# СИММЕТРИЧНЫЕ КОДЫ ПