятся в ней самой. Исключение составляет блок, выполняющий функцию двоичной переменной F. Эта переменная является обычным контрольным вентилем, который разрешает или запрещает порогу увеличиться в зависимости от того, имеет место $F = 0$ или $F = 1$. Таким образом, F должно быть положено равным 0, когда декодирующее устройство выбирает ветвь в первый раз, и равным 1, когда та же ветвь будет прослеживаться после снижения порога. Значение F устанавливается равным 1 каждый раз, когда ветвь отвергается; до выбора новой ветви оно вновь устанавливается равным 0, если только $T \leqslant L_n < T + T_0$ для узла, в который декодирующее устройство было вынуждено возвратиться. Значение F вновь устанавливается равным 0, если после того, как ветвь принимается для узла, которым она заканчивается, $T \leqslant L_{n+1} < T + T_0$.

Можно проверить, что значение F остается равным 1 тогда, когда путь вновь прослеживается после снижения порога, и заменяется на 0 в узле, в котором L спускается ниже ранее установленного порога.



Р и с. 7. Схема операций для процедуры последовательного декодирования.

$1 \longrightarrow F$ означает положить F равным 1;

$L_n+\lambda_{i(n)} \longrightarrow L_{n+1}$ означает положить L_{n+1} равным $L_n+\lambda_{i(n)}$;

$n+1 \longrightarrow n$ означает подставить $n+1$ вместо n (увеличить n на единицу);

$i(n)+1 \longrightarrow i(n)$ означает подставить $i(n)+1$ вместо $i(n)$ (увеличить $i(n)$ на единицу);

$T+T_0 \longrightarrow T$ означает подставить $T+T_0$ вместо T (увеличить T на T_0);

$L_{n+1}:T$ означает сравнить L_{n+1} и T, если $L_{n+1} \geqslant T$, то направление движения то, которое обозначено $\geqslant$.

IX. Оценка процедуры

Действие описанной в предыдущем разделе процедуры последовательного декодирования было аналитически исследовано для произвольного дискретного канала без памяти. Подробные вычисления и результаты даны в работе, которая скоро будет опубликована. Ниже будут обсуждены общие стороны этих результатов и их смысл. Характеристики, которые при последовательном декодировании имеют наибольшую важность,— это сложность процесса, результирующая вероятность ошибки, отнесенная к одной цифре, и вероятность отказа при декодировании. Определим и опишем эти характеристики по порядку.

Понятие сложности в действительности состоит из двух связанных, но различных понятий. Это — количество аппаратуры, требуемой, чтобы выполнить операцию декодирования, и скорость, с которой аппаратура должна работать. Исследование схемы операций, изображенной на рис. 7, показывает, что необходимая аппаратура включает как основную часть аппаратуру, предназначенную для генерирования возможных сигналов на входе канала (точных копий сигналов, идущих с выхода кодирующего устройства), и аппаратуру, предназначенную для запоминания сигналов на выходе канала и декодированных информационных цифр. Все другие данные, которые требуются для операции декодирования, могут быть либо вычислены по сигналу на выходе канала и декодированным информационным цифрам, либо, если это окажется практически более удобным, их можно запомнить в добавление к данным, которые уже хранятся в памяти. В разд. I было найдено, что сложность кодирующей аппаратуры увеличивается линейно вместе с ростом объема памяти кодирующего устройства ν, так как двоичное кодирующее устройство должно свертывать две двоичные последовательности, длины которых пропорциональны ν. Требования к памяти будут обсуждаться вместе со случаями отказа при декодировании.

Скорость, с которой должна работать декодирующая аппаратура, не является одной и той же для всех ее частей. Оказывается, однако, разумным измерять требуемые скорости средним числом $\bar{n}$ ветвей, которые декодирующее

устройство должно просмотреть, отнесенным к одной переданной ветви. Для $\bar{n}$ может быть получена неплохая оценка сверху, которая обладает следующими свойствами. Для любого данного дискретного канала без памяти существует максимальная скорость передачи информации, для которой оценка $\bar{n}$ остается конечной при неограниченном возрастании длины сообщений. Эта максимальная скорость выражается формулой

$$R_{\text{выч}} = \max_{P_0(x)} \{-\log \sum_Y [\sum_X P_0(x) \sqrt{P(y|x)}]^2\}. \qquad (28)$$

При этом для любой скорости передачи $R < R_{\text{выч}}$ оценка для $\bar{n}$ не только конечна, но и не зависит от ν. Это значит, что средняя скорость, с которой работает декодирующая аппаратура, не зависит от ν.

Максимальная скорость, представленная формулой (28), устанавливает интересную связь между множителем α в показателе экспоненты оценки (9) и вероятностью ошибки при оптимальном блочном декодировании. Как показано на рис. 4, кривая α от R для малых значений R совпадает с прямой линией, имеющей наклон -1. Эта прямая линия пересекает ось R в точке $R = R_{\text{выч}}$. Ясно, что $R_{\text{выч}} < C$. Автор не знает ни одного канала, для которого $R_{\text{выч}}$ меньше, чем $\frac{1}{2}C$, однако точная нижняя грань для $R_{\text{выч}}$ еще не найдена.

Обратим далее внимание на два случая, в которых декодирующее устройство отказывается воспроизвести переданные информационные цифры. В описанном выше процессе декодирования не установлен предел того, как далеко декодирующее устройство может возвращаться назад для того, чтобы исправлять ошибки. На практике, однако, такой предел устанавливается доступной емкостью памяти. Таким образом, отказы при декодировании возникают всегда, когда декодирующее устройство уходит так далеко по неверному пути, что за время его возвращения к узлу, в котором произошла ошибка, необходимая информация уже исчезает из памяти. Любой такой отказ немедленно распознается декодирующим устройством, так как оно не способно выполнить следующую операцию согласно установленной процедуре.

То, каким образом поступают на практике с такими отказами, зависит от того, имеется ли канал обратной связи. Если имеется канал обратной связи, то декодирующее устройство может автоматически запросить повторение передачи [10]. Если канал обратной связи отсутствует, то поток информационных цифр может быть разбит на отрезки подходящей длины и заданная последовательность, состоящая из $\nu - \nu_0$ цифр, должна быть помещена между отдельными такими отрезками. В этом случае, если отказы при декодировании возникнут в течение некоторого отрезка, то остаток этого отрезка теряется, но декодирующее устройство начнет работать снова в начале следующего отрезка.

Другой тип отказов при декодировании состоит в ошибочном декодировании цифр, которые не могут быть исправлены, независимо от объема памяти, которым располагает декодирующее устройство. Эти ошибки по своей природе не обнаруживаются декодирующим устройством и поэтому не останавливают операцию декодирования. Они возникают следующим образом.

Для того чтобы породить ветви, которые должны быть рассмотрены, на декодирующее устройство должны поступать информационные цифры, переведенные в точные копии сигналов, идущих с выхода кодирующего устройства. Как было рассмотрено в разд. VI, множество ветвей, исходящих из некоторого узла, задается последними $\nu - \nu_0$ информационными цифрами. Предположим теперь, что декодирующее устройство движется вперед по неверному пути и что оно воспроизводит после нескольких неправильных цифр последовательность $\nu - \nu_0$ информационных цифр, которая случайно совпадает с переданной. Это весьма невероятное событие, так как декодирующее устройство обычно вынуждено возвращаться назад задолго до того, как оно может воспроизвести эти многочисленные цифры. Однако в действительности это может случиться, если помехи в канале достаточно сильны во время соответствующего интервала. После такого события точные копии сигналов, идущих с выхода кодирующего устройства (которое производит ветви, необходимые для рассмотрения), становятся полностью свободными от неправильных цифр и, следовательно, операция декодирования

совершается в точности так же, как если бы декодирующее устройство следовало все время по правильному пути. Таким образом происходящие ошибки не могут быть исправлены. В самом деле, если бы декодирующее устройство возвратилось к узлу, в котором была совершена первая ошибка, то оно в конце концов проследовало бы снова по тому же самому неверному пути.

Для результирующей вероятности ошибки, отнесенной к одной декодированной цифре, справедлива оценка, подобная (9). Однако множитель α в показателе экспоненты здесь будет больше, чем при блочном кодировании, хотя он, конечно, обращается в нуль при $R = C$. Этот факт можно наглядно объяснить, если заметить, что зависимость выхода кодирующего устройства от его прошлого простирается дальше символов, соответствующих последним ν информационным цифрам. Таким образом мы можем сказать, что для одного и того же значения ν эффективная ограничивающая длина больше для последовательного кодирования, чем для блочного кодирования.

И, наконец, рассмотрим отказы при декодировании, которые упоминались ранее. Так как эти отказы при декодировании являются результатом недостаточной емкости памяти, необходимо точнее установить тип используемого запоминающего устройства. Предположим, что запоминающее устройство способно запоминать выход канала, соответствующий n последним переданным ветвям. Тогда отказ при декодировании возникнет всегда, когда декодирующее устройство вынуждено возвратиться назад на n узлов, предшествующих ветви, которая в настоящий момент передавалась. Иначе говоря, декодирующее устройство вынуждено принимать окончательное решение по каждой информационной цифре в течение заданного времени после ее передачи. Любая ошибка в этом окончательном решении, отличная от ошибок рассмотренного выше типа, будет полностью останавливать операцию декодирования. Для вероятности возникновения отказов при декодировании, соответствующих этому частному типу запоминающего устройства, не может быть получена обычная оценка.

Предположим далее, что выход канала записывается на магнитную ленту или другое буферное устройство

подобного типа, с которого отрезки, соответствующие последовательным ветвям, могут быть в случае надобности считаны декодирующим устройством по отдельности. Предположим также, что внутренняя память декодирующего устройства ограничена n ветвями. Тогда отказ при декодировании возникает всегда, когда декодирующее устройство возвращается на n ветвей назад от самой дальней ветви, которую оно когда-либо рассматривало, независимо от того, как далеко позади эта ветвь находится от той, которая в настоящий момент передается.

Обозначим через k порядковый номер последней ветви, сброшенной внутренней памятью декодирующего устройства. Существуют два различных случая, когда декодирующее устройство возвращается к этой ветви после того, как оно рассмотрит ветвь с порядковым номером $k+n$: 1) значение L на правильном пути спускается ниже L_k в некотором узле, порядковый номер которого равен или больше, чем $k+n$; 2) значение L спускается ниже некоторого порогового значения $T \leqslant L_k$ в некотором предыдущем узле, и существует неправильный путь, исходящий из узла с порядковым номером k, на котором значение L остается выше T вплоть до узла с порядковым номером $k+n$.

Оценка сверху для вероятности возникновения таких событий может быть легко найдена. Она подобна оценке (9), в которой $\nu = n\nu_0$, а значение α приблизительно равно тому, которое получается для порогового, блочного декодирования.

X. Заключение

Основное свойство последовательного декодирования, которое делает его особенно привлекательным на практике, состоит в том, что сложность требуемой аппаратуры возрастает только лишь линейно по ν в то время, как требуемая скорость работы не зависит от ν. Таким образом экономически возможно использование достаточно больших значений ν с тем, чтобы иметь пренебрежимо малую вероятность ошибки при скоростях передачи, относительно близких к пропускной способности [6].

Другая важная черта последовательного декодирования состоит в том, что способ его выполнения очень слабо

зависит от характеристик канала, и, следовательно, большая часть аппаратуры может быть использована для широкого класса каналов.

Наконец, следует подчеркнуть, что последовательное декодирование является по существу поиском процедуры „восходящего к вершине" типа. В принципе она может быть использована для различения любого множества возможностей, представленных деревом, в котором ветви, исходящие из различных узлов с одним и тем же порядковым номером, существенно отличаются одна от другой.

ЛИТЕРАТУРА

1. Peterson W. W., Error-correcting codes, M. I. T. Press, Cambridge, Mass, John Wiley and Sons, Inc., New York, N. Y., 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М, 1964.]
2. Fano R. M., Transmission of information, M. I. T. Press, Cambridge, Mass., John Wiley and Sons, Inc., New York, N. Y., 1961. [Готовится русский перевод.]
3. Shannon C. E., A mathematical theory of communication, *Bell. Sys. Tech. J.*, 27, 379—623; (July, October 1948). [Русский перевод издания 1948 г.: Шеннон К., Работы по теории информации и кибернетике, ИЛ, М., 1963 г., стр. 243—332.]
4. Wozencraft J. M., Sequential decoding for reliable communications, Res. Lab. of Electronics, M. I. T., Cambridge, Mass., Technical Rept., 325, 1957. [См. также Возенкрафт Д. М., Рейффен Б., Последовательное декодирование, ИЛ, М., 1963.]
5. Reiffen B., Sequential encoding and decoding for the discrete memoryless channel, Res. Lab. of Electronics, M. I. T., Cambridge, Mass., Technical Rept. 374, 1960.
6. Perry K. M., Wozencraft J. M., SECO: A self-regulating error-correcting coder. decoder, *IRE Trans. on Information Theory*, **IT-8**, September (1962), 125—135.
7. Ziv J., Coding and decoding of time discrete amplitude continuous memoryless channels, *IRE Trans. on Inormation Theory*, **IT-8**, September (1962), 199—205.
8. Gallager R. G., Low density parity-check codes, *IRE Trans. on Information Theory*, **IT-8**, January (1962), 21—28. [Русский перевод см. настоящий сборник стр. 139—165.]
9. Gallager R. G., Sequential decoding for binary channels with noise and synchronization errors, Lincoln Lab., M. I. T., Lexington, Mass., Rept. No 25G—2.
10. Wozencraft J. M., Horstein M., Coding for two-way channels, Information Theory, Fourth London Symposium, C. Cherry, Ed. Butterworths, Scientific Publications, London, England, 1961, p. 11.

ГРАНИЦЫ ДЛЯ КОДОВ, ИСПРАВЛЯЮЩИХ ПАКЕТЫ ОШИБОК [1])

К. Кампопиано

I. Введение

В недавно вышедшей книге Питерсона [2]) опубликованы результаты автора относительно границ для кодов, исправляющих открытые пакеты ошибок. В данной статье приводится аналогичная граница для кодов, исправляющих замкнутые пакеты ошибок, а также даются некоторые дополнительные оценки.

Все результаты формулируются для линейных кодов над полем Галуа. В разделе II мы суммируем некоторые элементы теории линейных кодов в удобной для наших целей форме. В разделе III приводятся оценки числа избыточных символов на кодовое слово для кодов, исправляющих пакеты ошибок. Численные результаты из табл. I и табл. II показывают, что приведенные здесь верхняя и нижняя границы часто очень близки.

II. Линейные коды [3])

Пусть $GF(q)$ — поле Галуа из q элементов. Пусть $V^n(q)$ — множество всех упорядоченных n-наборов с компонентами в $GF(q)$. $V^n(q)$ образует n-мерное пространство

---

[1]) Campopiano C., Bounds on burst-error-correcting codes, *IRE Transactions on Information Theory*, IT-8 (1962), № 3, 257—259.

[2]) Peterson W. W., Error-Correcting Codes, Technology Press and Wiley, New Jork, N. Y., 1961, 63. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964.]

[3]) Подробное описание и доказательства см. в главах 2 и 3 книги Питерсона (см. предыдущее примечание).

Таблица I

Нижняя граница r_* для числа проверочных символов

(Справедлива только при $n \geqslant 2l+1$)

	r_*			
n	$l=2$	$l=3$	$l=4$	$l=5$
5—7	4	6		
8—15	5	6	8	10
16—31	6	7	8	10
32—63	7	8	9	10
64—127	8	9	10	11
128—255	9	10	11	12
256—511	10	11	12	13

Таблица II

Верхняя граница r^* для минимального числа проверочных символов

n				r^*
$l=2$	$l=3$	$l=4$	$l=5$	
8				5
9—13				6
14—24				7
25—45	12—14			8
46—88	15—22			9
89—173	23—38			10
174	39—70	16—22		11
	71—134	23—34		12
	135—262	35—60	20—22	13
	263—518	61—111	23—33	14
		112—214	34—54	15
		215—418	55—97	16
		419—828	98—182	17
			183—353	18
			354—694	19

над $GF(q)$ относительно операций

$$(\alpha_1, \ldots, \alpha_n) + (\beta_1, \ldots, \beta_n) = (\alpha_1 + \beta_1, \ldots, \alpha_n + \beta_n),$$
$$\gamma(\alpha_1, \ldots, \alpha_n) = (\gamma\alpha_1, \ldots, \gamma\alpha_n),$$
$$\alpha_i, \beta_j, \gamma \in GF(q).$$

Если не оговорено противное, предполагается, что все векторы, принадлежащие $V^n(q)$, являются вектор-строками. Подмножества множества $V^n(q)$ назовем *кодами*. Элементы множества $V^n(q)$ назовем *векторами*, или *словами*. Элементы кода назовем *кодовыми словами*, или *кодовыми векторами*. Число векторов, принадлежащих коду C, назовем *объемом* C и обозначим через $|C|$. Подмножество $V^n(q)$ назовем *линейным кодом*, если оно является линейным подпространством пространства $V^n(q)$. В частности, линейный код размерности k в $V^n(q)$ назовем $L(q, n, k)$-кодом.

Мы говорим, что $L(q, n, k)$-код имеет k *информационных* и $(n-k)$ *проверочных символов* на кодовое слово. Если C есть $L(q, n, k)$-код, существует $n \times (n-k)$-матрица A с элементами из $GF(q)$ ранга $(n-k)$, такая, что код C совпадает с левым нуль-пространством матрицы A, т. е. с множеством всех векторов x из $V^n(q)$, таких, что $xA = 0$. Мы назовем A *кодовой матрицей* [1]) кода C.

Пусть Q — *дискретный канал с шумами* [2]), на входе и выходе которого допустимы любые элементы из $GF(q)$. Пусть C есть $L(q, n, k)$-код с кодовой матрицей A. Если через Q передается кодовое слово x из C, то на выходе получают вектор $x+e$. Вектор e называют *вектором шума* [3]).

Пусть $E = \{e_1, e_2, \ldots, e_m\}$ — множество всех возможных векторов шума, которые мы хотим исправлять. Под *процессом декодирования* мы понимаем следующий процесс:

---

[1]) Ее называют также *проверочной* матрицей. — *Прим. перев.*

[2]) Shannon C. E., Weaver W., Mathematical Theory of communication, University of Illinois Press, Urbana, Ill. 1949, p. 34. [Русский перевод издания 1948 г. см. Шеннон К., Работы по теории информации и кибернетике, ИЛ, М., 1963, стр. 275].

[3]) Его называют также *сочетанием ошибок*. — *Прим. перев.*

а) вычисление *проверочного вектора* [1])

$$c=(x+e)A=eA,$$

б) определение элемента E, например e_i (если он существует), такого, что

$$e_iA=c,$$

в) вычисление *декодированного вектора* $(x+e)-e_i$.

Говорят, что код C исправляет все векторы шума из E, если для любого вектора c в $V^{n-k}(q)$ существует самое большее один вектор в E, скажем e_i, такой, что $e_iA=c$. Следующий результат очевиден.

Теорема 1. *Левое нуль-пространство кодовой матрицы A исправляет все векторы шума в E тогда и только тогда, когда выполняются следующие эквивалентные условия:*

а) *для различных* e_i *и* $e_j\in E$, $e_iA\neq e_jA$;

б) *для различных* e_i *и* $e_j\in E$, $(e_i-e_j)A\neq 0$.

III. Коды, исправляющие пакеты ошибок

Пусть l—целое и $x=(\xi_1, \ldots, \xi_n)$—вектор, принадлежащий $V^n(q)$. Назовем x открытым пакетом длины l, если $1\leqslant l\leqslant n$ и все ненулевые компоненты x расположены среди l рядом стоящих элементов, первый и последний из которых отличны от нуля. Если $2\leqslant l\leqslant(n+1)/2$, x называют *замкнутым пакетом* длины l [2]), когда кроме того существует i, такое, что $1\leqslant i\leqslant l-1$, $\xi_i\cdot\xi_{n-l+i+1}\neq 0$, $\xi_{i+1}=\xi_{i+2}=\ldots=\xi_{n-l+i}=0$. В дальнейшем будем обозначать через $E_0(q, n, l)$ любое подмножество множества $V^n(q)$, состоящее из всех открытых пакетов длины l или меньше и нулевого вектора. Запись $E_c(q, n, l)$ будет обозначать подмножество мно-

[1]) Другое название—*синдром*.— *Прим. перев.*

[2]) Термины „открытый“ и „замкнутый“ были впервые использованы в этом значении в работе: Fire P., A class of multiple-error-correcting binary codes for non-independent error, Sylvania Reconnaissance Systems Lab., Mountain View, Calif., Sylvania Rept. RSL-2, 1959.

жества $V^n(q)$, состоящее из элементов множества $E_0(q, n, l)$ и всех замкнутых пакетов длины l или меньше. Пусть C есть $L(q, n, k)$-код с кодовой матрицей A. Положим, что для любого вектора x из $V^n(q)$ существует самое большее один вектор y в $E_0(q, n, l)$ $\{E_c(q, n, l)\}$, такой, что $yA = xA$. В этом случае говорят, что C — код, *исправляющий открытые (замкнутые) пакеты ошибок порядка* l, и пишут; что C есть $B_0(q, n, k, l)$ $\{B_c(q, n, k, l)\}$-код.

Мы получим теперь две нижние границы числа проверочных символов для $B_c(q, n, n-r, l)$-кода.

Теорема 2. *Число избыточных символов на кодовое слово* $B_0(q, n, k, l)\{B_c(q, n, k, l)\}$*-кода равно по крайней мере* $2l$.

Доказательство. Случай открытых пакетов приведен у Питерсона [1]). Для замкнутых пакетов теорема следует из того, что $B_c(q, n, k, l)$-код есть также $B_0(q, n, k, l)$-код.

Теорема 3. *Если существует* $B_c(q, n, k, l)$*-код, то* $r \geqslant \log_q [1 + n(q-1) q^{l-1}]$.

Доказательство. Пусть $B_c(q, n, k, l)$-код есть левое нуль-пространство матрицы A. Если e_1 и e_2 суть различные элементы множества $E_c(q, n, l)$, то $e_1 A \neq e_2 A$. Но для всех x в $V^n(q)$ xA принадлежит $V^r(q)$. Отсюда

$$|V^r(q)| \geqslant |E_c(q, n, l)|.$$

Однако при $n/2 > l \geqslant 2$ мы имеем

$$|E_c(q, n, l)| = 1 + n(q-1) + \sum_{i=2}^{l} (q-1)^2 q^{i-2} n =$$
$$= 1 + n(q-1) q^{l-1},$$

откуда и следует утверждение.

Эта теорема является обобщением одного результата Файра [2]).

---

[1]) См. Питерсон, стр. 78—79 (примечание 2 на стр. 199).
[2]) См. Файр, стр. 7 (примечание 1 на стр. 202).

Предыдущий результат определяет нижнюю оценку числа проверочных символов $B_c(q, n, n-r, l)$-кода. Мы получим теперь достаточное условие существования $B_c(q, n, n-r, l)$-кода, определяющее также верхнюю границу минимального числа проверочных символов $B_c(q, n, n-r, l)$-кода. Прежде всего из теоремы 1 мы сразу же получаем следующий результат [1]):

Теорема 4. *Пусть левое нуль-пространство матрицы A есть $L(q, n, n-r)$-код. Пусть строки A суть $a_1, a_2, \ldots, a_n$, т. е. положим*

$$A = \left\| \begin{matrix} a_1 \\ \vdots \\ a_n \end{matrix} \right\|. \tag{1}$$

Положим, что $n > r \geqslant 2b$. Для $i > n$ определим $a_i = a_{i-n}$. Левое нуль-пространство A будет тогда и только тогда $B_c(q, n, n-r, l)$-кодом, когда любое множество $2l$ векторов-строк A вида

$$a_i, a_{i+1}, \ldots, a_{i+l-1};$$
$$a_{i+l-1+j}, a_{i+l+j}, \ldots, a_{i+2l-2+j},$$
$$(i \geqslant 1,\ 1 \leqslant j \leqslant n-2l+1)$$

есть линейно независимое множество векторов над $GF(q)$.

Используя теорему 4, мы можем получить достаточное условие существования $B_c(q, n, n-r, l)$-кода следующим образом. Нам нужно построить матрицу A, удовлетворяющую теореме 4.

Положим, что n и l — фиксированные положительные целые и $n > 2l$. Для матрицы A мы будем использовать обозначение (1). Вначале положим $r = 2l$ и возьмем линейно независимое множество векторов $a_1, a_2, \ldots, a_{2l}$ в $V^r(q)$. Затем выберем вектор a_{2l+1}, удовлетворяющий теореме 4, увеличив r, если это необходимо. Продолжая таким же образом и увеличивая r, когда это необходимо, мы можем построить векторы $a_1, \ldots, a_{n-1}$, удовлетворяющие теореме 4. Теперь нужно найти такой элемент a_n, чтобы матрица A вида (1) удовлетворяла теореме 4.

[1]) Более общий результат можно найти у Питерсона, теорема 3.1, стр. 47 (см. примечание 2 на стр. 199),

Поэтому необходимо, чтобы имело место неравенство

$$a_n \neq \sum_{\substack{s=i \\ s \neq n}}^{i+l-1} \alpha_s a_s + \sum_{t=j}^{j+b-1} \beta_t a_t, \qquad (2)$$

где $n-l+1 \leqslant i \leqslant n$, $i+l \leqslant j \leqslant n+i-1$, $a_{i+n}=a_i$ и α_s, β_t принадлежат $GF(q)$. Перепишем (2) так, чтобы было удобно подсчитывать число систем с различными коэффициентами в правой части выражения (2). Чтобы сделать это без применения длинных формул, введем дополнительные обозначения. Для $x=(\xi_1, \ldots, \xi_s)$ и $y=(\eta_1, \ldots, \eta_m)$ положим $(x, y)=(\xi_1, \ldots, \xi_s, \eta_1, \ldots, \eta_m)$; символы $x^{(s)}$, $y^{(s)}$ будут обозначать векторы из $V^s(q)$. Через $0^{(s)}$ обозначим нулевой вектор в $V^s(q)$. Пусть $\tilde{E}_0(q, n, l)$ — множество всех открытых пакетов длины l в $V^n(q)$. Определим

$$A' = \left\| \begin{matrix} a_1 \\ \vdots \\ a_{n-1} \end{matrix} \right\|.$$

Пусть $\{x \mid P(x)\}$ обозначает множество всех x со свойством $P(x)$. Допустим, кроме того, что $n \geqslant 4l$. Тогда неравенство (2) эквивалентно тому условию, что a_n не принадлежит множеству $\{xA' \mid x \in X\}$, где X есть объединение следующих непересекающихся подмножеств из $V^{n-1}(q)$:

$$\begin{aligned}
X_0 = \{ x \mid x = (&x^{2l-1}, 0^{(n-4l+1)}, y^{(2l-1)}); \\
&(y^{(2l-1)}, x^{(2l-1)}) \in E_0(q, 2(2l-1), 2l-1)\}; \\
X_1 = \{ x \mid x = (&x^{(l-1)}, 0^{(l)}, z^{(n-4l+1)}, 0^{(l)}, y^{(l-1)}). \\
&(y^{(l-1)}, x^{(l-1)}) \in E_0(q, 2(l-1), l-1); \\
&z^{(n-4b+1)} \in \tilde{E}_0(q, n-4b+1, 1)\}; \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
X_t = \{ x \mid x = (&x^{(l-1)}, 0^{(1)}, z^{(n-2l-1)}, 0^{(1)}, y^{(l-1)}); \\
&(y^{(l-1)}, x^{(l-1)}) \in E_0(q, 2(l-1), l-1); \\
&z^{(n-2l-1)} \in \tilde{E}_0(q, n-2l-1, e)\}.
\end{aligned}$$

Заметим, что у множества X есть два непересекающихся

подмножества. Элементы множества X_0 обладают следующим свойством: их $(2l)$-я, $(2l+1)$-я, ..., $(n-2l)$-я компоненты равны 0, тогда как члены объединения $X_1, \ldots, X_l$ имеют по крайней мере одну из компонент отличной от нуля. Из предыдущего получаем:

$$|X| = |E_0(q,\ 2(2l-1),\ 2l-1)| + \\ + |E_0(q,\ 2(l-1),\ l-1)| \sum_{i=1}^{l} |\tilde{E}_0(q,\ n-4l+2i-1,\ i)|.$$

Легко видеть, что

$$|\tilde{E}_0(q,\ n,\ l)| = \begin{cases} (n-l+1)(q-1)^2 q^{l-2}, & l \geqslant 2 \\ n(q-1), & l = 1. \end{cases}$$

Кроме того, известно [1]), что

$$|E_0(q,\ n,\ l)| = q^{l-1}[(q-1)(n-l+1)+1].$$

Отсюда получаем соотношение

$$|X| = q^{2l-2}[(q-1)2l+1] + \\ + q^{l-2}[(q-1)l+1] \cdot \{q^{l-1}[(n-3l)(q-1)-1]+1\},$$

когда $n \geqslant 4l$.

Поскольку всегда можно найти число a_n, удовлетворяющее (2), когда $q^r > |X|$, мы делаем следующий вывод.

Теорема 5. *Пусть n и l—целые и $n \geqslant 4l \geqslant 8$. Тогда существует $B_c(q,\ n,\ n-r,\ l)$-код с r проверочными символами, где r удовлетворяет условию*

$$q^n > q^r > q^{2l-2}[(q-1)2l+1] + \\ + q^{l-2}[(q-1)l+1]\{q^{l-1}[(n-3l)(q-1)-1]+1\}. \quad (3)$$

Теорема 5 несколько ограничена в том смысле, что требует выполнения неравенства $n \geqslant 4l$. С другой стороны, она охватывает большой диапазон значений n, имеющих практическое значение. Теорема 5 вместе с теоремами 2 и 3 должна в этих случаях очень хорошо показывать, чего можно достичь в классе кодов, исправляющих замкнутые пакеты ошибок.

[1]) См. Питерсон (примечание 2 на стр. 199), стр. 62.

Пусть n, l и q фиксированы и $n \geqslant 2l+1$. Пусть r_* — наименьшее целое, удовлетворяющее условиям $r_* \geqslant 2l$, и

$$r_* \geqslant \log_q [1 + (q-1)\, nq^{l-1}].$$

Тогда для $B_c(q,\ n,\ n-r,\ l)$-кода имеем $r \geqslant r_*$. Пусть $n \geqslant 4l$ и r^* — наименьшее целое, удовлетворяющее (3). Тогда для $n \geqslant 4l$ наименьшее значение r, для которого существует $B_c(q,\ n,\ n-r,\ q)$-код, должно удовлетворять условию $r_* \leqslant r \leqslant r^*$. Мы даем значения r_* и r^* для $GF(2)$ в таблицах I и II. Заметим, что данные табл. I и II справедливы только при $n \geqslant 2l+1$.

НОВАЯ ВЕРХНЯЯ ГРАНИЦА ДЛЯ КОДОВ, ИСПРАВЛЯЮЩИХ ОШИБКИ [1])

Селмер М. Джонсон

Краткое содержание. С помощью усовершенствованной модели упаковки сфер Хэмминга [1] найдена новая верхняя граница для несистематических двоичных кодов исправляющих ошибки. Используется только аппарат комбинаторики.

