Кодирование информации

1. Количество информации. *Кодирование* - представление сообщения последовательностью элементарных символов.

Рассмотрим кодирование дискретных сообщений. Символы в сообщениях могут относиться к алфавиту, включающему n букв (буква - символ сообщения). Однако число элементов кода k существенно ограничено сверху энергетическими соображениями, т.е. часто n > k. Так, если отношение сигнал/помеха для надежного различения уровня сигнала должно быть не менее q, то наименьшая амплитуда для представления одного из k символов должна быть q*g, где g - амплитуда помехи, а наибольшая амплитуда соответственно q*g*k. Мощность передатчика пропорциональна квадрату амплитуды сигнала (тока или напряжения), т.е. должна превышать величину, пропорциональную $(q*g*k)^2$. В связи с этим распространено двоичное кодирование с k = 2. При двоичном кодировании сообщений с n типами букв, каждая из n букв кодируется определенной комбинацией 1 и 0 (например, код ASCII).

Кодирование аналоговых сообщений после их предварительной дискретизации должно выполняться в соответствии с теоремой Котельникова: если в спектре функции f(t) нет частот выше F_B, то эта функция может быть полностью восстановлена по совокупности своих значений, определенных в моменты времени t_k, отстоящие друг от друга на величину 1/(2*Fв). Для передачи аналогового сигнала производится его дискретизация с частотой отсчетов $2*F_B$ и выполняется кодово-импульсная модуляция последовательности отсчетов.

Количество информации в сообщении (элементе сообщения) определяется по формуле

$$I = -\log_2 P,$$

где P - вероятность появления сообщения (элемента сообщения). Из этой формулы следует, что единица измерения количества информации есть количество информации, содержащееся в одном бите двоичного кода при условии равной вероятности появления в нем 1 и 0. В то же время один разряд десятичного кода содержит $I = -\log_2 P = 3{,}32$ единиц информации (при том же условии равновероятности появления десятичных символов, т.е. при P = 0,1).

2. Энтропия. Энтропия источника информации с независимыми и равновероятными сообщениями есть среднее арифметическое количеств информации сообщений

$$H = -\sum_{k=1..N} P_k * \log_2 P_k$$

где P_k - вероятность появления k-го сообщения. Другими словами, энтропия есть мера неопределенности ожидаемой информации.

П р и м е р. Пусть имеем два источника информации, один передает двоичный код с равновероятным появлением в нем 1 и 0, другой имеет вероятность 1, равную 2^{-10}, и вероятность 0, равную $1-2^{-10}$. Очевидно, что неопределенность в получении в очередном такте символа 1 или 0 от первого источника выше, чем от второго. Это подтверждается количественно оценкой энтропии: у первого источника H = 1, у второго приблизительно $H = -2^{-10}*\log_2 2^{-10}$, т.е. значительно меньше.

3. Коэффициент избыточности сообщения. Коэффициент избыточности сообщения A определяется по формуле

$r = (I_{max} - I)/I_{max}$,

где I - количество информации в сообщении A, I_{max} - максимально возможное количество информации в сообщении той же длины, что и A.

Пример избыточности дают сообщения на естественных языках, так, у русского языка r находится в пределах 0,3...0,5.

Наличие избыточности позволяет ставить вопрос о сжатии информации без ее потери в передаваемых сообщениях.

4. Основные используемые коды. Широко используются двоичные коды:

EBCDIC (Extended Binary Coded Decimal Interchange Code) - символы кодируются восемью битами; популярен благодаря его использованию в IBM;

ASCII (American Standards Committee for Information Interchange) - семибитовый двоичный код.

Оба этих кода включают битовые комбинации для печатаемых символов и некоторых распространенных командных слов типа NUL, CR, ACK, NAK и др.

Для кодировки русского текста нужно вводить дополнительные битовые комбинации. Семибитовая кодировка здесь уже недостаточна. В восьмибитовой кодировке нужно под русские символы отводить двоичные комбинации, не занятые в общепринятом коде, чтобы сохранять неизменной кодировку латинских букв и других символов. Так возникли кодировка КОИ-8, затем при появлении персональных ЭВМ - альтернативная кодировка и при переходе к Windows - кодировка 1251. Множество используемых кодировок существенно усложняет проблему согласования почтовых программ в глобальных сетях.

5. Асинхронное и синхронное кодирование. Для правильного распознавания позиций символов в передаваемом сообщении получатель должен знать границы передаваемых элементов сообщения. Для этого необходима синхронизация передатчика и приемника. Использование специального дополнительного провода для сигналов синхронизации (в этом случае имеем *битовую* синхронизацию) слишком дорого, поэтому используют другие способы синхронизации.

В *асинхронном режиме* применяют коды, в которых явно выделены границы каждого символа (байта) специальными стартовым и стоповым символами. Подобные побайтно выделенные коды называют *байт-ориентированными,* а способ передачи - *байтовой* синхронизацией. Однако это увеличивает число битов, не относящихся собственно к сообщению.

В *синхронном режиме* синхронизм поддерживается во время передачи всего информационного блока без обрамления каждого байта. Такие коды называют *бит-ориентированными*. Для входа в синхронизм нужно обозначать границы лишь всего передаваемого блока информации с помощью специальных начальной и конечной

комбинаций байтов (обычно это двубайтовые комбинации). В этом случае синхронизация называется *блочной (фреймовой*).

Для обрамления текстового блока (текст состоит только из печатаемых символов) можно использовать символы, отличающиеся от печатаемых. Для обрамления двоичных блоков применяют специальный символ (обозначим его DLE), который благодаря *стаффингу* становится уникальным. Уникальность заключается в том, что если DLE встречается внутри блока, то сразу вслед за ним вставляется еще один DLE. Приемник будет игнорировать каждый второй идущий подряд символ DLE. Если же DLE встречается без добавления, то это граница блока.

6. Манчестерское кодирование. Передаваемые данные представляются электрическими сигналами. Возможны коды RZ (Return-to-zero), использующие двуполярные сигналы для изображения 1 и 0, и коды NRZ (non-return-to-zero) - коды без возвращения к нулю.

Для кодирования информации наибольшее распространение получили самосинхронизирующиеся коды, так как при этом отпадает необходимость иметь дополнительную линию для передачи синхросигналов между узлами сети. В ЛВС чаще других применяют манчестерский код, одна из разновидностей которого пояснена на рис. 3.1. Самосинхронизация обеспечивается благодаря формированию синхроимпульсов из перепадов, имеющихся в каждом такте манчестерского кода.

Представленная на рис. 3.1 разновидность манчестерского кода используется при байт-ориентированном кодировании, при котором каждый байт, состоящий из 1 и 0, обрамляется символами j и k. В этом случае станция, получившая полномочия, начинает передавать серию сигналов jkjkjk... для того, чтобы станция-получатель могла войти в синхронизм с передающей станцией. После нескольких пар jk начинают передаваться байты самого сообщения. Различение четырех возможных значений сигнала выполняется в соответствии с правилами кодирования, представленными в нижней части рисунка.

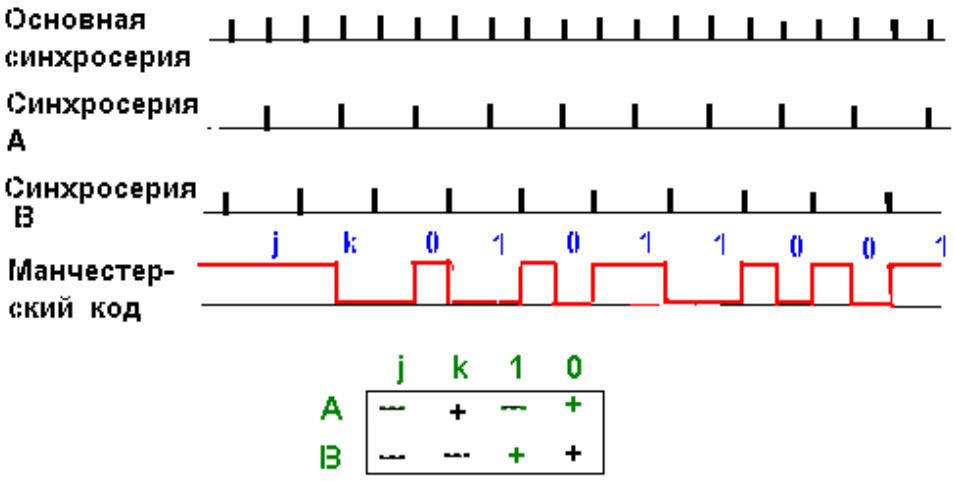

Рис. 3.1. Манчестерское кодирование

В случае бит-ориентированного кода после входа в синхронизм не нужно обрамлять байты символами j и k, т.е. используется двузначное кодирование. Чаще используется код, в котором "1" представляется положительным, а "0" - отрицательным перепадом.

