**Практическое занятие № 3**

**Тема: Исследование методов сжатия дискретных сообщений**

**1. Теоретическая часть**

1.1 Математические основы сжатия дискретных сообщений

1.2 Методы сжатия дискретных сообщений

**2. Практическая часть**

**1.1 Математические основы сжатия дискретных сообщений**

Сжатие информации представляет собой операцию, в результате которой данному коду или сообщению ставится в соответствие более короткий код или сообщение.

Сжатие информации ставит целью - ускорение и удешевление процессов механизированной обработки, хранения и поиска информации, экономии памяти аппаратуры систем документальной электросвязи. При сжатии следует стремиться к минимальной неоднозначности сжатых кодов при максимальной простоте алгоритма сжатия. Рассмотрим наиболее характерные методы сжатия информации.

**Сжатие информации делением кода на части**, меньше некоторой заданной величины А, заключается в том, что исходный код делится на части, полученные части кода складываются между собой, либо по правилам двоичной арифметики, либо по модулю 2. Например, исходный код 101011010110 при ***А=4.***

|  |  |
| --- | --- |
| 1. 0 1 0 2. 1 0 1   0 1 1 0  1 1 1 0 1  сжатый код двоичной логикой | 1. 0 1 0 2. 1 0 1   0 1 1 0  0 0 0 1  сжатый код по модулю 2 |

**Сжатие информации с побуквенным сдвигом**.

Очень часто обрабатываемая информация бывает представлена в виде набора однородных массивов, в которых элементы столбцов или строк массивов расположены в нарастающем порядке. Длина строки известна. Если считать старшими разряды, расположенные левее данного элемента, а младшими – расположенные правее, то можно заметить, что во многих случаях строки матриц отличаются друг от друга в младших разрядах. Если при записи каждого последующего элемента массива отбрасывать все повторяющиеся в предыдущем элементы, например в строке стоящие подряд элементы старших разрядов, то массивы могут быть сокращены от 2 до 10 и более разрядов.

Для учета выброшенных разрядов вводится знак раздела, который позволяет отделить элементы в свернутом массиве. В случае полного повторения строк записывается соответствующее количество раз *р*. При развертывании вместо знака *р* восстанавливаются все пропущенные разряды, которые были до элемента, стоящего непосредственно за *р* в сжатом тексте.

**Для примера рассмотрим следующий массив:**

9 5 7 0 1 2 4

9 5 7 0 1 2 5

9 5 7 0 3 8 6

9 5 7 0 3 9 0

1 2 3 4 5 6 7

1 2 3 4 5 9 1

1 2 3 4 5 9 3

**Свернутый массив будет иметь вид:**

1. 5 7 0 1 2 4

р 5 р 3 8 6 р

9 0 1 2 3 4 5

6 7 р 9 1 р 3

**Расшифровка (развертывание) происходит с конца массива. Переход на следующую строку происходит по двум условиям:**

1. 5 7 0 1 2 4

. . . . . . 5

. . . . 3 8 6

. . . . . 9 0

1 2 3 4 5 6 7

. . . . . 9 1

. . . . . . 3

Пропущенные цифры заносятся автоматически по аналогичным разрядам предыдущей строки. Заполнение производится с начала массива. Этот метод можно развить и для свертывания массивов, в которых повторяющиеся разряды встречаются не только с начала строки. Если в строке один повторяющийся участок, то кроме *р* добавляется еще один дополнительный символ *К*, означающий конец строки. Расшифровка ведется от *К* до *К.*  Длина строки известна. Нужно, чтобы оставшиеся между *К* цифры вместе с пропущенными разрядами составляли бы не более одного участка с повторяющимися разрядами. Например:

|  |  |
| --- | --- |
| **Исходный массив** 1 2 3 4 5 6 7  1 2 3 4 5 8 6  2 1 3 4 5 2 4  2 1 3 4 5 2 9  4 2 9 4 5 2 9  4 2 9 4 5 2 9  4 2 9 4 5 2 9  5 2 9 4 5 2 1 | **Свернутый массив**  1 2 3 4 5 6 7  К р 8 6 К 2 1  р 2 4 К р 9 К  4 2 9 р К р К  р К 5 р 1 К |

Если в строке есть два повторяющихся участка, то, используя этот метод, выбрасываем больший.

Процесс развертывания массива осуществляется следующим образом: переход на следующую строку при встрече *К*

1 2 3 4 5 6 7

. . . . . 8 6

2 1 . . . 2 4

. . . . . . 9

4 2 9 . . . .

. . . . . . .