В то время как вычисление верхней оценки Хэмминга для кодов, исправляющих e ошибок, содержит подсчет всех точек, находящихся на расстоянии меньшем чем e от множества кодовых точек, представленная здесь модель включает рассмотрение точек, которые находятся на расстоянии большем чем e от множества кодовых точек. Некоторое улучшение оценки Хэмминга иногда оказывается довольно значительным для случая исправления двух или более ошибок. Новая оценка лучше оценки Вакса [2] для всего его списка, кроме четырех случаев.

Введение

Мы предполагаем, что читатель знаком с наиболее известными работами по кодам, исправляющим ошибки. Предметом этой статьи является построение новой верхней оценки числа кодовых точек несистематического кода с исправлением ошибок. Геометрическая интерпретация рассматриваемой задачи состоит в следующем. Найти максимальное подмножество вершин n-мерного единичного куба, такое, что расстояние Хэмминга между двумя любыми вершинами этого подмножества не менее d; расстояние измеряется вдоль ребер куба. Для кода, исправляющего e ошибок, $d = 2e + 1$.

Эти вершины суть кодовые точки, которые Хэмминг [1] окружает сферами радиуса e. Он подсчитывает число

---

[1]) Jonson M., A new upper bound for error-correcting codes. *IRE Trans. on Information Theory*, IT-8 (1962) № 3, 203—207,

вершин, которые отстоят от кодового слова не более чем на e, т. е. находятся внутри сферы. Если поделить 2^n на это число, то получается верхняя оценка количества n-значных последовательностей двоичного кода, исправляющего e ошибок.

С другой стороны, Вакс [2] применяет непрерывную модель упаковки сфер с функцией переменной плотности, чтобы получить верхнюю оценку, которая выгоднее оценки Хэмминга для некоторых значений n и e. Его вычисления чрезвычайно сложны и были выполнены на электронной машине.

В настоящей статье мы возвращаемся к дискретной модели Хэмминга и так расширяем подсчет точек, чтобы он содержал в себе оценки нижней границы числа точек, которые находятся вне пространства, занятого сферами Хэмминга, упакованными в n-мерном кубе.

В то время как этот метод может быть формально описан для задачи упаковки сфер при любой метрике, специальные свойства метрики Хэмминга ведут к полезным результатам, использующим только элементарные комбинаторные рассуждения и расчеты вручную.

Результаты для этих моделей упаковки сфер сравниваются в табл. I. Из нее видно, что новые оценки оказываются лучше результатов Вакса во всех случаях, кроме четырех.

Существуют другие методы, дающие обычно более хорошие верхние границы, такие, как граница Плоткина [3], но они применимы только к случаям, где $n < 2d$ или иногда немного более. Следовало бы отметить, что настоящий метод в соединении со специальными рассуждениями может дать улучшенные результаты для многих случаев, где $n < 2d$, но эти рассуждения стоят в стороне от основных идей нашей модели упаковки сфер и в настоящей работе не используются. Асимптотически предлагаемый метод значительно превосходит все известные автору методы для больших n и e в грубо ограниченной области, где $4e < n < e^2$ наряду с меньшими улучшениями, когда n превосходит e^2. Эти асимптотические оценки будут представлены во второй статье.

Таблица 1

Результаты оценки верхней границы

n	e	1	2	3	4	5	6
7	H	16	4,4	2			
	W	18,2	3,1	2,1			
	R	7	1	1			
	J	16	3,0	2			
	N	16	2	2			
8	H	28,4	6,9	2,8			
	W	20,3	5,3	3			
	R	8	2	1			
	J	25,6	4,6	2,3			
	N	20	4	2			
9	H	51,2	11,1	3,9	2		
	W	39,7	9,2	4,2	2,7		
	R	12	3	1	1		
	J	51,2	8	2,9	2		
	N	38	6	2	2		
10	H	93,1	18,3	5,8	2,7		
	W	82,2	16,6	6,1	3,4		
	R	13	6	1	1		
	J	83,9	13,4	3,8	2,2		
	N	68	12	2	2		
11	H	170,7	30,6	8,8	3,6	2	
	W	154,8	26,5	8,6	4,3	2,8	
	R	17*	11	2	1	1	
	J	160	24	5,3	2,8	2	
	N	128	24	4	2	2	
12	H	315,1	51,8	13,7	5,2	2,6	
	W	346,8	46,7	12,7	5,7	3,3	
	R	20	12*	3	1	1	
	J	292,5	39,3	9,5	3,6	2	
	N	256	32	4	2	2	
13	H	585,1	89,0	21,7	7,5	3,4	2
	W	806	85,1	19,3	7,2	4,0	2,9
	R	26	19*	4	1	1	1
	J	585,1	70	14,3	4,8	2,7	2
	N	512	32	8	2	2	2

Продолжение табл. I

n	e	1	2	3	4	5	6
14	H	1 092	154,6	34,9	11,1	4,7	2,5
	W	1 913	160	30,1	9,8	4,9	3,3
	R	28	29 *	8	2	1	1
	J	1 024	131	22,9	6,9	3,4	2,2
	N	1 024	64	16	4	2	2
15	H	2 048	270,8	56,9	16,9	6,6	3,3
	W	4 656	309,8	48,9	13,8	6,2	3,9
	R	35	42	15	3	1	1
	J	2 048	256	38,2	11,6	4,5	2,6
	N	2 048	128	32	4	2	2
16	H	3 860	478,4	94,0	26,0	9,5	4,4
	W	11 600	617,3	82,0	24,2	8,8	4,7
	R	37	48	22	4	1	1
	J	3 616	428,3	68,3	17,2	6,1	3,3
	N	2 048	128	32	6	2	2
17	H	7 280	851,1	157,2	40,8	13,9	6,0
	W	29 600	1 264	140	37,5	12,1	5,9
	R	44 *	68	37 *	6	2	1
	J	7 090	851,1	118	26,0	9,9	4,2
	N	4 096	256	64	8	4	2

H — граница Хэмминга для $A(n, d)$.
W — граница Вакса для $A(n, d)$.
R — лучшая граница для $R(n, d, e)$, полученная по (5),(6) и (7).
J — граница для $A(n, d)$, полученная с помощью (2), (5), (6) и (7).
N — наилучшая длина кода.

* Улучшенное значение R для этого случая. См. добавление.

Вывод новой оценки

В то время как основа этой статьи геометрическая, рассуждения главным образом алгебраические. Алгебраическая постановка задачи состоит в следующем.

Пусть $A(n, d)$ — максимальное число строк матрицы с n столбцами, состоящей из нулей и единиц и обладающей

тем свойством, что любые две строки отличаются друг от друга не менее чем в d позициях.

Попытаемся усовершенствовать верхнюю границу для $A(n, d)$. Как хорошо известно, эта функция представляет собой максимальную мощность кода длины n, корректирующего e ошибок, где $d=2e+1$. Нам понадобится определить несколько вспомогательных функций. Рассмотрим вектор, состоящий из m элементов 0 и 1, имеющий вес r. Тогда пусть $R(m, r, \lambda)$ есть максимальное число таких векторов длины m и веса r, что скалярное произведение любой пары из них не превосходит λ. Таким образом, два любых вектора нѐ должны иметь более чем λ единиц на одних и тех же позициях. Например, можно проверить, что $R(5, 3, 1)=2$. Векторы (1, 1, 1, 0, 0) и (0, 0, 1, 1, 1) являются такой парой.

Наша оценка верхней границы для $A(n, d)$ будет существенно зависеть от функции $R(m, r, \lambda)$, которая тесно связана с работой о блок-схемах в комбинаторной математике [4] [1]), но отличается от обычного определения тем, что любых два вектора должны иметь не в точности λ общих единиц, а не более чем λ общих единиц. Две элементарные оценки верхней границы для $R(m, r, \lambda)$ будут представлены в этой статье.

Перейдем к изучению этой функции.

Для двух векторов, отличающихся индексами i и j, имеем соотношение

$$r_i + r_j = 2\lambda_{ij} + d_{ij}, \tag{1}$$

где r_i и r_j—веса векторов, λ_{ij}—число позиций, на которых в обоих векторах стоят единицы, т. е. скалярное произведение, и d_{ij}—число позиций, на которых один вектор имеет единицу, а другой—нуль. Это простое соотношение будет неоднократно использовано в наших исследованиях. Наше первое усовершенствование границы Хэмминга состоит в следующем.

---

[1]) Для ознакомления с понятиями „блок-схема“, „сбалансированная блок-схема“ (см. стр. 216) и „тройка Штейнера“ (см. стр. 223) можно обратиться к вышедшей на русском языке книге Холла „Комбинаторный анализ“, ИЛ, М., 1963 г., гл. IV.— *Прим. перев.*

Теорема 1.

$$A(n, d) \leqslant \frac{2^n}{1+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{e}+\dfrac{\binom{n}{e+1}-\binom{d}{e}R(n, d, e)}{\left[\frac{n}{e+1}\right]}}, \quad (2)$$

где, как и на протяжении всей статьи, для числа сочетаний и символа целой части использованы стандартные обозначения.

Наше доказательство основано на усовершенствовании геометрической модели Хэмминга [1]. Пусть E_n — множество 2^n вершин n-мерного единичного куба, которое разбито на непересекающиеся подмножества следующим образом:

S_0 — искомое максимальное подмножество вершин, расстояние Хэмминга между которыми не менее d;

S_k — подмножество вершин, расстояние от которых до вершин подмножества S_0 равно k.

Тогда

$$E_n = S_0 + S_1 + \ldots + S_{d-1}. \quad (3)$$

Действительно, если бы нашлась хоть одна точка в S_k при $k \geqslant d$, она должна была бы относиться к S_0, которое по определению максимально возможное. Вспомним, что в верхней оценке Хэмминга рассмотрены только множества S_0, S_1, ..., S_e. Мы улучшим результат Хэмминга тем, что рассмотрим сначала множества S_{e+1}. Дальнейшее расширение комплекта множеств S_k для $k > e+1$ будет обсуждено в этой статье ниже.

Чтобы оценить нижнюю границу числа точек в S_{e+1}, рассмотрим произвольно выбранную точку P в S_0. Подходящим выбором координатных осей преобразуем эту точку к началу; геометрически — вращением единичного куба. Алгебраически это выполняется заменой на противоположные всех элементов тех столбцов матрицы координат, в которых стоят единичные координаты точки P. Ясно, что это не влияет на расстояние Хэмминга между любыми двумя вершинами.

Сосредоточим внимание на окрестности этой начальной кодовой точки $P_0 = (0, 0, \ldots, 0)$ в S_0. В единичном кубе

имеется $\binom{n}{k}$ вершин веса k и потому находящихся на расстоянии k от P. Множество таких точек образует гиперплоскость W_k. Любая точка в W_k принадлежит, таким образом, некоторому S_i для $i \leqslant k$.

Кодовые точки веса r будут обозначаться символом P_r; P_r содержится в $W_r \cap S_0$. Точки веса r, не принадлежащие коду, будут обозначаться символом Q_r; Q_r содержится в $W_r \cap S_i$ для некоторого $i > 0$. Пусть $c(X)$ — мощность множества точек X. Тогда число точек, принадлежавших обоим множествам S_0 и W_d, есть $c(W_d \cap S_0) \leqslant R(n, d, e)$, как следует из (1).

Если две точки содержатся в S_0 и W_d, то $r_i = r_j = d$ и, таким образом, d_{ij} должно быть четным. Так как d нечетно, мы должны иметь $d_{ij} \geqslant d+1$. Из этого следует, что $2\lambda_{ij} + d + 1 \leqslant 2d$ или $\lambda_{ij} \leqslant e$.

Каждая из этих точек P_d имеет $\binom{d}{e}$ точек Q_{e+1} в $W_{e+1} \cap S_e$, которые находятся на расстоянии e от P_d. Это можно увидеть, вычеркивая e из d единичных координат точки P_d. Так как $d + e + 1 = 2\lambda_{ij} + e$, из [1] имеем $\lambda_{ij} = e + 1$. Таким образом, каждая единичная координата точки Q_{e+1} встречается также среди координат соответствующей точки P_d. Более того, эти e-окрестности множества $W_d \cap S_0$ все различны в W_{e+1}, так как любые две точки в $W_d \cap S_0$ отстоят друг от друга не менее чем на $d = 2e + 1$, фактически же не менее чем на $d+1$. Из равенства $W_{e+1} = (W_{e+1} \cap S_e) + (W_{e+1} \cap S_{e+1})$ мы получаем

$$c(W_{e+1} \cap S_{e+1}) \geqslant \binom{n}{e+1} - \binom{d}{e} R(n, d, e).$$

Точка Q_{e+1} в $W_{e+1} \cap S_{e+1}$ находится на расстоянии $e+1$ не более чем от $[n/(e+1)]$ точек в S_0. В этом можно убедиться, если преобразовать к началу точку Q_{e+1} и посмотреть сколько точек веса $e+1$ может находиться друг от друга на расстоянии не меньшем чем d, или, точнее, $d+1$, так как d нечетно. Здесь

$$2(e+1) = 2\lambda_{ij} + 2(e+1), \quad \text{или} \quad \lambda_{ij} = 0.$$

Таким образом, мы находим $R(n, e+1, 0) = [n/(e+1)]$, что представляет собой число способов, которыми можно

выбрать непересекающиеся множества $e+1$ точек из их общего числа n. Этот результат был указан Ллойдом Шепли. Если мы теперь просуммируем по всем точкам в S_0, учитывая их k-окрестности для $k \leqslant e+1$, то все точки k-окрестности для $k \leqslant e$ единственным образом будут соответствовать одной точке в S_0 и поэтому будут считаться в точности один раз. С другой стороны, точки в S_{e+1} могут быть сосчитаны $[n/(e+1)]$ раз, так что мы должны поделить оценку нижней границы $(e+1)$-окрестности на

$$[n/(e+1)] = R(n,\ e+1,\ 0).$$

Таким образом, получаем неравенство

$$A(n,\ d)\left\{1+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{e}+\right.$$
$$\left.+\frac{\binom{n}{e+1}-\binom{d}{e}R(n,\ d,\ e)}{\left[\frac{n}{e+1}\right]}\right\} \leqslant 2^n,$$

что и дает теорему 1.

Будем теперь искать подходящие верхние оценки для $R(n,\ d,\ e)$. Два простых приема будут представлены здесь. Изучаются более усовершенствованные оценки; они, быть может, будут опубликованы в следующей статье.

Теорема 2.

$$R(n,\ d,\ e) \leqslant \left[\frac{n}{d}\left[\frac{n-1}{d-1}\left[\ldots\left[\frac{n-e}{d-e}\right]\ldots\right]\right]\right]. \qquad (4)$$

Заметим, что здесь квадратные скобки означают вложенные одна в другую целые части.

Докажем сначала более общий результат

$$R(m,\ r,\ \lambda) \leqslant \left[\frac{m}{r}R(m-1,\ r-1,\ \lambda-1)\right]. \qquad (5)$$

В этом можно убедиться при рассмотрении матрицы из m столбцов и R строк, элементы которой суть нули и единицы, и такой, что $R = R(m,\ r,\ \lambda)$ есть максимальное

число вектор-строк длины m и веса r, скалярное произведение любой пары которых не превосходит λ.

Максимальное число единиц в некотором данном столбце, или иначе, максимальная сумма по столбцу, не превосходит $R(m-1, r-1, \lambda-1)$, так как рассматриваются векторы с единицами в этом столбце. После вычеркивания этого столбца длина, вес и попарные скалярные произведения этих векторов уменьшатся на единицу. Так как каждая сумма по столбцу не превосходит $R(m-1, r-1, \lambda-1)$, то общая сумма по всем столбцам не превосходит $mR(m-1, r-1, \lambda-1)$, как и сумма по всем строкам, которая равна $rR(m, r, \lambda)$. Многократное применение неравенства (5) дает теорему 2, так как $R(n-e, d-e, e-e)=[(n-e)/(d-e)]$. Когда $r^2>m\lambda$ по аналогии с комбинаторным изучением сбалансированных блок-схем (см., например, [4]), можно получить другую верхнюю границу для

$$R=R(m, r, \lambda).$$

Рассмотрим матрицу с R строками и m столбцами, такую, что суммы по строкам есть r и максимальное, по всем парам, скалярное произведение строк есть λ.

Подсчет по строкам суммы всех скалярных произведений строк по всем упорядоченным парам (i, j) дает

$$\sum_{i=1}^{R}\sum_{j=1}^{R}\lambda_{ij}=R(R-1)\bar{\lambda}\leqslant R(R-1)\lambda,$$

где $\bar{\lambda}$ — среднее значение скалярного произведения строк.

Подсчет по столбцам той же суммы всех скалярных произведений строк (если k_j есть сумма в j-ом столбце) дает

$$\sum_{j=1}^{m}k_j(k_j-1)=\sum_{j=1}^{m}k_j^2-rR,$$

так как

$$\sum_{j=1}^{m}k_j=\sum_{i=1}^{R}r_i=Rr.$$

Таким образом,

$$R(R-1)\lambda\geqslant R(R-1)\bar{\lambda}=\sum_{j=1}^{m}k_j^2-Rr.$$

Теперь, при условии, что $\sum_{j=1}^{m} k_j = Rr$, сумма $\sum_{j=1}^{m} k_j^2$ принимает минимальное значение, когда величины k_j по возможности приблизительно одинаковы. Если $k = [rR/m]$ — целая часть от rR/m, так что $rR = mk + t$, $0 \leqslant t \leqslant m$, то мы получаем верхнюю границу для $R(m, r, \lambda)$ в качестве наибольшего R, удовлетворяющего условию

$$R(R-1)\lambda \geqslant (m-1)k^2 + t(k+1)^2 - rR. \tag{6}$$

Первое приближенное значение величины R может быть найдено путем усреднения величин k_j, если пренебречь условием, что k_j целые. В этом случае $m\bar{k} = rR$,

$$\sum_{j=1}^{m} \bar{k}^2 = r^2R^2/m,$$

и

$$R(R-1)\lambda \geqslant \frac{r^2R^2}{m} - rR.$$

Решение этого неравенства относительно R дает следующую теорему.

Теорема 3. *Если* $r^2 > m\lambda$, *то*

$$R(m, r, \lambda) \leqslant \left[\frac{m(r-\lambda)}{r^2 - m\lambda}\right]. \tag{7}$$

Это приближенное значение может быть использовано для $R(m, r, \lambda)$, когда $r^2 > m\lambda$. Его можно использовать также в неравенстве (6), которое в иных случаях еще более понижает нашу оценку верхней границы. Таким образом, мы можем использовать комбинацию неравенств (5), (6) и (7), применение которых при любом частном сочетании параметров всякий раз приводит к относительно простому процессу подсчета для $R(n, d, e)$.

Теорема 3 дает почти всегда более сильный результат, чем теорема 2; поэтому попытаемся выяснить, когда она применима. Если $r^2 > m\lambda$, мы используем (7) и затем (6). Если $r^2 \leqslant m\lambda$, мы многократно используем (5), уменьшая r, m и λ каждый раз на единицу. В конце концов мы либо придем к теореме 2, либо после t-кратного применения неравенства (5) для некоторого наименьшего

$t<e$ получим $(d-t)^2>(n-t)(e-t)$, когда применима теорема 3. Иногда $(t+1)$-кратное применение неравенства (5) дает более хорошие результаты, если $(d-t)^2-(n-t)(e-t)$ оказывается достаточно малым положительным целым числом. Таким образом, мы получим значительное улучшение границы Хэмминга для случаев, грубо ограниченных неравенством $n<e^2+5e+2$. В противном случае мы достигнем только незначительного улучшения, получаемого путем повторения несколько раз неравенства (5).

Для иллюстрации использования всех этих соотношений (5), (6) и (7) рассмотрим пример $R(20, 7, 3)$. Здесь (7) неприменимо.

В силу соотношения (5) имеем

$$R(20, 7, 3)\leqslant\left[\frac{20}{7}R(19, 6, 2)\right]\leqslant\left[\frac{20}{7}\left[\frac{19}{6}R(18, 5, 1)\right]\right].$$

По соотношению (7) находим

$$R(18, 5, 1)\leqslant\left[\frac{18(5-1)}{25-18}\right]=\left[\frac{72}{7}\right]=10.$$

Однако, если $R(18, 5, 1)=10$, мы должны прийти к противоречию с соотношением (6), так как

$$10\cdot9\cdot1=90<92=14\cdot3^2+4\cdot(2)^2-50.$$

Таким образом,

$$R(18, 5, 1)\leqslant9,$$

$$R(20, 7, 3)\leqslant\left[\frac{20}{7}\left[\frac{19}{6}\cdot(9)\right]\right]=\left[\frac{20\cdot(28)}{7}\right]=80.$$

По теореме 1

$$A(20, 7)\leqslant\frac{2^{20}}{1+\binom{20}{1}+\binom{20}{2}+\binom{20}{3}+\dfrac{\binom{20}{4}-\binom{7}{3}(80)}{\left[\frac{20}{4}\right]}}$$

и мы понизили верхнюю границу с хэмминговой 776 до 595. В качестве метода сравнения верхних границ для

$A(n, d)$, выведенных из этих моделей упаковки сфер, мы частично воспроизведем таблицу, примененную Ваксом.

Профессор Боуз дал некоторые новые значения $A(n, d)$ для хорошо известных кодов.

Грайсмер [5] скомбинировал границы Хэмминга со своими собственными и другими приемами, чтобы улучшить оценки верхней границы для групповых кодов. В этой связи улучшенная оценка, данная в нашей статье, может иногда использоваться для получения дальнейших усовершенствований границ для групповых кодов.

Распространение на S_k при $k > e+1$.

Для случая $k = e+2$ существуют четыре множества точек, которые должны быть изъяты из общего числа $\binom{n}{e+2}$ точек в W_{e+2}. Любая точка в W_{e+2} должна содержаться в одном из множеств S_{e-1}, S_e, S_{e+1}, S_{e+2}.

Рассуждая как прежде, мы можем заключить из (1), что точки $W_{e+2} \cap S_{e-1}$, т. е. точки, принадлежащие одновременно множествам W_{e+2} и S_{e-1}, образуются вычеркиванием $e-1$ единиц из d единиц среди координат вектора каждой кодовой точки в W_d. Таким образом, существует не более $\binom{d}{e-1} R(n, d, e)$ точек в $W_{e+2} \cap S_{e-1}$. Точки множества $W_{e+2} \cap S_e$ образуются вычеркиванием e единиц из $d+1$ единиц среди координат вектора каждой кодовой точки в W_{d+1}.

Таким образом, существует не более $\binom{d+1}{e} R(n, d+1, e+1)$ точек в $W_e \cap S_e$, так как из (1) видно, что существует не более $R(n, d+1, e+1)$ кодовых точек в W_{d+1}.

Точки в $W_{e+2} \cap S_{e+1}$ находятся на расстоянии $e+1$ от кодовых точек в W_d или W_{d+2} или одновременно от точек обоих множеств.

Таким образом, существует ровно

$$\binom{d+2}{e+1} R(n, d+2, e+2)$$

или меньше точек, которые находятся на расстоянии

$e+1$ от некоторых кодовых точек в W_{d+2}. Существуют не более $\binom{d}{e}\binom{n-d}{1}R(n, d, e)$ точек в $W_{e+2}\cap S_{e+1}$, которые находятся на расстоянии $e+1$ от кодовых точек в W_d, так как, согласно (1), мы имеем $e+2+d=2\lambda_{ij}+e+1$, откуда $\lambda_{ij}=e+1$. Таким образом, мы можем вычеркнуть e из d единичных координат кодовой точки P_d и добавить одну новую единицу, чтобы образовать координаты точки Q_{e+2}, которая окажется на расстоянии $e+1$.

Задача здесь состоит в том, чтобы оценить нижние границы для точек в $W_{e+2}\cap S_{e+1}$, которые находятся на расстоянии $e+1$ от нескольких кодовых точек либо одного из множеств W_d и W_{d+2}, либо одновременно обоих. Она пока еще не решена. Игнорируя это пересечение, мы все же можем получить некоторую грубую границу числа точек в $W_{e+2}\cap S_{e+2}$, которая выглядит следующим образом:

$$\binom{n}{e+2}-\binom{d}{e-1}R(n, d, e)-\binom{d}{e}\binom{n-d}{1}R(n, d, e)-$$

$$-\binom{d+1}{e}R(n, d+1, e+1)-\binom{d+2}{e+1}R(n, d+2, e+2). \tag{8}$$

Это число нужно еще поделить на $R(n, e+2, 1)$, т. е. на максимальное число кодовых точек, которые могут находиться на расстоянии $e+2$ от некодовых точек в $W_{e+2}\cap S_{e+2}$. Это следует из (1) путем рассуждений, аналогичных тем, которые использованы для точек в $W_{e+1}\cap S_{e+1}$.

Вообще, оценка нижней границы числа точек в $W_{e+t}\cap S_{e+t}$ для $t\geqslant 1$ включает вычитание из $\binom{n}{e+t}t^2$ членов указанного типа. Эти члены представляют собой верхние границы подсчета точек в множествах, которые могут пересекаться, и, следовательно, дают несколько грубые оценки.

Полагая $r_1=e+t$, мы получаем наше основное улучшение для $A(n, d)$ следующим образом.

Теорема 4.

$$A(n, d) \leqslant \frac{2^n}{\sum_{i=0}^{e} \binom{n}{i} + \sum_{r_1=e+1}^{2e} \frac{c(W_{r_1} \cap S_{r_1})}{R(n, r_1, r_1-e-1)}}, \tag{9}$$

где

$$c(W_{r_1} \cap S_{r_1}) \geqslant \binom{n}{r_1} - \sum_{r_2=d}^{2r_1-1} R(n, r_2, r_2-e-1) \sum_{\lambda_{12}=[(r_2+1)/2]}^{r_1} \binom{r_2}{\lambda_{12}} \binom{n-r_2}{r_1-\lambda_{12}} \tag{10}$$

для каждого $r_1 > e$, *такого, что* $c(W_{r_1} \cap S_{r_1}) > 0$.

Из (1) и соотношения $r_2 - r_1 \leqslant d_{12} \leqslant r_1 - 1$ мы видим, что $[(r_2+1)/2] \leqslant \lambda_{12} \leqslant r_1$. Таким образом, для каждой кодовой точки P_{r_2} мы имеем $\binom{r_2}{\lambda_{12}}\binom{n-r_2}{r_1-\lambda_{12}}$ точек в W_{r_1}, которые находятся на расстоянии $r_1 + r_2 - 2\lambda_{12}$ от P_{r_2}. Такая точка Q_{r_1} образуется путем сохранения λ_{12} единиц из r_2 единиц в координатах точки P_{r_2} и добавления $r_1 - \lambda_{12}$ новых единиц в позициях, выбранных из $n - r_2$ позиций, содержащих нулевые координаты точки P_{r_2}. Выведена оценка для случая $A(100, 41)$ и приводятся результаты, которые дают представление об относительной величине (размерах) множеств $W_{e+t} \cap S_{e+t}$. Хэммингова сумма

$$\sum_{i=0}^{20} \binom{100}{i} = 7{,}1 \times 10^{20}.$$

Первый поправочный член, представляющий собой

$$c(W_{21} \cap S_{21})/R(100, 21, 0),$$

дает $5{,}1 \times 10^{20}$. Последующие шаги дают $14{,}6 \times 10^{20}$; $40{,}9 \times 10^{20}$; 104×10^{20}; 130×10^{20} для $t = 2, 3, 4, 5$. Почти все точки в W_{e+t} содержатся также и в S_{e+t}: при $t = 1$ и 2, не менее 98% общих точек — при $t = 3{,}91\%$ при $t = 4{,}48\%$ при $t = 5$ и наша грубая оценка дает 0% при $t > 5$. Таким образом. пренебрегая, как указано, пересечением множеств, мы получаем верхнюю границу $A(100, 41)$, составляющую $^1/_{42}$ значения, полученного Хэммингом

Дальнейшее исследование асимптотической оценки

Более детальные рассуждения ведут к тому результату, что когда e и n растут, почти все точки в W_{e+t} принадлежат S_{e+t} при $0<t\leqslant g(F)$, где $g(F)$ — сложная функция от $F=n/e$. Таким образом, например, $g(4)\approx e$, $g(5)\approx 0{,}3\ e$, $g(6)\approx 0{,}2\ e$, $g(10)\approx 0{,}09\ e$ и т. д., где десятичные коэффициенты взяты только приблизительно.

Это может быть интерпретировано кривой, которая проходит ниже кривых, изображающих асимптотические верхние границы Плоткина и Хэмминга, как показано на рис. 4.1 в [6]. Этот результат будет опубликован в другой статье[1]).

Добавление

Следующее неравенство иногда уточняет наши оценки для $R(n,\ d,\ e)$:

$$R(m,\ r,\ \lambda)\leqslant\left[\frac{m}{m-r}R(m-1,\ r,\ \lambda)\right]. \tag{11}$$

Это можно увидеть, если в матрице сначала заменить все нули единицами и наоборот, а затем, применив (5), снова произвести замену нулей на единицы и наоборот. Таким образом,

$$\begin{aligned}&R(m,\ r,\ \lambda)=R(m,\ m-r,\ m-r-(r-\lambda))\leqslant\\&\leqslant\left[\frac{m}{m-r}R(m-1,\ m-1-r,\ m-1-r-(r-\lambda))\right]=\\&=\left[\frac{m}{m-r}R(m-1,\ r,\ \lambda)\right].\end{aligned}$$

Это соотношение применимо в разнообразных случаях. В частности, мы можем показать, что для $n=6u-1$, $u\geqslant 1$,

$$R(n,\ 3,\ 1)<\left[\frac{n}{3}\left[\frac{n-1}{2}\right]\right].$$

[1]) В настоящее время этот результат уже опубликован в статье: S. M. Jonson, Improved asymptotic bounds for error-correcting codes, *IEEE Trans. Inf. Theory*, **IT-9** (1963), № 3.— *Прим. перев.*

В то время как для $n=6u+1$ или $6u+3$ при $u \geqslant 1$

$$R(n,\ 3,\ 1)=\left[\frac{n}{3}\left[\frac{n-1}{2}\right]\right],$$

так как в этих случаях существуют тройки Штейнера (см. [4]).

Для того чтобы показать, что $R(12,\ 5,\ 2)$ есть только 12, а не 14, необходимы специальные комбинаторные рассуждения.

Мы можем также показать, что $R(13,\ 5,\ 2)=18$; $A(13,\ 5) \leqslant 69$; $R(14,\ 5,\ 2) \leqslant 27$; $A(14,\ 5) \leqslant 127$; $R(15,\ 5,\ 2) \leqslant 40$; $A(15,\ 5) \leqslant 248$.

В теореме 1 член $[n/(e+1)]$ можно заменить выражением

$$1+\frac{\binom{d+1}{e+1} R(n,\ d+1,\ e+1)}{\binom{n}{e+1}-\binom{d}{e} R(n,\ d,\ e)},$$

всякий раз, когда оно оказывается меньше.

Таким образом, например, $A(16,\ 7) \leqslant 62$.