7. Способы контроля правильности передачи данных. Управление правильностью (помехозащищенностью) передачи информации выполняется с помощью помехоустойчивого кодирования. Различают коды, обнаруживающие ошибки, и корректирующие коды, которые дополнительно к обнаружению еще и исправляют ошибки. Помехозащищенность достигается с помощью введения избыточности. Устранение ошибок с помощью *корректирующих кодов* (такое управление называют Forward Error Control) реализуют в симплексных каналах связи. В дуплексных каналах достаточно применения кодов, *обнаруживающих* ошибки (Feedback or Backward Error Control), так как сигнализация об ошибке вызывает повторную передачу от источника. Это основные методы, используемые в информационных сетях.

Простейшими способами обнаружения ошибок являются контрольное суммирование, проверка на нечетность. Однако они недостаточно надежны, особенно при появлении пачек ошибок. Поэтому в качестве надежных обнаруживающих кодов применяют *циклические коды*. Примером корректирующего кода является код Хемминга.

8. Код Хемминга. В коде Хемминга вводится понятие *кодового расстояния* d (расстояния между двумя кодами), равного числу разрядов с неодинаковыми значениями. Возможности исправления ошибок связаны с минимальным кодовым расстоянием d_{min}. Исправляются ошибки кратности r = ent (d_{min}-1)/2 и обнаруживаются ошибки кратности d_{min}-1 (здесь ent означает "целая часть"). Так, при контроле на нечетность d_{min} = 2 и обнаруживаются одиночные ошибки. В коде Хемминга d_{min} = 3. Дополнительно к информационным разрядам вводится $L = \log_2 K$ избыточных контролирующих разрядов, где K - число информационных разрядов, L округляется до ближайшего большего целого значения. L-разрядный контролирующий код есть инвертированный результат поразрядного сложения (т.е. сложения по модулю 2) номеров тех информационных разрядов, значения которых равны 1.

П р и м е р 1. Пусть имеем основной код 100110, т.е. К = 6. Следовательно, L = 3 и дополнительный код равен

010 # 011 # 110 = 111,

где # - символ операции поразрядного сложения, и после инвертирования имеем 000. Теперь вместе с основным кодом будет передан и дополнительный. На приемном конце вновь рассчитывается дополнительный код и сравнивается с переданным. Фиксируется код сравнения (поразрядная операция отрицания равнозначности), и если он отличен от нуля, то его значение есть номер ошибочно принятого разряда основного кода. Так, если принят код 100010, то рассчитанный в приемнике дополнительный код равен инверсии от 010 # 110 = 100, т.е. 011, что означает ошибку в 3-м разряде.

П р и м е р 2. Основной код 1100000, дополнительный код 110 (результат инверсии кода 110 # 111 = 001). Пусть принятый код 1101000, его дополнительный код 010, код сравнения 100, т.е. ошибка в четвертом разряде.

9. Циклические коды. К числу эффективных кодов, обнаруживающих одиночные, кратные ошибки и пачки ошибок, относятся *циклические коды* (CRC - Cyclic Redundance Code). Они высоконадежны и могут применяться при блочной

синхронизации, при которой выделение, например, бита нечетности было бы затруднительно.

Один из вариантов циклического кодирования заключается в умножении исходного кода на образующий полином g(x), а декодирование - в делении на g(x). Если остаток от деления не равен нулю, то произошла ошибка. Сигнал об ошибке поступает на передатчик, что вызывает повторную передачу.

Образующий полином есть двоичное представление одного из простых множителей, на которые раскладывается число X^n-1, где X^n обозначает единицу в n-м разряде, n равно числу разрядов кодовой группы. Так, если n = 10 и X = 2, то X^n-1 = 1023 = 11*93, и если g(X)=11 или в двоичном коде 1011, то примеры циклических кодов A_i*g(X) чисел A_i в кодовой группе при этом образующем полиноме можно видеть в следующей табл. 3.1.

Основной вариант циклического кода, широко применяемый на практике, отличается от предыдущего тем, что операция деления на образующий полином заменяется следующим алгоритмом: 1) к исходному кодируемому числу А справа приписывается К нулей, где К - число битов в образующем полиноме, уменьшенное на единицу; 2) над полученным числом A*(2^K) выполняется операция О, отличающаяся от деления тем, что на каждом шаге операции вместо вычитания выполняется поразрядная операция "исключающее ИЛИ": 3) полученный остаток В и есть CRC - избыточный К-разрядный код, который заменяет в закодированном числе С приписанные справа К нулей, т.е.

