СУБД MS Access является в настоящее время одной из самых популярных среди настольных систем. Информационная система Страховая компания состоит из пяти таблиц, содержащей сведения заданной тематики. Также подготовлены запросы различного типа, по которым созданы соответствующие отчеты, реализованы формы для ввода и редактирования данных, кнопочная навигационная форма. Отчеты позволяют выводить информацию на принтер в удобном оформлении. В работе приведено подробное описание процесса разработки и создания информационной системы, представлены результаты, необходимые поясняющие иллюстрации.

1. Алексеева, Т.И., Башин Ю.Б. Проектирование баз данных в примерах и задачах. – М.: Экспресс, 2015. – 532 с.
2. Биллинг, В.А., Дехтярь М.И. VBA и Office 2016. Офисное программирование. – М.: Русская редакция, 2016. – 431 с.
3. Горев А., Макашарипов С., Ахьян Р. Эффективная работа с СУБД. – СПб: «Питер», 2016. – 704 с.

Гочияева М.Д., Жданова В.В.

Автоматизация кода Хэмминга в двоичном кодировании

*ФГБОУ ВО «Северо-Кавказская государственная академия»
(Россия, Черкесск)*

doi: 10.18411/lj-12-2019-18

idsp: ljjournal-12-2019-18

Аннотация

В статье, были рассмотрены способы реализации кодера/декодера несистематического циклического кода Хэмминга в программном виде. Программа содержит в себе алгоритмы исправления ошибок Хэмминга и справляется со своей задачей, исправляя двоичный код передаваемого сообщения.

Ключевые слова: автоматизация, кодирование, код Хэмминга.

Abstract

In the article, the ways of implementing the encoder/decoder of non-systematic cyclic Hamming code in software form were considered. The program contains Hamming error correction algorithms and copes with its task by correcting the binary code of the transmitted message.

Keywords: automation, coding, Hamming code.

Объектом исследования является самокорректирующийся код, а именно код Хэмминга. Код Хэмминга является самым известным, вероятно, первым из кодов самоанализа и самокоррекции. Используется по отношению к системе двоичных номеров. Другими словами, он шифрует, а затем обрабатывает данные, идентифицирует ошибки, например, в результате помех, и пытается восстановить поврежденные данные. Это важно для получения и отправки достоверного информационного сообщения. Код самокоррекции используется и сейчас для защиты данных, правда, все реже. На основе кода Хэмминга построены многие современные программы защиты, как при передачи данных, так и защиты памяти компьютера от непредвиденных ситуаций. Актуальность исследования состоит в том, что разработки метода получения достоверных индикаторов для надежных кодов, не зависящих от свойств канала связи и в то же время пригодных для использования со сложным комплексом, в дополнение к алгоритмам декодирования кода на основе анализа номера списка (кластера) и моделирования рендеринга программного обеспечения для этих процессов и определения потенциала различных схем декодирования, являются очень важным аспектом при создании защиты каналов связи.

Алгоритм кода Хэмминга позволяет шифровать любой информационное сообщение определенным образом и после передачи (например, по сети), чтобы определить, произошла ли ошибка в этом сообщении (например, из-за «шумов») и, если возможно, исправить ошибки этого сообщения. Следует также отметить, что в этот алгоритм входят более сложные модификации, которые позволяют обнаруживать больше ошибок. Сразу же следует сказать, что код Хэмминга состоит из двух частей. Первая часть шифрует исходное сообщение и в некоторых местах вставляет управляющие биты (вычисленные особым образом). Вторая часть принимает входящее сообщение и пересчитывает биты (в соответствии с тем же алгоритмом, что и первая часть). Если все вновь проверенные символы совпадают с полученными, то сообщение принимается без ошибок. В противном случае отображается сообщение об ошибке, и, если это возможно, исправляется. Для создания таких кодов достаточно назначить дополнительный двоичный номер каждому слову один раз и указать номер этого слова, чтобы общее число единиц любого числа, например, было нечетным. Единственная ошибка в любом количестве отправляемого слова (включая номер теста) изменит отношение общего количества единиц. Счетчики Modulo 2, которые являются числом между двоичными числами, могут выдавать сигнал ошибки. В то же время невозможно определить количество ошибок, и нет возможности исправить их. Ошибки, которые происходят одновременно в два, четыре и т. д. - в четном числе, поскольку были не замечены. Однако вряд ли будут двойные и квадратичные ошибки.

Для реализация кода Хэмминга была спроектирована на языке программирования Паскаль (Pascal) программа «Hemming».

Особенности языка Паскаль - строгая типизация и наличие инструментов структурного (процедурного) программирования. Паскаль был одним из первых таких языков. По словам Вирта, язык должен способствовать дисциплинированному программированию, поэтому, помимо строгой типизации, возможные синтаксические неоднозначности в Pascal сведены к минимуму, и сам автор попытался сделать синтаксис интуитивно понятным, даже если он впервые введен в язык. Однако изначально язык имел ряд ограничений: невозможность передачи функций массивов переменной длины, отсутствие обычных рабочих методов с динамическим хранением, ограниченная библиотека ввода-вывода, отсутствие средств для связывания функций, написанных на других языках. Отсутствие отдельных инструментов для компиляции и т. д. Некоторые из недостатков Паскаля были учтены в стандарте ISO 1982 года, и, в частности, открытые массивы отображались в языке, так что одни и те же методы можно было использовать для обработки одномерных массивов различных размеров.