Соответствующим обобщенным выражением можно заменить члены $R(n,\ r_1,\ r_1-e-1)$ в теореме 4, в результате чего, например, $A(100,\ 41)$ составляет $^1/_{190}$ значения оценки Хэмминга. Также можно показать, что в асимптотическом случае любой код, исправляющий e ошибок, будет исправлять почти все последовательности с $e+t$ ошибками для $t<g(F)$, где $g(F)$ описана в этой статье выше.

Подробности будут представлены позже.

ЛИТЕРАТУРА

1. Hamming R. W., Error detecting and error-correcting codes, *Bell. Sys. Tech. J.*, **29**, April (1950), 147—160. [Русский перевод: Хэмминг Р., Коды с обнаружением и исправлением ошибок, сб., ИЛ, М., 1956].
2. Wax N., On upper Bounds for error detecting and error-correcting codes of finite length, *IRE Trans. on Inform. Theory*, **IT-5**, December (1959), 168—174.

3. Plotkin M., Binary codes with specified minimum distance, *IRE Trans. on Inform. Theory*, **IT-6**, September (1960), 445—450. [Русский перевод: Плоткин М., Двоичные коды с заданным минимальным расстоянием, Кибернетический сб., вып. 7, ИЛ, М., 1963, 60—70].
4. Ryser H. J., Matrices of zeros and ones, *Bull. Amer. Math. Soc.*, **66**, November (1960), 442—464.
5. Griesmer J. H., A bound for error-correcting codes, *IBM J. Res. and Dev.*, **4**, November (1960), 532—542.
6. Peterson W. W., Error-correcting codes, M.I.T. Press and J. Wiley and Sons, Inc., New York, N. Y., 1961. [Русский перевод: Питерсон У., Коды, исправляющие ошибки, ИЛ, М., 1964].

КОДЫ, ИСПРАВЛЯЮЩИЕ ОШИБКИ, И ИХ ПРИМЕНЕНИЕ [1] В СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Дж. Меггит

Описываемая здесь практически автоматическая система для исправления ошибок может быть использована во многих случаях передачи информации. В частности, она пригодна для исправления пакетов ошибок и, следовательно, может быть применена при передаче данных по телефонным сетям.

Привлекательной чертой системы является особая простота ее выполнения. Она настолько проста, что легко может быть собрана из большинства уже существующих готовых узлов.

Сообщения в этой системе передаются блоками, а каждый блок кодируется отдельно. Коды, которые применяются для кодирования блоков, являются циклическими. Это означает, что кодирующее и декодирующее устройства содержат регистры сдвига с линейной обратной связью, которая используется и для образования проверочных символов, и для исправления ошибок.

Основные идеи изложены в технических терминах, а затем уже в математических, так что они легко могут быть поняты теми, кто их будет применять.

Обычно теория применяется к двоичным сообщениям, передаваемым последовательно. В работе сделано обобщение, показывающее, как те же самые идеи могут быть применены к двоичным кодам, если информация поступает параллельно.

Введение

Задача автоматического исправления ошибок является центральной в теории передачи сообщений. Большинство существующих систем связи подвергается воздействию шума, и поэтому возникает альтернатива между дорогим ее улучшением и применением системы для исправления ошибок, если необходимо передавать информацию точно. Становится все более и более ясным, что вычислительные

---

[1] Meggit J. E., Error-correcting codes and their implementation for data transmission systems, *IRE Transactions on Information Theory*, **IT-7** (1961), № 4, 234—244.

машины должны быть связаны между собой каналами связи, и поэтому эта задача приобретает особое значение.

Современная телефония обеспечивает широковещательную связь, и было бы очень желательно приспособить ее для этих новых целей. Однако она сильно шумящая в значительной степени благодаря импульсному шуму, возникающему на телефонных станциях, а любая исправляющая ошибки система, способная устранить эти помехи, была бы все же достаточно громоздкой. В этой новой области передачи информации основной целью является наиболее быстрая передача данных, так как чем больше информации может быть за секунду передано по каналу, тем дешевле сама передача. Таким образом, очевидно, следует увеличивать скорость передачи реальной системы до тех пор, пока начнут появляться ошибки и разумным будет дальнейшее автоматическое исправление этих ошибок; такая система является дешевой и, в частности, пригодна для исправления пакетов ошибок. Она будет описана ниже. Сама система достаточно гибкая с точки зрения использования кода для согласования его с характером ожидаемого шума, создаваемого различными телефонными каналами, и, вероятно, скоро можно будет сделать точные рекомендации относительно кода, который следует применять в любом конкретном случае.

Эти замечания относятся к телефонным каналам. Те же системы для исправления ошибок найдут применение в системах радиосвязи и в магнитной записи. В последнем случае данные могут быть записаны значительно более плотно, если ошибки, вызываемые незначительным несовершенством магнитных материалов, могут быть автоматически исправлены.

Содержание. В следующем разделе будет описана аппаратура. Затем будут приведены примеры, показывающие, как применяется система, и далее — некоторые обобщения теории.

Коды. Далее будет предполагаться, что данные передаются в двоичной форме, т. е. сообщения состоят из нулей и единиц. В первой части предполагается, что они передаются последовательно, один символ в единицу времени. В последней части статьи приводится некоторое

обобщение для случая, когда посылается по несколько символов в единицу времени.

Сама процедура кодирования заключается в разделении сообщения на отдельные блоки и добавлении к информационным символам в каждом блоке нескольких проверочных символов, которые зависят от информационных. Избыточность такова, что при появлении ошибок остается еще достаточно данных для правильного приема сообщения. Проблема кодирования и заключается в создании этой избыточности простым способом, а проблема декодирования состоит в восстановлении верного сообщения.

Проверочные символы являются линейными функциями от информационных символов. Это, конечно, не необходимо, но значительно упрощает дело. Задача построения кода состоит в таком выборе проверочных символов, чтобы сам процесс кодирования был простым и чтобы исправлялся ожидаемый вид ошибок.

Циклические коды. Коды, рассматриваемые здесь, представляют дальнейшее ограничение класса систематических кодов и называются циклическими кодами. Они представляют интерес с точки зрения простоты реализации. Было установлено, что они обладают рядом интересных свойств. Абрамсон [1], Мелас [2] и Файр [3], например, установили некоторые из этих свойств, не отметив, правда, при этом присущую этим кодам простоту.

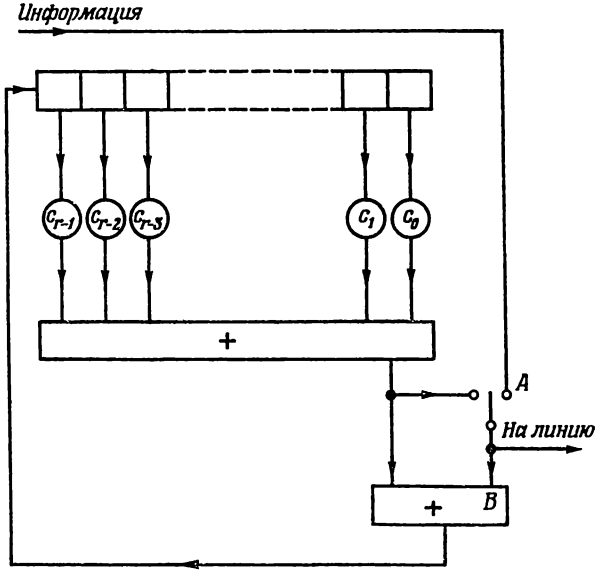
Циклические коды могут быть определены математически, а для инженеров их можно определить с помощью тех средств, которые нужны для их реализации. Основой всей системы является регистр сдвига с обратной связью, а его циклические свойства привели к названию *циклический код.*

В статье будет принят инженерный подход и сначала будет описана аппаратура. Математика же введена только для анализа ее свойств.

Общие свойства кодирующего устройства для циклического кода

При передаче блок символов упорядочивается так, что сначала передаются информационные символы, а затем проверочные. Пусть общее число символов равно n, а число проверочных символов — r.

Кодирующее устройство изображено на рис. 1 и состоит главным образом из регистра сдвига длины r с обратной связью. Связи в регистре, которые могут или быть, или отсутствовать, обозначаются символом c. Их выбор остается за проектировщиком. Удобно считать $c=1$, если связь существует и $c=0$, когда она отсутствует; $c_0=1$, потому что в противном случае можно было бы обойтись меньшим регистром.



Р и с. 1. Кодирующее устройство.

Сумматоры образуют сумму по модулю два из входных значений. Набор значений c определяет свойства кода, и когда будет рассматриваться пример для конкретного кода, будут приведены конкретные соединения.

Работа кодирующего устройства заключается в том, что сначала ключ A замкнут и $n-r$ информационных символов непосредственно передаются так, как они поступают, и одновременно направляются в регистр сдвига, который вначале содержит только нули. Когда вся информация передана, ключ A переключается так, что

вход кодирующего устройства оказывается изолированным, тогда как выход его подключен на передачу. В этом случае передаются r проверочных символов. Когда передача информационных символов закончится, на входе регистра сдвига будет нуль, так как сумматор B имеет два одинаковых входа и, стало быть, его выход равен нулю. Таким образом, к концу передачи сдвигающий регистр снова окажется содержащим одни нули.

Число r равно числу состояний регистра; n таково, что если регистр сдвига вначале имел $100\ldots0$, то при включенной обратной связи и в отсутствии ключа A он будет иметь снова $100\ldots0$ точно после n сдвигов, но не ранее.

Теперь необходимо провести анализ кода, определяемого кодирующим устройством. Если обозначить символы сообщения как $a_1, a_2, \ldots, a_n$ (a_1 вначале), то очевидно, что $a_{n-r+1}, \ldots, a_n$ определяются по $a_1, a_2, \ldots, a_{n-r}$, и ниже можно будет записать это соотношение. Чтобы это сделать, необходимо описать операции сдвига в регистре математически, а для этого удобно обозначить его содержимое в любой момент времени вектором **y**, а его содержимое после сдвига при помощи **Ty**, где **T**— квадратная матрица порядка r;

$$\mathbf{T}=\begin{Vmatrix} c_{r-1} & c_{r-2} & \cdot & \cdot & \cdot & c_1 & c_0 \\ 1 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 1 & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & 1 & 0 \end{Vmatrix}. \qquad (1)$$

(Первый элемент столбца вектора **y** соответствует значению левого элемента регистра сдвига.)

При помощи такого обозначения может быть описано содержимое всего регистра сдвига. Сначала он содержит нули. В следующий момент он содержит $a_1\mathbf{x}$, где **x** — вектор

$$\mathbf{x}=\begin{Vmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{Vmatrix}. \qquad (2)$$

Когда поступает второй символ a_2, регистр содержит

$$a_1\mathbf{T}\mathbf{x}+a_2\mathbf{x}.$$

Когда появляется третий символ, регистр содержит

$$a_1\mathbf{T}^2\mathbf{x}+a_2\mathbf{T}\mathbf{x}+a_3\mathbf{x}, \quad \text{и т. д.}$$

и это продолжается до тех пор, пока не поступят все информационные символы. Но даже если формируются и проверочные символы $a_{n-r+1}, \ldots, a_n$, они поступают обратно в регистр тем же самым способом. Таким образом, регистр в конечном счете содержит

$$a_1\mathbf{T}^{n-1}\mathbf{x}+a_2\mathbf{T}^{n-2}\mathbf{x}+\ldots+a_{n-1}\mathbf{T}\mathbf{x}+a_n\mathbf{x}; \qquad (3)$$

n выбирается так, чтобы $\mathbf{T}^n\mathbf{x}=\mathbf{x}$. Далее, из принципа работы устройства следует, что после последних r сдвигов, таких, когда поступают только нули, регистр будет содержать одни нули. Отсюда

$$a_1\mathbf{T}^{-1}\mathbf{x}+a_2\mathbf{T}^{-2}\mathbf{x}+\ldots+a_{n-1}\mathbf{T}^{-(n-1)}\mathbf{x}+a_n\mathbf{T}^{-n}\mathbf{x}=0, \qquad (4)$$

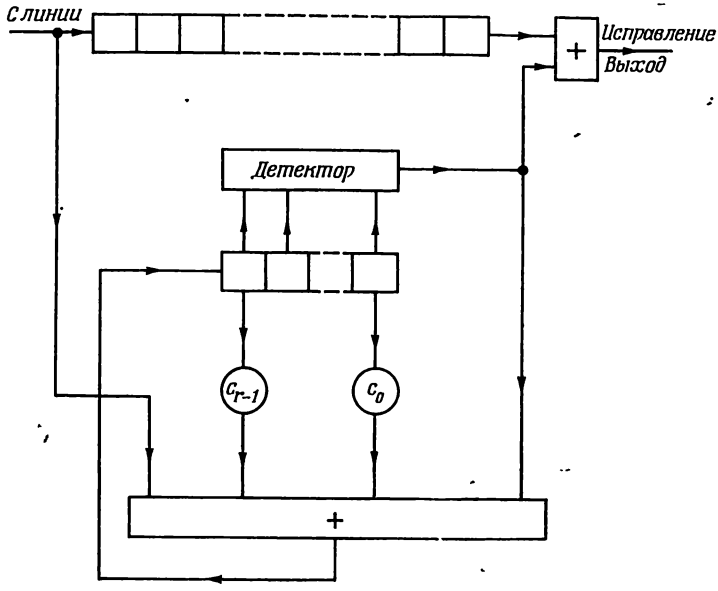
и это есть математическое определение циклического кода. Эта форма потребуется далее для анализа ошибок, которые код может исправить. Она представляет собой систему линейных уравнений, которая определяет проверочные разряды через информационные.

Общие свойства декодирующего устройства для циклического кода

Декодирующее устройство для циклического кода изображено на рис. 2. Оно содержит главным образом два регистра сдвига, один из них длины n, где сообщение запоминается до тех пор, пока не будут приняты проверочные символы, так как до этого неясно, что следует исправлять. Второй регистр сдвига длины r содержит обратную связь и имеет те же самые соединения, что и регистр сдвига кодирующего устройства.

Декодирующее устройство содержит также детектор. Его функции заключаются в выявлении определенных значений второго регистра сдвига. Если хотя бы одно из таких значений определено, образуется единица: она

изменяет символ, покидающий главный регистр сдвига, и добавляется в сумматор второго регистра. Значения, которые выявляются детектором, зависят от ошибок, которые код должен исправить; как это делается, будет описано ниже.



Р и с. 2. Декодирующее устройство.

Работа декодирующего устройства состоит в следующем. Как только сообщение получено, оно запоминается в главном регистре сдвига, причем одновременно принимаемые символы поступают во второй регистр сдвига. В течение этого времени детектор отключен. Удобно обозначить поступающие символы через $a_1', a_2', \ldots, a_n'$, которые, конечно, могут отличаться и от $a_1, a_2, \ldots, a_n$.

Второй регистр сдвига вначале содержит нули, так что если первый символ a_1' получен, он продвигается здесь точно так же, как в кодирующем устройстве: $a_1'\mathbf{x}$. В следующий момент регистр содержит $a_1'\mathbf{T}\mathbf{x} + a_2'\mathbf{x}$ и т. д., так что если сообщение полностью принято,

регистр содержит

$$a_1'\mathbf{T}^{-1}\mathbf{x}+a_2'\mathbf{T}^{-2}\mathbf{x}+\ldots+a_n'\mathbf{T}^{-n}\mathbf{x}=\mathbf{z}. \qquad (5)$$

Если не было ошибок и все a_i' те же самые, что и a_i, то ясно, что $\mathbf{z}=0$, в общем же случае $\mathbf{z}\neq 0$. По определению $\mathbf{z}$ должно иметь разное значение для разных ошибок, которые необходимо исправлять. На этом этапе аппаратура уже может быть использована для обнаружения ошибок. Все ошибки, которые приводят к ненулевому $\mathbf{z}$, будут обнаружены.

Если же устройство используется для исправления ошибок, процесс на этом не заканчивается, а включается детектор, причем вход декодирующего устройства отключается от сумматора регистра сдвига. Это происходит тогда, когда $\mathbf{z}$ уже образовано, а сдвиг все еще продолжается. Поскольку сдвиг продолжается, содержимое нижнего регистра управляется с помощью $\mathbf{T}$, в то время как символы принятого сообщения начинают покидать запоминающий регистр. Мы выберем детектор таким образом, чтобы он выявлял необходимую комбинацию, которая появляется во втором регистре всякий раз, когда ошибочный символ достигает правого конца запоминающего регистра. При следующем сдвиге детектор образует единицу, которая исправляет ошибку и в то же время эта единица добавляется по цепи обратной связи ко второму регистру, для обозначения того, что осталась некоторая более простая комбинация в регистре, которая должна быть еще исправлена. Процесс продолжается до тех пор, пока входное сообщение полностью не покинет главный запоминающий регистр.

Следует заметить, что новое сообщение не может быть принято до тех пор, пока предыдущее исправляется. Если же символы поступают непрерывно, то необходимо еще предусмотреть некоторое дополнительное буферное устройство.

Детектор

Состояние детектора зависит от кода, который используется для исправления. Самый простой детектор применяется в том случае, когда требуется исправить единичную ошибку.

Детектор для единичной ошибки

Если, скажем, появилась ошибка в s-й позиции, то

$$a_s' = a_s + 1,$$

в то время как остальные a_i' равны соответствующим a_i.

Отсюда из (5)

$$\mathbf{z} = \mathbf{T}^{-s}\mathbf{x}. \tag{6}$$

Теперь s-й символ покинет главный запоминающий регистр после $s-1$ сдвигов. К этому времени состояние второго регистра будет

$$\mathbf{T}^{s-1}(\mathbf{T}^{-s}\mathbf{x}) = \mathbf{T}^{-1}(\mathbf{x}).$$

Таким образом необходимо просто определить состояние $\mathbf{T}^{-1}\mathbf{x}$, и, если это сделано, любая единичная ошибка будет исправлена. Так как

$$\mathbf{x} = \begin{Vmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{Vmatrix} \quad \text{и} \quad c_0 = 1,$$

то

$$\mathbf{T}^{-1}\mathbf{x} = \begin{Vmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{Vmatrix} \tag{7}$$

для любого $\mathbf{T}$. Это как раз и означает, что необходимо заметить тот момент, когда последний элемент регистра содержит единицу, а все другие элементы нули. Это и отмечает детектор.

Когда детектор работает, он посылает единицу в сумматор регистра сдвига, что изменяет его состояние y в

$$y + T^{-1}x.$$

В этом случае $y = T^{-1}x$, так что после следующего сдвига он содержит

$$T(T^{-1}x + T^{-1}x) = 0.$$

Содержимое регистра сдвига соответствует теперь неискаженному сообщению и дальнейшее исправление не производится.

Детектор для двойной смежной ошибки

Если код предназначен для исправления единичной и двойной смежной ошибок, то детектор должен определять и другое состояние, кроме $T^{-1}x$.

Предположим, что имеются ошибки в s-й и $(s+1)$-й позициях. Тогда, так же как и раньше,

$$z = T^{-s}(1 + T^{-1})x; \qquad (8)$$

s-й символ покидает главный запоминающий регистр после $s-1$ сдвигов; к этому времени второй регистр содержит

$$(T^{-1} + T^{-2})x.$$

Поэтому, приспособив детектор для определения $(T^{-1} + T^{-2})x$, можно исправить s-й символ. Так как он осуществляет добавление $T^{-1}x$ ко второму регистру, когда имеет место первое исправление, второй регистр будет содержать после следующего сдвига

$$T[(T^{-1} + T^{-2})x + T^{-1}x] = T^{-1}x.$$

Это состояние теперь фиксируется детектором как единичная ошибка; таким образом исправляется $(s+1)$-й символ и регистр становится пустым после следующего сдвига.

Состояние $(T^{-1} + T^{-2})x$ имеет форму, которая зависит от T, и может быть вычислено, если известно T. Это та форма детектора, при которой исправляется единичная и двойная смежная ошибка.

Детектор для пакетов ошибок длины $l (l < r)$

Изложенная выше теория обобщается очевидным образом на тот случай, когда исправляются пакеты ошибок вплоть до длины l. Это означает, что если появляются ошибки, то они распространяются на l или менее последовательных символов, хотя и не обязательно все l символов неверные. Предполагается, что применяемый код может осуществить исправление таких ошибок, и задача состоит в том, чтобы показать, как будет выглядеть детектор для этого кода.

Если продолжить рассуждения последнего раздела, можно найти, что для пакета ошибок детектор должен генерировать 2^{l-1} состояний, соответствующих 2^{l-1} различным значениям пакетов. Эти состояния запишутся в форме

$$\mathbf{z} = (\mathbf{T}^{-1} + \sum_{i=2}^{l} q_i \mathbf{T}^{-i})\,\mathbf{x}, \qquad (9)$$

где все q принимают значения нуль или единица. Такой детектор вообще можно построить, хотя, по-видимому, это и приведет к некоторым трудностям. Тем не менее имеется почти тривиальное решение, которое определяет все состояния в форме (9). Оно показано на рис. 3, который изображает основной регистр сдвига с обратной связью декодирующего устройства и с соответствующим образом приспособленным детектором. Он работает тогда, когда первые $r-l$ символов регистра сдвига равны нулю, а следующие l символов таковы, что образуют на выходе цепи обратной связи (сумматора) единицу. Утверждается, что устройство определяет все состояния, представленные в форме (9).

Доказательство. Из теории следует, что

$$\mathbf{T}^{-1}\mathbf{x} = \begin{Vmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \\ 1 \end{Vmatrix}; \quad \mathbf{T}^{-2}\mathbf{x} = \begin{Vmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ t_1 \end{Vmatrix}; \quad \mathbf{T}^{-3}\mathbf{x} = \begin{Vmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ t_1 \\ t_2 \end{Vmatrix} \text{ и т. д., } (10)$$

где, оперируя с T, получаем, что t_1, t_2, t_3 и т. д. определяются из системы

$$\begin{aligned} c_1 + t_1 &= 0, \\ c_2 + c_1 t_1 + t_2 &= 0, \\ c_3 + c_2 t_1 + c_1 t_2 + t_3 &= 0. \end{aligned} \tag{11}$$

Таким образом, если регистр сдвига содержит $\mathbf{T}^{-1}\mathbf{x}$, на выходе сумматора появится единица; когда регистр

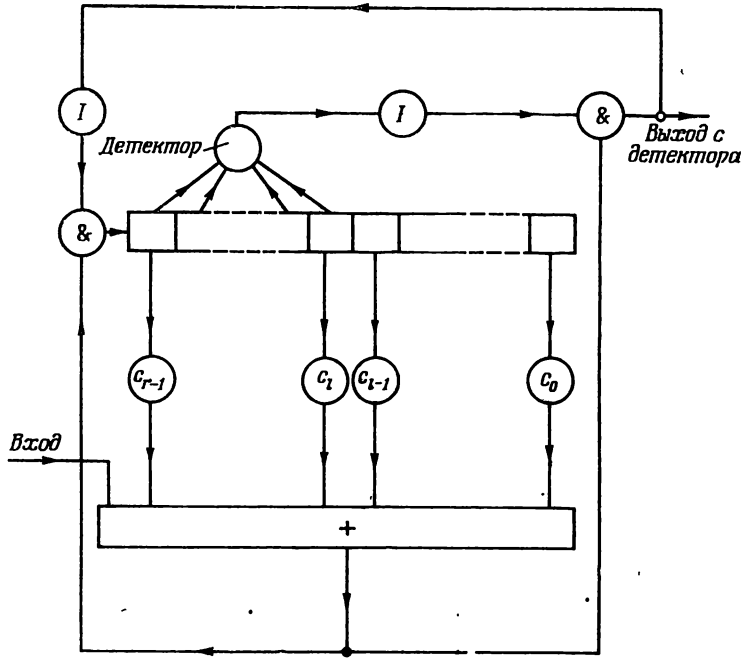


Рис. 3.

содержит $\mathbf{T}^{-2}\mathbf{x}$, выходом сумматора является $c_1 + t_1 = 0$, когда в регистре сдвига $\mathbf{T}^{-3}\mathbf{x}$, на выходе сумматора получается $c_2 + c_1 t_1 + t_2$ и т. д.

Однако, когда регистр сдвига содержит $\mathbf{T}^{-1}\mathbf{x}$ и линейную комбинацию $\mathbf{T}^{-i}\mathbf{x}$ ($i = 2, \ldots, l$), выход сумматора (в цепи обратной связи) равен точно единице и, конечно, первые $r - l$ элементов равны нулю. Когда первые $r - l$

элементов равны нулю, выход сумматора равен нулю для 2^{l-1} состояний и единице для 2^{l-1} состояний. Поэтому описанное только что устройство определяет точно те состояния, которые обозначены в (9) и никакие другие, чем и завершается доказательство.

В ранее описанной аппаратуре детектор направляет единицу в сумматор регистра сдвига с обратной связью только тогда, когда он работает. В новой упрощенной аппаратуре это может быть сделано отдельным сумматором с двумя входами в петле обратной связи. Следует заметить, что эффект суммирования эквивалентен появлению нуля в обратной связи регистра сдвига. Следовательно, тот же самый эффект может быть достигнут разрывом петли обратной связи, что и показано на рис. 3.

Особо следует заметить, что теория такого детектора имеет смысл лишь пока используемый код в принципе может исправлять пакеты ошибок вплоть до длины l. Если используют, например, детектор для $l=2$ и код, который предназначен для исправления единичной ошибки, то может случиться, что возникнет неопределенность. Единичные ошибки на определенных позициях будут обнаружены и ошибочно исправлены как двойные смежные ошибки на других позициях.

Следует отметить также, что $l \ll r$, потому что $l-1$ символов из r проверочных описывают характер пакета ошибок, а остальные $r-l+1$ определяют положение пакета.

Детектор для общего вида циклического кода

В общем случае циклический код может исправлять самые разнообразные виды наборов шумовых последовательностей. Если код такой, что исправляется ошибка вида $1\ q_1, q_2 \ldots$, то точно так же, как и в (9), детектор в декодирующем устройстве должен быть таким, чтобы он определял состояния

$$z = (\mathbf{T}^{-1} + \sum_{i=2} q_i \mathbf{T}^{-i})\, x. \qquad (12)$$

Далее везде будет предполагаться, что код исправляет, а детектор выявляет и все более простые, но возможные виды ошибок.

Завершение описания аппаратуры

Этим завершается описание аппаратуры. Главной задачей, которую необходимо еще обсудить, являются соединения, которые необходимо осуществить в регистрах сдвига, чтобы получился код с предписанными свойствами. Следующий раздел содержит ряд правил для построения некоторых очень эффективных кодов. Полное же исследование возможных соединений может привести к некоторым другим полезным кодам, а это уже заслуживает внимания. Было бы весьма желательным применение вычислительной техники для исчерпывающего исследования свойств всех кодов, которые могут быть созданы такой аппаратурой.

Коды для исправления единичной ошибки

Сначала будут рассмотрены соединения, которые требуются для образования кода, исправляющего единичную ошибку.

Как было отмечено, единичная ошибка в s-й позиции приводит к $\mathbf{z} = \mathbf{T}^{-s}\mathbf{x}$.

Поэтому код будет способен исправлять единичную ошибку, если все векторы $\mathbf{T}^{-s}\mathbf{x}$ различны для различных s, и это действительно учитывается при построении детектора. Наиболее эффективный код может быть получен, если регистр сдвига соединен так, чтобы в нем образовался максимально длинный цикл длины $2^r - 1$. Циклическая структура (4) регистров сдвига в общем случае исследуется с помощью характеристического уравнения, которому удовлетворяет матрица $\mathbf{T}$, и из (1) видно, что это уравнение имеет вид

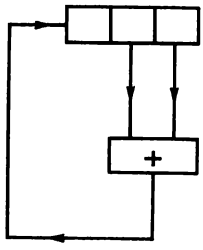


Рис. 4.

$$\mathbf{T}^r + c_{r-1}\mathbf{T}^{r-1} + c_{r-2}\mathbf{T}^{r-2} + \ldots + c_1\mathbf{T} + \mathbf{1} = \mathbf{0}. \qquad (13)$$

Характеристические уравнения, которые образуют максимально длинные циклы, изучены, и имеются соответствующие таблицы. Они могут быть использованы для кодов, исправляющих единичную ошибку.

Пример. Код для сообщения длины 7 с тремя проверочными символами получается при использовании характеристического уравнения

$$\mathbf{T}^3+\mathbf{T}+\mathbf{1}=\mathbf{0}, \tag{14}$$

которое очень просто выписать из таблиц. Соединения основного регистра изображены на рис. 4.

Из (4) найдем, что, уравнения кодирования имеют вид

$$a_{6+i}+a_{3+i}+a_{2+i}+a_{1+i}=0 \text{ для } i=0,\ 1,\ 2.$$

Коды для исправления двойной смежной ошибки

Теперь будут рассмотрены соединения для кодов, исправляющих двойные смежные ошибки. Как видно из (8), требуется, чтобы векторы

$$\mathbf{T}^{-s}\mathbf{x} \text{ и } \mathbf{T}^{-t}(\mathbf{1}+\mathbf{T}^{-1})\,\mathbf{x}$$

были все различны для различных величин s и t. Это будет иметь место, если регистр сдвига обратной связи таков, что векторы $\mathbf{x}$ и $(\mathbf{1}+\mathbf{T}^{-1})\,\mathbf{x}$ находятся в различных циклах одинаковой длины.

Наиболее просто этого достигнуть, если взять характеристическое уравнение в форме

$$(\mathbf{1}+\mathbf{T})\,M(r\mathbf{T})=\mathbf{0}, \tag{15}$$

где $M(r\mathbf{T})=\mathbf{0}$ является характеристическим уравнением, которое образует максимально длинный цикл длины 2^r-1.

Доказательство. Рассмотрим матрицу порядка $r+1$

$$\mathbf{T}=\left\|\begin{array}{c|c}\mathbf{T}_1 & \mathbf{0}\\ \hline \mathbf{0} & \mathbf{1}\end{array}\right\|, \tag{16}$$

где $\mathbf{T}_1$ — матрица, удовлетворяющая условию

$$M(r\mathbf{T}_1)=\mathbf{0}. \tag{17}$$

Тогда ясно, что характеристическое уравнение, которому удовлетворяет $\mathbf{T}$, есть точно (15), хотя по-новому определенное $\mathbf{T}$ имеет несколько отличную форму. Циклическая структура определяется характеристическим

уравнением (за исключением особых случаев), которому удовлетворяет T.

Имеется цикл длины 2^r-1 с векторами в форме

$$\left\| \frac{\mathbf{T}_1^{-s}\mathbf{x}_1}{0} \right\|, \quad s=1, \ \ldots, \ 2^r-1,$$

где $\mathbf{x}_1$ — вектор-столбец порядка r

$$\left\| \begin{matrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right\|;$$

имеется другой цикл длины 2^r-1 с векторами в форме

$$\left\| \frac{\mathbf{T}_1^{-s}\mathbf{x}_1}{1} \right\|, \quad s=1, \ \ldots, \ 2^r-1.$$

Теперь

$$\left\| \frac{\mathbf{x}_1}{1} \right\| + \left\| \frac{\mathbf{T}_1^{-1}\mathbf{x}_1}{1} \right\| = \left\| \frac{(1+\mathbf{T}_1^{-1})\,\mathbf{x}_1}{0} \right\|,$$

так что сумма двух последовательных векторов в этом цикле дает вектор другого цикла, а это и есть требуемая нами структура.