C= A*(2^K)+B.

На приемном конце над кодом С выполняется операция О. Если остаток не равен нулю, то при передаче произошла ошибка и нужна повторная передача кода А.

.П р и м е р. Пусть А = 1001 1101, образующий полином 11001.

Так как К = 4, то A*($2^{K)=100111010000.}$ Выполнение операции О расчета циклического кода показано на рис. 3.2.

Таблица 3.1

Число	Циклический код	Число	Циклический код
0	0000000000.	13	0010001111
1	0000001011	14	0010011010
2	0000010110	15	0010100101
3	0000100001	16	0011000110
5	0000110111	18	0011000110
6	0001000010	19	0011010001
.....	.....	.....	.....

Положительными свойствами циклических кодов являются малая вероятность необнаружения ошибки и сравнительно небольшое число избыточных разрядов.

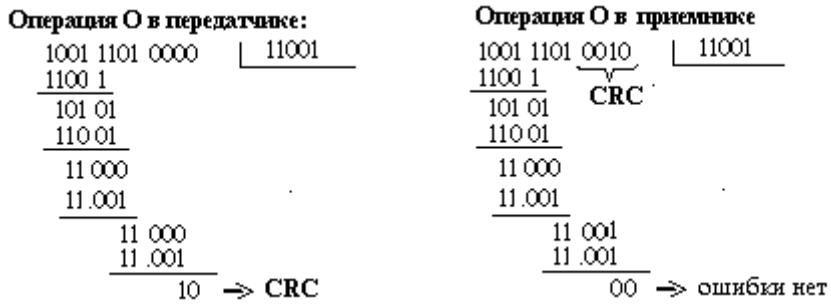

Рис. 3.2. Пример получения циклического кода

Общепринятое обозначение образующих полиномов дает следующий пример:

$g(X) = X^{16} + X^{12} + X^5 + 1,$

что эквивалентно коду 1 0001 0000 0010 0001. Этот полином используется в протоколе V.42 для кодирования кодовых групп в 240 разрядов с двумя избыточными байтами. В этом протоколе возможен и образующий полином для четырех избыточных байтов

$g(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5$

$+ X^4 + X^2 + 1.$

10. Коэффициент сжатия. Наличие в сообщениях избыточности позволяет ставить вопрос о сжатии данных, т.е. о передаче того же количества информации с помощью последовательностей символов меньшей длины. Для этого используются специальные алгоритмы сжатия, уменьшающие избыточность. Эффект сжатия оценивают *коэффициентом сжатия*

$K = n/q,$

где n - число минимально необходимых символов для передачи сообщения (практически это число символов на выходе эталонного алгоритма сжатия); q - число символов в сообщении, сжатом данным алгоритмом. Так, при двоичном кодировании n равно энтропии источника информации.

Наряду с методами сжатия, не уменьшающими количество информации в сообщении, применяются методы сжатия, основанные на потере малосущественной информации.

11. Алгоритмы сжатия. Сжатие данных осуществляется либо на прикладном уровне с помощью программ сжатия, таких, как ARJ, либо с помощью устройств защиты от ошибок (УЗО) непосредственно в составе модемов по протоколам типа V.42bis.

Очевидный способ сжатия числовой информации, представленной в коде ASCII, заключается в использовании сокращенного кода с четырьмя битами на символ вместо восьми, так как передается набор, включающий только 10 цифр, символы "точка", "запятая" и "пробел".

Среди простых алгоритмов сжатия наиболее известны *алгоритмы RLE (Run Length Encoding)*. В них вместо передачи цепочки из одинаковых символов передаются символ и значение длины цепочки. Метод эффективен при передаче растровых изображений, но малополезен при передаче текста.

К методам сжатия относят также *методы разностного кодирования*, поскольку разности амплитуд отсчетов представляются меньшим числом разрядов, чем сами амплитуды. Разностное кодирование реализовано в методах дельта-модуляции и ее разновидностях.

Предсказывающие (предиктивные) методы основаны на экстраполяции значений амплитуд отсчетов, и если выполнено условие

Ar - Ap > d,

то отсчет должен быть передан, иначе он является избыточным; здесь Ar и Ap - амплитуды реального и предсказанного отсчетов, d - допуск (допустимая погрешность представления амплитуд). Иллюстрация предсказывающего метода с линейной экстраполяцией представлена рис. 3.3. Здесь точками показаны предсказываемые значения сигнала. Если точка выходит за пределы "коридора" (допуска d), показанного пунктирными линиями, то происходит передача отсчета. На рисунке передаваемые отсчеты отмечены темными кружками в моменты времени $t_{1, t2, t4, t7}$. Если передачи отсчета нет, то на приемном конце принимается экстраполированное значение.

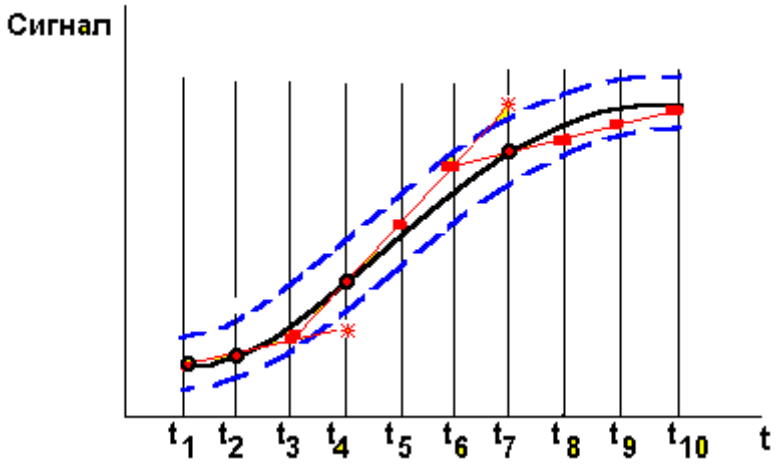

Рис. 3.3. Предиктивное кодирование

Методы MPEG (Moving Pictures Experts Group) используют предсказывающее кодирование изображений (для сжатия данных о движущихся объектах вместе со звуком). Так, если передавать только изменившиеся во времени пиксели изображения, то достигается сжатие в несколько десятков раз. Этот алгоритм сжатия используется также в стандарте H.261 ITU. Методы MPEG становятся мировыми стандартами для цифрового телевидения.

Для сжатия данных об изображениях можно использовать также методы типа JPEG (Joint Photographic Expert Group), основанные на потере малосущественной

информации (не различимые для глаза оттенки кодируются одинаково, коды могут стать короче). В этих методах передаваемая последовательность пикселей делится на блоки, в каждом блоке производится преобразование Фурье, устраняются высокие частоты, передаются коэффициенты разложения для оставшихся частот, по ним в приемнике изображение восстанавливается.

Другой принцип воплощен в фрактальном кодировании, при котором изображение, представленное совокупностью линий, описывается уравнениями этих линий.

Более универсален широко известный метод Хаффмена, относящийся к статистическим методам сжатия. Идея метода - часто повторяющиеся символы нужно кодировать более короткими цепочками битов, чем цепочки редких символов. Строится двоичное дерево, листья соответствуют кодируемым символам, код символа представляется последовательностью значений ребер (эти значения в двоичном дереве суть 1 и 0), ведущих от корня к листу. Листья символов с высокой вероятностью появления находятся ближе к корню, чем листья маловероятных символов.

Распознавание кода, сжатого по методу Хаффмена, выполняется по алгоритму, аналогичному алгоритмам восходящего грамматического разбора. Например, пусть набор из восьми символов (A, B, C, D, E, F, G, H) имеет следующие правила кодирования:

A ::= 10; B ::= 01; C ::= 111; D ::= 110;

E ::= 0001; F ::= 0000; G ::= 0011; H ::= 0010.

Тогда при распознавании входного потока 101100000110 в стек распознавателя заносится 1, но 1 не совпадает с правой частью ни одного из правил. Поэтому в стек добавляется следующий символ 0. Полученная комбинация 10 распознается и заменяется на A. В стек поступает следующий символ 1, затем 1, затем 0. Сочетание 110 совпадает с правой частью правила для D. Теперь в стеке AD, заносятся следующие символы 0000 и т.д.

Недостаток метода заключается в необходимости знать вероятности символов. Если заранее они не известны, то требуются два прохода: на одном в передатчике подсчитываются вероятности, на другом эти вероятности и сжатый поток символов передаются к приемнику. Однако двухпроходность не всегда возможна.

Этот недостаток устраняется в однопроходных алгоритмах адаптивного сжатия, в которых схема кодирования есть схема приспособления к текущим особенностям передаваемого потока символов. Поскольку схема кодирования известна как кодеру, так и декодеру, сжатое сообщение будет восстановлено на приемном конце.

Обобщением этого способа является алгоритм, основанный на *словаре сжатия данных*. В нем происходит выделение и запоминание в словаре повторяющихся цепочек символов, которые кодируются цепочками меньшей длины.

Интересен *алгоритм "стопка книг"*, в котором код символа равен его порядковому номеру в списке. Появление символа в кодируемом потоке вызывает его перемещение в начало списка. Очевидно, что часто встречающиеся символы будут тяготеть к малым номерам, а они кодируются более короткими цепочками 1 и 0.

Кроме упомянутых алгоритмов сжатия существует ряд других алгоритмов, например LZ-алгоритмы (*алгоритмы Лемпеля-Зива*). В частности, один из них (LZW) применен в протоколе V.42bis.

12. Сжатие данных по методу Лемпеля-Зива. Лемель и Зив используют следующую идею: если в тексте сообщения появляется последовательность из двух ранее уже встречавшихся символов, то эта последовательсность объявляется новым символом, для нее назначается код, который при определенных условиях может быть значительно короче исходной последовательности. В дальнейшем в сжатом сообщении вместо исходной последовательности записывается назначенный код. При декодировании повторяются аналогичные действия и потому становятся известными последовательности символов для каждого кода.

Одна из алгоритмических реализаций этой идеи включает следующие операции. Первоначально каждому символу алфавита присваивается определенный код (коды - порядковые номера, начиная с 0). При *кодировании*:

1. Выбирается первый символ сообщения и заменяется на его код.

2. Выбираются следующие два символа и заменяются своими кодами. Одновременно этой комбинации двух символов присваивается свой код. Обычно это номер, равный числу уже использованных кодов. Так, если алфавит включает 8 символов, имеющих коды от 000 до 111, то первая двухсимвольная комбинация получит код 1000, следующая - код 1001 и т.д.

3. Выбираются из исходного текста очередные 2, 3,...N символов до тех пор, пока не образуется еще не встречавшаяся комбинация. Тогда этой комбинации присваивается очередной код, и поскольку совокупность A из первых N-1 символов уже встречалась, то она имеет свой код, который и записывается вместо этих N-1 символов. Каждый акт введения нового кода назовем шагом кодирования.

4. Процесс продолжается до исчерпания исходного текста.

При *декодировании* код первого символа, а затем второго и третьего заменяются на символы алфавита. При этом становится известным код комбинации второго и третьего символов. В следующей позиции могут быть только коды уже известных символов и их комбинаций. Процесс декодирования продолжается до исчерпания сжатого текста.

Сколько двоичных разрядов нужно выделять для кодирования? Ответ может быть следующим: число разрядов R на каждом шаге кодирования равно числу разрядов в наиболее длинном из использованных кодов (т.е. числу разрядов в последнем использованном порядковом номере). Поэтому если последний использованный код (порядковый номер) равен 13=1101, то коды A всех комбинаций должны быть четырехразрядными при кодировании вплоть до появления номера 16, после чего все коды символов начинают рассматриваться как пятиразрядные (R=5).

Пример. Пусть исходный текст представляет собой двоичный код (первая строка таблицы 1), т.е. символами алфавита являются 0 и 1. Коды этих символов соответственно также 0 и 1. Образующийся по методу Лемпеля-Зива код (LZ-код) показан во второй строке таблицы 1. В третьей строке отмечены шаги кодирования, после которых происходит переход на представление кодов A увеличенным числом разрядов R. Так, на первом шаге вводится код 10 для комбинации 00 и поэтому на

следующих двух шагах R=2, после третьего шага R=3, после седьмого шага R=4, т.е. в общем случае R=K после шага 2^{K-1}-1.

В приведенном примере LZ-код оказался даже длиннее исходного кода, так как обычно короткие тексты не дают эффекта сжатия. Эффект сжатия проявляется в достаточно длинных текстах и особенно заметен в графических файлах.

Таблица 1

Исходный текст	0.00.000. 01. 11. 111.1111. 110. 0000.00000. 1101. 1110.
LZ-код	0.00.100.001.0011.1011.1101. 1010.00110.10010.10001.10110.
R	2 3 4
Вводимые коды	- 10 11 100 101 110 111 1000 1001 1010 1011 1100

В другой известной реализации LZ-метода любая ранее встречавшаяся последовательность в сжатом тексте представляет собой совокупность следующих данных:

- номер первого символа в ранее встречавшейся последовательности;
- число символов в последовательности;
- следующий символ в текущей позиции кодируемого текста.