. . . . . . .

5 . . . . . 1

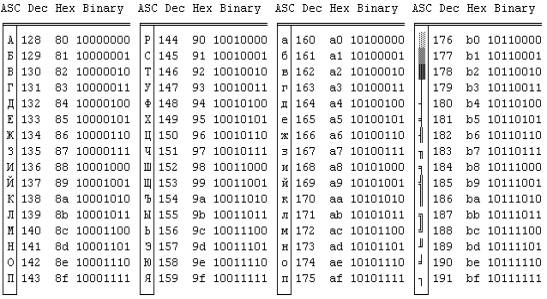
**Пропущенные цифры заполняются по аналогичным разрядам предыдущей строки, начиная с конца массива.**

1. Таблица правил выполнения арифметических действий над двоичными числами (двоичных сложения, вычитания и умножения).

|  |  |  |
| --- | --- | --- |
| **Таблица двоичного сложения** | **Таблица двоичного вычитания** | **Таблица двоичного умножения** |
| 0+0=0 0+1=1 1+0=1 1+1=10 | 0-0=0 1-0=1 1-1=0 10-1=1 | 00=0 01=0 10=0 11=1 |

При сложении двоичных чисел в каждом разряде производится сложение цифр слагаемых и переноса из соседнего младшего разряда, если он имеется. При этом необходимо учитывать, что 1+1 дают нуль в данном разряде и единицу переноса в следующий.

1. Таблица десятичных, шестнадцатеричных и двоичных ASCII - кодов



1. Таблица правил сложения по модулю 2: результат равен , если оба операнда равны; во всех остальных случаях результат равен .



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Для тернарного сложения по модулю 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z |  | ⊕(X,Y,Z) |
| 0 | 0 | 0 |  | 0 |
| 1 | 0 | 0 |  | 1 |
| 0 | 1 | 0 |  | 1 |
| 1 | 1 | 0 |  | 0 |
| 0 | 0 | 1 |  | 1 |
| 1 | 0 | 1 |  | 0 |
| 0 | 1 | 1 |  | 0 |
| 1 | 1 | 1 |  | 1 |

**1.2 Методы сжатия дискретных сообщений**

**1.2.1 Условия существования оптимального неравномерного кода**

При передаче сообщения осуществляется его преобразование в сигнал, пригодный для передачи по каналу связи. При этом необходимо согласовывать источник с каналом путём определения правила, по которому каждому элементу сообщения ставится в соответствие некоторый код, преобразуемый далее в сигнал.

В настоящее время существует два основных направления развития теории кодирования. В одном из них рассматриваются задачи повышения достоверности передачи в каналах с помехами, решаемые применением помехоустойчивых кодов, которые позволяют обнаруживать или исправлять ошибки. Такое кодирование называется помехоустойчивым. При этом избыточность кодовой последовательности выше, чем избыточность источника сообщений. Благодаря этому и оказывается возможным обнаружение и исправление ошибок передачи.

Другое направление теории кодирования связано с вопросами устранения избыточности при передаче сообщений в каналах без помех. Цель кодирования при этом состоит в таком преобразовании сообщения, при котором избыточность кодовой последовательности должна стать меньше, чем избыточность сообщений источника. В результате появляется возможность увеличения скорости передачи информации или снижаются требования к пропускной способности канала. Это особенно важно в случае, когда источники сообщений имеют большую избыточность, например, источники речевых сообщений, изображений и т.д.

Процесс кодирования с целью уменьшения избыточности источника сообщений носит название согласования источника с каналом или сжатия источника (экономного кодирования, энтропийного кодирования).

Избыточность равна нулю только в том случае, когда элементы сообщения появляются на выходе источника с равными вероятностями

() и независимо друг от друга . Если же , то оказывается возможным построение кодов, имеющих меньшую избыточность, чем источник сообщений.



**Покажем это на простейшем примере:**

Пусть источник имеет алфавит из четырех символов А, Б, В, Г с вероятностями ; ; .



**Энтропия такого источника:**

|  |  |
| --- | --- |
|  |  |

Для передачи по каналу будем использовать равномерное кодирование, например, , , , . Тогда среднее число двоичных символов в сообщении, приходящихся на один символ источника, равно двум. Поскольку это на 12,5% больше энтропии источника, то используемый код не является оптимальным.



**Рассмотрим теперь неравномерный код:**

; ; ; . В этом случае среднее число двоичных символов, приходящихся на один символ источника в сообщении,



|  |  |
| --- | --- |
| . |  |

Таким образом, среднее число двоичных символов, приходящихся на один символ источника, равно энтропии источника, т.е. для указанного источника неравномерный код оказывается более экономичным, чем равномерный.