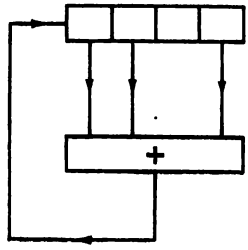


Рис. 5.

Пример. Код для сообщения длины 7 с 4 проверочными символами может быть получен, если взять

$$M(r\mathbf{T}) = \mathbf{T}^3 + \mathbf{T} + 1, \qquad (18)$$

так что этот код описывается характеристическим уравнением

$$\begin{aligned} &(\mathbf{T}+1)(\mathbf{T}^3+\mathbf{T}+1)=0, \\ &\mathbf{T}^4+\mathbf{T}^3+\mathbf{T}^2+1=0. \end{aligned} \qquad (19)$$

Этот код способен исправлять единичные и двойные смежные ошибки. Основной регистр сдвига изображен на рис. 5.

Уравнения кодирования имеют форму

$$a_{4+i}+a_{3+i}+a_{1+i}=0, \quad i=0, \ 1, \ 2, \ 3.$$

Детектор этого декодирующего устройства определяет состояния

$$\left\|\begin{matrix}0\\0\\0\\1\end{matrix}\right\| \text{ и } \left\|\begin{matrix}0\\0\\1\\1\end{matrix}\right\|.$$

Это эквивалентно определению двух нулей в первых двух элементах регистра и единицы в последнем, что находится в соответствии с правилами для детекторов для исправления пакетов.

Коды Файра [3]

Существует обобщение только что описанного кода, исправляющего двойную смежную ошибку. Оно получается из характеристического уравнения

$$(\mathbf{T}^l + \mathbf{1})\, M(r\mathbf{T}) = 0, \tag{20}$$

где l взаимно просто с $2^r - 1$. Это приводит к кодам длины $l(2^r - 1)$ с $l + r$ проверочными символами. Основное достоинство этой формы то, что результирующая циклическая структура легко анализируется. Для этого рассмотрим квадратную матрицу порядка $l + r$, определяемую следующим образом:

$$\mathbf{T} = \left\|\begin{array}{c|c} \mathbf{T}_1 & 0 \\ \hline 0 & \mathbf{T}_2 \end{array}\right\|, \tag{21}$$

где $\mathbf{T}_1$ удовлетворяет уравнению $M(r\mathbf{T}_1) = 0$ и $\mathbf{T}_2$ — квадратная матрица порядка l

$$\mathbf{T}_2 = \left\|\begin{matrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 1 & 0 \end{matrix}\right\|. \tag{22}$$

Как и ранее, эти $\mathbf{T}$ отличаются от обычно рассматриваемых матриц, однако они удовлетворяют тем же самым характеристическим уравнениям и, отсюда, имеют ту же самую циклическую структуру.

Основной цикл длины $l(2^r-1)$ может быть получен, если рассмотреть векторы

$$\left\| \frac{\mathbf{T}_1^{-s}\mathbf{x}_1}{\mathbf{T}_2^{-s}\mathbf{x}_2} \right\|,$$

где $\mathbf{x}_1$ — вектор-столбец порядка r

$$\left\| \begin{matrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right\|$$

и $\mathbf{x}_2$ — вектор-столбец порядка l

$$\left\| \begin{matrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right\|.$$

Тогда вектор

$$\left(1+\sum_{i=1} q_i \mathbf{T}^{-i}\right) \left\| \frac{\mathbf{x}_1}{\mathbf{x}_2} \right\|,$$

который является линейной комбинацией векторов основного цикла, будет иметь вид

$$\left\| \begin{matrix} \left(1+\sum\limits_{i=1} q_i \mathbf{T}^{-i}\right)\mathbf{x}_1 \\ \hline 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ q_3 \\ q_2 \\ q_1 \end{matrix} \right\|.$$

Далее, все другие члены цикла, относящиеся к вектору, будут содержать в нижних l позициях как раз циклическую перестановку l величин.

Так, в общем случае векторы

$$\left(1+\sum_{i=1} q_i \mathbf{T}^{-i}\right)\left\|\begin{matrix}\mathbf{x}_1\\ \mathbf{x}_2\end{matrix}\right\|$$

будут получены из различных циклов для различных множеств q_i, каждый цикл будет длины $l(2^r-1)$, а это и есть требуемая структура для кода, который исправляет набор ошибок вида $1\ q_1\ q_2\ q_3\ \ldots$.

Цикл действительно будет иметь длину $l(2^r-1)$ при условии, во-первых, что все q таковы, что требуется l циклических сдвигов, прежде чем комбинация повторится,

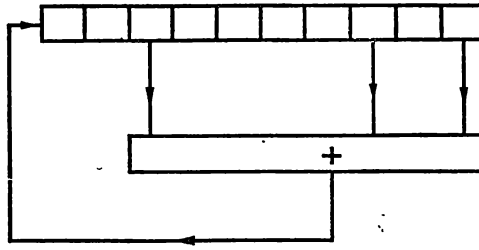


Рис. 6.

и во-вторых, что $\left(1+\sum_{i=1} q_i \mathbf{T}_1^{-i}\right)\mathbf{x}_1 \neq 0$. Последнее условие может быть удовлетворено, если взять характеристическое уравнение достаточно высокой степени относительно $\mathbf{T}_1$.

Пример. Когда $l=5$, могут быть исправлены ошибки 1, 11, 101, 111, 1111 и 1011 или 1101.

Характеристическое уравнение

$$\begin{aligned}&(\mathbf{T}^5+1)(\mathbf{T}^5+\mathbf{T}^2+1)=0,\\ &\mathbf{T}^{10}+\mathbf{T}^7+\mathbf{T}^2+1=0\end{aligned}\qquad(23)$$

дает код длины 155, содержащий 10 проверочных символов, способных исправлять приведенные выше типы ошибок.

Соединения показаны на рис. 6, и детектор определяет шесть возможных наборов ошибок. Эти наборы

ошибок могут быть легко вычислены и имеют вид

0 0 0 0 0 0 0 0 0 1 для образца вида 1,
0 0 0 0 0 0 0 0 1 1 для образца вида 11,
0 0 0 0 0 0 0 1 0 0 для образца вида 101,
0 0 0 0 0 0 0 1 1 0 для образца вида 111,
0 0 0 0 0 0 1 1 0 0 для образца вида 1111
и 0 0 0 0 0 0 1 0 0 1 для образца вида 1101
или 0 0 0 0 0 0 1 1 1 0 для образца вида 1011.

Очевидно, что детектор для выделения первых четырех наборов ошибок совпадает с детектором для пакетов ошибок длины 3. Другие два набора ошибок должны выделяться отдельно.

Если этот код будет использоваться в практических случаях, то целесообразно использовать отдельно простой детектор для пакета длины 3. Следует все же помнить, что после полного исправления регистр сдвига детектора содержит нули. Отсюда, если сообщение было получено верно или с ошибками типа 1, 11, 101, 111, которые затем исправлены, регистр сдвига будет содержать нули. Поэтому, если в конце этого процесса исправления регистр не содержит одних нулей, это означает, что появились некоторые другие ошибки, и, следовательно, система может использоваться для обнаружения некоторого числа других ошибок.

Коды для исправления общего вида пакетов ошибок

Ясно, что только что описанный код Файра [3] будет исправлять все ошибки внутри пакета длины $(l+1)/2$ (l нечетное) и $l/2$ (l четное) и даже некоторые другие. Это и определяет общий вид кода для исправления пакетов ошибок.

Например, взяв

$$(\mathbf{T}^{11}+1)(\mathbf{T}^{6}+\mathbf{T}^{4}+\mathbf{T}^{3}+\mathbf{T}+1)=0, \qquad (24)$$

можно получить код для сообщения длины 693 с 17 проверочными символами, который способен исправлять

все пакеты ошибок вплоть до длины 6 и даже некоторые другие ошибки.

Следует все же напомнить, что аппаратура для кодирования и декодирования этим кодом, содержащая регистр сдвига длины 17, имеет в декодирующем устройстве еще буферный регистр сдвига длины 693.

Для приведенного примера очень просто применить код для исправления пакетов ошибок, используя упрощенный детектор в декодирующем устройстве, который уже был описан. Оставшуюся избыточность, которую содержит код, можно использовать для обнаружения других ошибок, а это достигается обследованием регистра сдвига в конце процесса исправления — содержит ли он одни нули, или нет.

Коды Боуза — Чоудхури

Боуз и Чоудхури [5] разработали теорию циклических кодов для общего случая исправления многократных ошибок. Из теории следует, что если взять характеристическое уравнение в форме

$$M(r\mathbf{T})\,N(\mathbf{T}) = 0, \tag{25}$$

где $N(\mathbf{S}) = 0$ — характеристическое уравнение, удовлетворяющееся при $\mathbf{S} = \mathbf{T}^3$, и $\mathbf{T}$ удовлетворяет уравнению $M(r\mathbf{T}) = 0$, то получаемый код способен исправлять все двойные ошибки.

Например, если

$$M(r\mathbf{T}) = \mathbf{T}^4 + \mathbf{T}^3 + 1, \tag{26}$$

то из этого следует, что

$$\mathbf{T}^{12} + \mathbf{T}^9 + \mathbf{T}^6 + \mathbf{T}^3 + 1 = 0.$$

Таким образом, равенство $\mathbf{S} = \mathbf{T}^3$ удовлетворяется и

$$\mathbf{S}^4 + \mathbf{S}^3 + \mathbf{S}^2 + \mathbf{S} + 1 = 0. \tag{27}$$

Следовательно, характеристическое уравнение

$$\begin{aligned}&(\mathbf{T}^4 + \mathbf{T}^3 + 1)(\mathbf{T}^4 + \mathbf{T}^3 + \mathbf{T}^2 + \mathbf{T} + 1) = 0,\\ &\mathbf{T}^8 + \mathbf{T}^4 + \mathbf{T}^2 + \mathbf{T} + 1 = 0\end{aligned} \tag{28}$$

в соответствии с теорией Боуза — Чоудхури дает код для

сообщения длиной 15 с 8 проверочными символами, который способен исправлять все двойные ошибки.

На рис. 7 изображены соединения в регистре сдвига. Ясно, что для этого кода имеем

$$a_{8+i}+a_{4+i}+a_{2+i}+a_{1+i}=0, \quad i=0, \ldots, 7.$$

Детектор в декодирующем устройстве должен, конечно, обнаруживать все 14 состояний вида

$$\mathbf{T}^{-1}\mathbf{x}+\mathbf{T}^{-i}\mathbf{x}, \quad i=2, \ldots, 15.$$

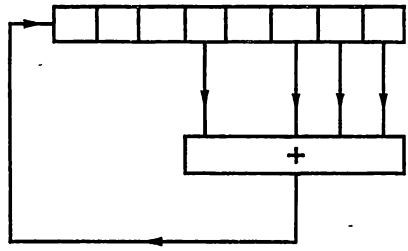


Рис. 7.

Та же самая теория показывает, как получить код для исправления тройных ошибок, если характеристическое уравнение взять в форме

$$M(r\mathbf{T})\,N(\mathbf{T})\,P(\mathbf{T})=0, \tag{29}$$

где условие $P(\mathbf{S})=0$ удовлетворяется при $\mathbf{S}=\mathbf{T}^5$.

Так, в предыдущем примере

$$P(\mathbf{S})=\mathbf{S}^2+\mathbf{S}+1, \tag{30}$$

и видно, что этим способом может быть построен код, исправляющий тройные ошибки длины 15 символов, 10 из которых проверочные.

К сожалению, число различных наборов, которые необходимо различать, остается все же очень большим ($\sim 15^2$), так что детектор теряет свою простоту. Здесь пригодны уже другие схемы, однако рассмотрение этого вопроса выходит за рамки статьи.

Обобщение кодов, исправляющих двойные ошибки

В изложенной выше теории показано, как построить код, исправляющий двойные смежные ошибки для сообщения длины 2^l-1, когда имеется не более чем $2l$ проверочных символов. В действительности можно построить код для исправления двойных смежных ошибок длины 2^l+1, используя $2l$ проверочных символов, при условии, что 2^l+1 не делится на 3.

Необходимые результаты можно легко выписать. Может быть показано, что характеристическое уравнение

$$\mathbf{T}^2+a\mathbf{T}+\mathbf{1}=0, \tag{31}$$

где a — элемент, принадлежащий $GF(2^l)$, дает циклическую структуру, которая сдержит 2^l-1 циклов каждый длины 2^l+1. Эта теория схожа с ранее изложенной, однако векторы и матрицы содержат элементы, принадлежащие $GF(2^l)$.

Далее, если 2^l+1 не делится на 3, то может быть доказано, что все векторы

$$\mathbf{x}, \quad (\mathbf{1}+\mathbf{T}^{-i})\,\mathbf{x}$$

лежат в различных циклах для $i=1, 2, \ldots, 2^{l-1}$. Поэтому циклическая структура пригодна для построения кода, исправляющего двойную ошибку.

$\mathbf{T}$ можно трактовать, как матрицу порядка 2 из элементов $GF(2^l)$ или же можно осуществить матричное представление для элементов поля, при котором $\mathbf{T}$ понимается как матрица порядка $2l$, элементами которой являются нули и единицы. Эта матрица должна иметь ту же самую циклическую структуру, поэтому требуется выписать характеристическое уравнение для этой матрицы порядка $2l$. Оно может быть получено, если использовать уравнение (31), которому удовлетворяет a.

Простой пример разъяснит эти соображения. Пусть a определяется из уравнения

$$a^4+a+1=0, \tag{32}$$

так что a принадлежит $GF(2^4)$. Тогда, так как в силу (31) $a = \mathbf{T}^{-1} + \mathbf{T}$, подставляем

$$(\mathbf{T}^{-1} + \mathbf{T})^4 + (\mathbf{T}^{-1} + \mathbf{T}) + \mathbf{1} = 0$$

и

$$\mathbf{T}^8 + \mathbf{T}^5 + \mathbf{T}^4 + \mathbf{T}^3 + \mathbf{1} = 0. \qquad (33)$$

Таким образом, код, имеющий соединения в регистре. изображенном на рис. 8, дает сообщение длины 17, причем 8 символов из них проверочные, и способен исправлять все двойные ошибки.

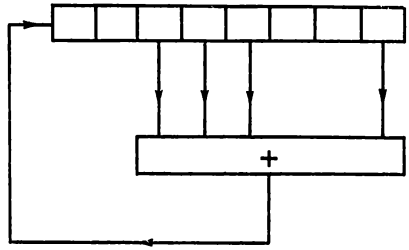


Рис. 8.

Ясно, что уравнениями кодирования будут

$$a_{10+i} + a_{7+i} + a_{6+i} + a_{5+i} + a_{4+i} + a_{1+i} = 0 \qquad (34)$$

для $i = 0, 1, \ldots, 7$.

Этот код, как видно, несколько превосходит по эффективности соответствующий код Боуза — Чоудхури [5].

Коды, исправляющие единичную ошибку для недвоичных символов

До сих пор теория применялась к двоичным сообщениям, и, как следствие, для описания использовалась арифметика по модулю 2. Конечно, не обязательно, чтобы сообщения всегда имели двоичную форму. В то же время существует простое обобщение теории на тот случай, когда сообщение принимает q значений (q простое). Регистры сдвига содержат элементы с q устойчивыми состояниями, а характеристические уравнения, описывающие код, со-

держат коэффициенты, принадлежащие простому полю. Сначала кажется, что это обобщение имеет только академический интерес, тем не менее существует такое же обобщение, при котором элементы обобщения принимают 2^l значений. Регистры сдвига имеют элементы с 2^l устойчивыми состояниями, а характеристическое уравнение имеет коэффициенты, принадлежащие $GF(2^l)$. Это обобщение уже интересно тем, что такая ситуация соответствует случаю, когда двоичное сообщение передается сериями по l символов в единицу времени. Поэтому полезно изучить, в каких случаях эти обобщенные уравнения образуют заслуживающие внимания коды и, стало быть, в каких случаях усложненные регистры сдвига могут создавать такие коды.

В предыдущей части было уже введено характеристическое уравнение в такой форме. Теперь же оно будет исследоваться более подробно.

Пример. Рассмотрим характеристическое уравнение

$$\mathbf{T}^2 + b\mathbf{T} + \mathbf{1} = \mathbf{0}, \tag{35}$$

где b принадлежит $GF(4)$ и удовлетворяет условию

$$b^2 + b + 1 = 0,$$

так что $b^3 = 1$.

(Если использовать здесь теорию последней части, то получим, что код для исправления двойной ошибки имеет длину 5, 4 же символа из них являются проверочными. Таким образом, имеется только один информационный символ, а код имеет только два сообщения 00000 и 11111. Это тривиальный случай, когда можно исправить двойную ошибку.)

Если, однако, (35) используются в прямом смысле, то $\mathbf{T}$ принимает форму

$$\mathbf{T} = \begin{Vmatrix} b & 1 \\ 1 & 0 \end{Vmatrix},$$

$$\mathbf{x} = \begin{Vmatrix} 1 \\ 0 \end{Vmatrix} \tag{37}$$

и найдем, что $\mathbf{T}^5\mathbf{x} = \mathbf{x}$.

Таким образом, уравнения кодирования (4) приобретают вид

$$a_1\begin{Vmatrix}0\\1\end{Vmatrix}+a_2\begin{Vmatrix}1\\b\end{Vmatrix}+a_3\begin{Vmatrix}b\\b\end{Vmatrix}+a_4\begin{Vmatrix}b\\1\end{Vmatrix}+a_5\begin{Vmatrix}1\\0\end{Vmatrix}=0, \quad (38)$$

длина сообщения равна 5, а число проверочных символов — 2; a, конечно, принимают значения 0, 1, b, b^2.

Основной регистр сдвига для этого кода изображен на рис. 9. Элементы регистра принимают четыре значения 0, 1, b, b^2 и кружок с буквой b обозначает умножение на b.

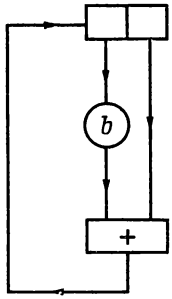


Рис. 9.

Ясно, что циклическая структура регистра содержит 3 цикла каждый длиной 5, один цикл начинается с $\begin{Vmatrix}1\\0\end{Vmatrix}$, второй — с $\begin{Vmatrix}b\\0\end{Vmatrix}$ и третий — с $\begin{Vmatrix}b^2\\0\end{Vmatrix}$.

Следовательно, все 15 векторов $b^i\dot{\mathbf{T}}^s\mathbf{x}$ различны для различных i и s ($i=0, 1, 2, \ldots, s=0, 1, \ldots, 4$). Таким образом, код, определяемый (38), способен исправить единичную ошибку, причем передаваемый символ может иметь четыре значения.

Кодирующее устройство для этого примера. Кодирующее устройство для этого случая имеет точно такую же форму, как и на рис. 1, но содержит регистр, изображенный на рис. 9.

Декодирующее устройство для этого примера. Декодирующее устройство имеет точно такую же форму, как и на рис. 2, за исключением детектора. Видно, что для ошибки, равной единице, детектор должен обнаруживать состояние $\begin{Vmatrix}0\\1\end{Vmatrix}$, для ошибки, равной b — состояние $\begin{Vmatrix}0\\b\end{Vmatrix}$ и для b^2 — состояние $\begin{Vmatrix}0\\b^2\end{Vmatrix}$.

Наиболее простое устройство изображено на рис. 10. Теперь детектор определяет не только символ, подлежащий исправлению, но и то, во что его надо исправить. Ключ „*и*“ (&) срабатывает в том случае, когда первый

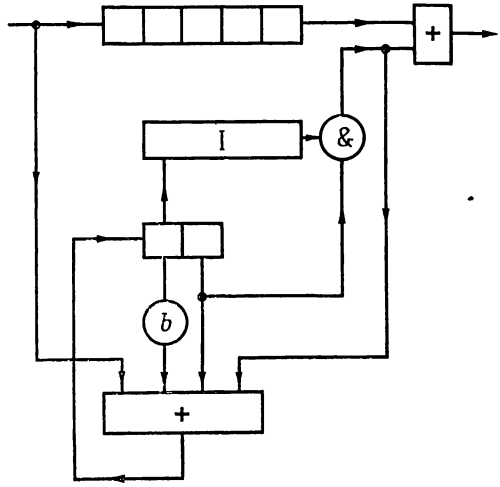


Рис. 10. Цифра I обозначает схему «не».

элемент регистра содержит нуль и тем самым позволяет добавить второй элемент к выходу главного регистра сдвига.

Построение регистра сдвига из элементов с четырьмя устойчивыми состояниями

Вопрос построения кодирующего и декодирующего устройств ясен, однако все же не ясно, как действительно построить основной регистр сдвига, изображенный на рис. 11. Для этого используется представление элементов из поля Галуа. Любой символ сообщения или любой элемент регистра может быть записан как

$$y = (A + Bb),$$

где A и B принимают значения нуль или единица; y может быть представлен вектором (A, B).

Состояние 0 представляется как (0, 0).
Состояние 1 представляется как (1, 0).
Состояние b представляется как (0, 1).
Состояние b^2 представляется как (1, 1).

Если суммируются два элемента y_1 и y_2, то ясно, что A и B суммируются раздельно.

Далее, если элемент y умножается на b, то

$$by=(Ab+Bb^2)=B+(A+B)\,b, \tag{39}$$

так что вектор, представляющий by, является $(B, A+B)$. [1])

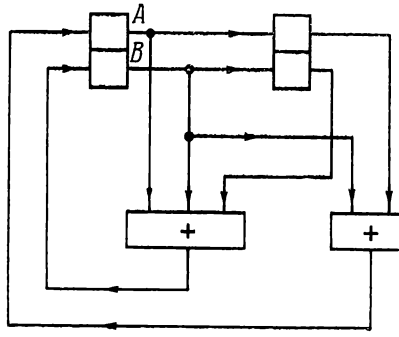


Рис. 11.

Таким образом, регистр сдвига может быть построен из двоичных элементов, которые осуществляют соответствующие операции над A и B. Двоичное представление основного регистра сдвига изображено на рис. 11.

Общий случай кодов для недвоичных сообщений

Идеи, которые иллюстрировались примерами, легко могут быть обобщены. Если символы сообщения принимают 2^l значений, то сначала необходимо найти характеристическое уравнение r-й степени с коэффициентами

[1]) Так как $b^2=1+b$. — *Прим. ред.*

из $GF(2^l)$, которое имеет циклическую структуру с 2^l-1 циклами, каждый длиной $h=(2^{lr}-1)/(2^l-1)$; в общем случае это действительно может быть сделано. Когда это сделано, циклы могут быть выписаны и может оказаться, что либо цикл, который начинается с

$$\begin{Vmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Vmatrix},$$

содержит

$$\begin{Vmatrix} b \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Vmatrix}, \quad \begin{Vmatrix} b^2 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Vmatrix} \text{ и т. д.,}$$

либо он их не содержит. В последнем случае ясно, что результатом является код, исправляющий единичную ошибку 1, b, b^2 и т. д. каждого из символов. В противном случае ясно, что 2^{l-1} должен быть множителем h.

Это и является общим методом построения кода для исправления единичной ошибки, при условии, что 2^{l-1} не является множителем h.

Примеры.

1) Символы принимают 4 значения;

$$b^2+b+1=0. \tag{40}$$

Уравнение

$$\mathbf{T}^2+b\mathbf{T}+\mathbf{1}=\mathbf{0} \tag{41}$$

дает код длины 5 с 2 проверочными символами. Уравнение

$$\mathbf{T}^4+b^2\mathbf{T}^3+b^2\mathbf{T}^2+b\mathbf{T}+\mathbf{1}=\mathbf{0} \tag{42}$$

дает код длины 85 с 4 проверочными символами.

Заметим, что вышеприведенное ограничение не позволяет с помощью этого метода найти код длины 21 с 3 проверочными символами.

2) Символы принимают 8 значений. Поле определяется с помощью

$$c^3 + c + 1 = 0. \tag{43}$$

Уравнение

$$\mathbf{T}^2 + c\mathbf{T} + 1 = 0 \tag{44}$$

дает код длины 9 с 2 проверочными символами. Уравнение

$$\mathbf{T}^3 + c\mathbf{T} + 1 = 0 \tag{45}$$

дает код длины 73 с 3 проверочными символами.

3) Символы принимают 16 значений. Поле определяется при помощи формулы

$$a^4 + a + 1 = 0; \tag{46}$$

$$\mathbf{T}^2 + a\mathbf{T} + 1 = 0 \tag{47}$$

дает код длины 17 с 2 проверочными символами.

При параллельной передаче информации этим способом можно построить много различных кодов для исправления единичных ошибок. Но использовать эти соображения для исправления пакетов ошибок не имеет смысла. До сих пор еще не найдено подходящее характеристическое уравнение для исправления пакетов при параллельной передаче данных.

Вероятно, такое исправление ошибок было бы плодотворным при магнитной записи, когда двоичная информация обрабатывается параллельно.

Заключение

Этим завершается описание кодов. Следует еще раз подчеркнуть, что требуемая для реализации аппаратура чрезвычайно проста и, тем не менее, может существенно изменить характеристики системы связи.

Теперь основной проблемой является перечисление различных связей в регистрах сдвига и получение раз-

личных кодов, а также согласование полученных кодов с характеристиками каналов, которые предстоит еще измерить.

Из описанных кодов, кажется, наиболее подходящими с практической точки зрения являются коды Файра, исправляющие пакеты длины l; оставшиеся потенциальные возможности кода используются для дополнительного обнаружения ошибок. Здесь и наибольшая простота и дополнительно еще обеспечивается некоторая защита. Инженеры связи сопротивляются еще применению чисто исправляющих кодов, потому что они боятся возможности появления катастрофически больших ошибок, которые не будут обнаружены, если не применяется некоторая система обнаружения.

Коды Боуза—Чоудхури следует применять тогда, когда ошибки появляются случайно. Следует отметить, что детектор для двойных ошибок менее громоздок, так как при этом необходимо различать значительно меньше наборов, чем при тройных ошибках, когда число наборов становится уже очень большим.

Было бы хорошо, если бы оптимальная система содержала иерархию описанных кодов и сообщение кодировалось бы несколько раз на различных уровнях. Но это уже проблема будущего.

ЛИТЕРАТУРА

1. Abramson N. M., A Class of systematic codes for non-independent errors, Electronic Res. Lab., Stanford University, Stanford, Calif., Rept. № 51; December 30, 1958.

2. Melas C. M., A new group of correction of dependent errors in data transmission, *IBM J. Res. and Dev.*, **4**, January 1960, 58—65.

3. Fire P., A class of multiple-error-correcting binary codes for non-independent errors, Sylvania Electronics Systems, Rept. BSL-E-2, March 1959; also presented at AIEE Mtg., Chicago, Ill., October 1959.

4. Elspas B., The theory of autonomous linear sequential networks, *IRE Trans. on Circuit Theory*, **CT-6**, March (1959), 45—60. [Русский перевод: Элспас Б., Теория автономных линейных последовательных сетей, Кибернетический сб., вып. 7, ИЛ, М., 1963, 90—128.]

5. Bose R. C., Ray-Chaudhuri D. K., On a class of error-correcting binary group codes, *Inform. and Control*, **3**, March (1960), 68—79. [Русский перевод: Боуз Р. К., Рой-Чоуд-хури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сб., вып. 2, ИЛ, М., 1961, 83—94.]

6. Meggitt J. E., Error correcting codes for correcting bursts of errors, *IBM J. Res. and Dev.*, **4**, July (1960), 329—334.

7. Peterson W. W., Brown D. T., Cyclic codes for error detection, *Proc. IRE*, **49**, January (1961), 228—235.

8. Peterson W. W., Encoding and error-correction procedures for the Bose—Chaudhuri codes, *IRE Trans. on Inform. Theory*, **IT-6**, September (1960), 459—470. [Русский перевод: Питерсон У., Кодирование и исправление ошибок для кодов Боуза — Чоудхури, Кибернетический сб., вып. 6, ИЛ, М., 1963, 25—54.]

СОДЕРЖАНИЕ

ТЕОРИЯ КОДИРОВАНИЯ

Редактор *А. А. Бряндинская*
Художник *М. Г. Ровенский*
Художественный редактор
В. И. Шаповалов
Технический редактор *А. В. Грушин*
Корректор *Т. А. Палладина*

Сдано в производство 1/II 1964 г.
Подписано к печати 27/IV 1964 г.
Бумага 84 × 108$^{1}/_{32}$ = 4,1 бум. л.
13,3 печ. л.
Уч.-изд. л. 12,1. Изд. № 1/2273.
Цена 85 коп. Зак. 1307.
(Темплан 1964 г. из-ва ИЛ, пор. № 15)

ИЗДАТЕЛЬСТВО «МИР»
Москва, 1-й Рижский пер., 2

Первая Образцовая типография
имени А. А. Жданова
Главполиграфпрома Государственного
комитета Совета Министров СССР
по печати
Москва, Ж-54, Валовая. 28

Издательство „Мир“

в 1965 году
выпустит следующие книги:

Фано Р., **Передача информации. Статистическая теория связи.** Нью-Йорк, 1961, перевод с английского, 20 изд. л.

В книге известного американского ученого Р. Фано систематически излагаются основы теории информации; наряду с известными результатами шенноновской теории кодирования приводится ряд новых интересных данных. Автор широко использует физическую интерпретацию изучаемого материала и приводит многочисленные примеры. Большое количество хорошо подобранных задач позволяет читателю закрепить полученные знания.

Кибернетический сборник. Новая серия, вып. 1 и 2, по 12 изд. л.

В 1960—1964 годах Издательством иностранной литературы и издательством „Мир“ выпускалась серия кибернетических сборников (выпуски 1—9), включающих работы зарубежных ученых, содержащие наиболее ценные данные по теории кодирования, теории автоматов и т. п. Новая серия является в некотором смысле продолжением предыдущей, но будет отличаться от нее более разнообразной тематикой, в частности, более широким освещением приложений кибернетических методов в самых различных областях науки и техники.

Издательство „Мир"

в 1965 году
выпустит следующие книги:

Ледли Р., **Программирование и использование цифровых вычислительных машин.** Нью-Йорк, 1962, перевод с английского, 30 изд. л.

Учебник для американских колледжей, посвященный современным идеям в области программирования для цифровых математических машин и разнообразным применениям этих машин. Автор уделяет большое внимание основным методам автоматизации программирования и дает подробное описание языков автоматического программирования (АЛГОЛ и КОБОЛ). В заключение описываются методы обработки данных (методы поиска, сортировки, упорядочения и кодирования информации).

Шварц Л., **Математические методы для физических наук.** Париж, 1961, перевод с французского, 25 изд. л.

Книга известного французского ученого Лорана Шварца, одного из создателей теории обобщенных функций, написана специально для физиков, механиков и инженеров, желающих применять эту теорию в практических задачах. Материал книги — классический: интеграл, теория потенциала, уравнения математической физики, ряды и интегралы Фурье, преобразование Лапласа. Однако весь этот классический материал изложен с новой точки зрения, пронизанной идеями теории обобщенных функций. Особенностью книги является то, что автор, не требуя от читателя большой математической подготовки, подводит его к самым актуальным вопросам современного математического анализа.

Цена 85 к.