Задача программы состоит в том, чтобы проверять закодированное сообщение в двоичном коде на ошибки и исправлять их. В конце выводить закодированное слово на понятный человеку язык. Спроектированная программа использует формулы по вычислению позиции ошибки в закодированном слове, которые были описаны выше. Для работы программы необязательно вводить уже заготовленное слово в двоичном коде, она принимает слова на человеческом языке, алгоритмы программы кодируют слово сами.

Рисунок 1 – Ввод данных в программу

На рисунке 1 представлено поле ввода данных в программу «Hemming». Для примера, введем слово model. В ходе работы алгоритмов программа должна вывести закодированное слово в двоичном коде с использованием уже рассчитанных контрольных битов:

Исходное слово:model
Код Хемминга:1 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0 1 1 0 0

Рисунок 2 – Кодирование сообщения

За кодирование слова отвечает программная функция, которая переводит слово по стандартам таблицы кодировки ASCII (рисунок 2). Работа алгоритма основывается сначала на подсчете количества символов в сообщении и после, посимвольно кодируется полностью все слово. В программной функции для перевода слова в двоичный код был использован циклический оператор while do – цикл с постусловием. Заголовок цикла содержит логическое выражение, если он возвращает true, тело цикла выполняется; если ложь, это не выполняется.

```
function get_word(const s:string; var i:byte):string;
var w:string;
    l:byte;
begin
    l:=length(s);
    while (i<=l) and (s[i]<=' ') do
        inc(i);
    w:='';
    while (i<=l) and (s[i]>' ') do
        begin
            w:=w+s[i];
            inc(i);
        end;
    get_word:=w;
end;
```

Рисунок 3 - Алгоритм кодирования на языке Паскаль

А за расчет контрольных битов и добавление их в закодированное слово отвечает процедура программы:

```
procedure control_bit(var a:t_arr; k,p:byte);
var i,s,t:byte;
begin
    a[p]:=0;
    i:=p;
    while i<=k do
        begin
            t:=i+p;
            while (i<t) and (i<=k) do
                begin
                    s:=s+a[i];
                    inc(i);
                end;
            i:=i+p;
        end;
    if s mod 2 = 0 then
        a[p]:=0
    else
        a[p]:=1;
    end;
```

Рисунок 4 – Расчет контрольных битов

Далее программа «Hemming» проводит проверку по формуле

$$V * H^T = S \quad (1)$$

(формула была преобразована на язык Паскаль для понимания программой необходимых действий).

Допустим, что при передаче данных возникла ошибка в 16-те
Код Хэмминга с ошибкой: 0 1 0 0 1 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0 1 1 0 0

Рисунок 5 – Код с ошибкой

В закодированном слове произошла ошибка в первом бите, а первый бит является контрольным. На рисунке ниже реализован код программы, который определяет место нахождения ошибочного бита по расчетной формуле:

```
procedure write_arr(const a:t_arr; k:byte);
var i:byte;
begin
  for i:=1 to k do
    write(a[i], ' ');
end;

function xor_ones(const a:t_arr; k:byte):byte;
var i,m:byte;
begin
  i:=1;
  while a[i]<>1 do
    inc(i);
  m:=i;
  for i:=i+1 to k do
    if a[i]=1 then
      m:=m xor i;
  xor_ones:=m;
end;
```

Рисунок 6 – Проверка закодированного слова на ошибки

И заключительной стадией работы программы «Hamming» является устранение ошибочного бита и сравнение полученного кода с исходным. В случае, если полученный код совпадает с исходным, то программа выводит его, если нет, то прекращает свою работу.

Исправленный код Хэмминга: 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0 1 1 0 0

Ввод данных:

Окно вывода | Список ошибок | Сообщения компилятора

Рисунок 7 – Исправленный код Хэмминга

После обнаружения ошибочного бита выполняется процедура изменения, т.е. ошибочный бит конвертируется.

```
procedure change_bit(var a:t_arr; n:byte);
begin
  if a[n]=0 then
    a[n]:=1;
  else
    a[n]:=0;
end;
```

Рисунок 8 – Конвертация ошибочного бита

Программа «Hamming» полностью выполняет поставленную задачу. Ошибка в коде была найдена и исправлена, код отлажен и работает исправно. Код Хэмминга используется в некоторых приложениях для хранения данных, особенно в RAID 2.

Кроме того, метод Хемминга давно используется в памяти ЕСС и позволяет исправлять одноразовые ошибки и выявлять двойные ошибки на лету.

В работе, были рассмотрены способы реализации кодера/декодера несистематического циклического кода Хэмминга в программном виде. Программа содержит в себе алгоритмы исправления ошибок Хэмминга и справляется со своей задачей, исправляя двоичный код передаваемого сообщения. Код Хэмминга все еще используется на практике, но не для передачи информации, а для защиты памяти компьютеров. Чтобы отладить передачу сообщения в бинарном виде, нужно исправить ошибки, и не одну, десятки, сотни или даже тысячи ошибок в соответствии с длиной кода.

1. Бояринов, И. М. Самопроверяющиеся схемы и алгоритмы декодирования двоичных кодов Хэмминга, БЧХ-кодов и кодов Рида-Соломона над $GF(2^m)$ / И. М. Бояринов // Проблемы передачи информации. – Т. 44, вып. 2. – М.: Техносфера, 2008. – 297 с.
2. Васильев, К. К. Математическое моделирование систем связи : учебное пособие / К. К. Васильев, М. Н. Служивый. – 2-изд., перераб. и доп. – Ульяновск : УлГТУ, 2010. – 170 с.
3. Кларк, Дж., мл., Кейнс, Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. – М.: Радио и связь, 2014. – 392 с.