Важно отметить, что при кодировании неравномерным кодом должна обеспечиваться возможность однозначного декодирования символов сообщения. Например, для рассмотренного источника, нецелесообразно применять код: ; ; ; , поскольку приём последовательности 10 может означать передачу символа , или двух символов и . Неоднозначно также декодирование символов 11. Для однозначного декодирования неравномерные коды должны удовлетворять **условию префиксности**: ***никакое более короткое слово не должно являться началом более длинного слова.*** Неравномерные коды, удовлетворяющие этому условию, называют **префиксными.**



Неравномерные коды позволяют в среднем уменьшить число двоичных символов на единичное информационное сообщение. Однако им присущ существенный **недостаток**: при возникновении ошибки она распространяется на все последующие элементы сообщения. Возникает ошибка синхронизации, приводящая к резкому ухудшению достоверности приёма. Этот недостаток отсутствует в равномерных кодах. ***При кодировании равномерными кодами используется одно и то же число двоичных символов – блок; поэтому такие коды называют блоковыми.***

**1.2.2 Прямая и обратная теоремы кодирования источника неравномерными кодами**

**Прямая теорема кодирования** ***состоит в том, что для любого однозначно декодируемого кода среднее число символов в двоичном кодовом слове всегда не меньше энтропии источника сообщений , и существует однозначно декодируемый код, для которого выполняется неравенство .***



**Обратная теорема кодирования** ***утверждает, что невозможно построить однозначно декодируемый код, для которого выполнялось бы неравенство .***



**Из этих теорем следует, что невозможно закодировать сообщение таким образом, чтобы средняя длина кодовых слов была меньше энтропии сообщения.** Кроме того, существует кодирование, при котором средняя длина кодового слова незначительно отличается от энтропии источника сообщений. Среднее число символов кода на сообщение можно уменьшить, если кодировать не каждый символ сообщения, а блоки по символов из алфавита . Используя кодирование блоков, можно получить среднее число символов на сообщение, сколь угодно мало отличающееся от энтропии источника, но при этом возрастает сложность кодирования.



**1.2.3 Показатели эффективности сжатия**

Наряду с коэффициентом избыточности, часто используется коэффициент сжатия источника:

|  |
| --- |
| . . |

**Коэффициент сжатия источника** ***показывает, во сколько раз можно уменьшить количество двоичных символов для представления единичного символа источника с энтропией по сравнению со случаем, когда все сообщения источника передаются равновероятно.***



Например, для источника, рассмотренного в п.1.2.1, коэффициент сжатия

|  |  |
| --- | --- |
| , |  |

т.е. скорость передачи информации по каналу связи при использовании экономичного кодирования может быть в 1,14 раза больше, чем при равномерном кодировании.

**1.2.4 Кодирование источника дискретных сообщений методом Шеннона-Фано**

Кодирование методом Шеннона – Фано рассмотрим на примере. Пусть алфавит источника содержит шесть элементов {А, Б, В, Г, Д, Е}, появляющихся с вероятностями , , , , , . Энтропия такого источника:



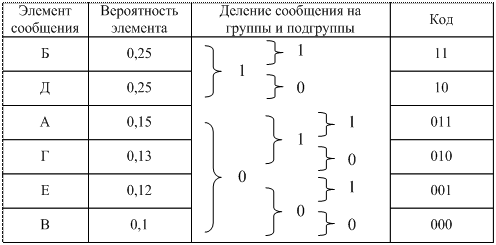
|  |
| --- |
|  |

**Алгоритм построения сжимающего кода Шеннона – Фано заключается в следующем:**

1. Все символов дискретного источника располагаются в порядке убывания вероятностей их появления (табл. 1).



**Таблица 1. Построение кода Шеннона-Фано**



2. Образованный столбец символов делится на две группы таким образом, чтобы суммарные вероятности каждой группы мало отличались друг от друга.

3. Верхняя группа кодируется символом «1», а нижняя – «0».

4. Каждая группа делится на две подгруппы с близкими суммарными вероятностями; верхняя подгруппа кодируется символом «1», а нижняя – «0».

5. Процесс деления и кодирования продолжается до тех пор, пока в каждой подгруппе не окажется по одному символу сообщения источника.

6. Записывается код для каждого символа источника; считывание кода осуществляется слева направо.

При использовании простейшего равномерного кода для кодирования шести элементов алфавита источника потребуется по три двоичных символа на каждую букву сообщения. Если же используется код Шеннона – Фано, то среднее число символов на одну букву

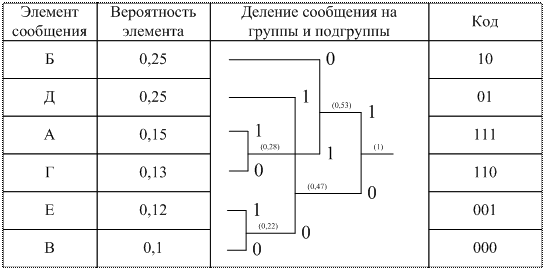
|  |
| --- |
| , |

что меньше, чем при простейшем равномерном коде и незначительно отличается от энтропии источника.

**1.2.5 Кодирование источника дискретных сообщений методом Хаффмена**

Рассмотрим еще один подход к кодированию, предложенный Хаффменом, на примере источника сообщений, заданного в табл. 2.

**Таблица 2. Построение кода Хаффмена**



**Алгоритм построения сжимающего кода Хаффмена включает в себя следующие действия:**

1. Все символов дискретного источника располагаются в таблице в порядке убывания вероятностей.



2. Два символа, имеющих наименьшие вероятности, объединяются в один блок, а их вероятности суммируются.

3. Ветви скобки, идущей к большей вероятности, присваивается символ «1», а идущей к меньшей – символ «0».

4. Операции 2 и 3 повторяются до тех пор, пока не сформируется один блок с вероятностью единица.

5. Записывается код для каждого символа источника; при этом считывание кода осуществляется справа налево.

Среднее число символов на одну букву для полученного кода

|  |
| --- |
| . |

Таким образом, для данного примера кодирование методами Хаффмена и Шеннона–Фано приводит к одинаковой эффективности. Однако **опыт кодирования показывает, что код Хаффмена часто оказывается экономичнее кода Шеннона–Фано.**

Рассмотренные методы построения сжимающих кодов широко известны и имеют практическое применение. Длина кодовой комбинации таких кодов зависит от вероятности выбора соответствующей буквы алфавита: наиболее вероятным буквам сопоставляются короткие кодовые комбинации, а менее вероятным – более длинные.

**2. Практическая часть**

**2.1 Задания первого уровня сложности**

**Задание 1.**

Используя таблицы, выполнить сжатие методами двоичного сложения, тернарного сложения, сложения по модулю 2 и побуквенного сдвига своих инициалов.

**Например:** Иванов Василий Олегович – ИВО – 100010001000001010001110.

**Примечание.** Для задач с побуквенным сдвигом выбрать одно слово с максимальным количеством совпадающих букв.

**Задание 2.**

Используя таблицы, выполнить сжатие методом побуквенного сдвига слова «аберрация».

**2.2 Задания второго уровня сложности**

**Задание 1.**

Выполнить задания первого уровня сложности.

**Задание 2.**

Используя метод Шеннона-Фэно составить кодовые комбинации соответствующих каждому сообщению, представленному в таблице 2.1.

Таблица 2.1

|  |  |  |
| --- | --- | --- |
| Сообщения | Вероятности появления сообщений | |
| Расчет 1 | Расчет 2 |
| А1 | 0,18 | 0,08 |
| А2 | 0,12 | 0,11 |
| А3 | 0,02 | 0,16 |
| А4 | 0,22 | 0,06 |
| А5 | 0,07 | 0,03 |
| А6 | 0,14 | 0,12 |
| А7 | 0,21 | 0,34 |
| А8 | 0,04 | 0,1 |

**Задание 3.**

Используя метод Хаффмена составить кодовые комбинации соответствующих каждому сообщению, представленному в таблице 2.1.

**Задание 4.**

Используя методы Шеннона-Фэно и Хаффмена составить кодовые комбинации, соответствующие каждому сообщению из представленных в таблице 2.2:

**Таблица 2.2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщения | А1 | А2 | А3 | А4 | А5 | А6 | А7 | А8 |
| Вероятности появления сообщений | 0,02 | 0,22 | 0,12 | 0,15 | 0,04 | 0,21 | 0,16 | 0,08 |

**Задание 5.**

Используя метод Хаффмена составить кодовые комбинации соответствующих каждому сообщению, представленному в таблице 2.2.

**Контрольные вопросы**

1. Сущность сжатия информации

2. Цели сжатия информации

3. Сущность условия префиксности

4. Основной недостаток неравномерного кодирования

5. Прямая теорема кодирования

6. Обратная теорема кодирования

7. Сущность коэффициента сжатия источника сообщений.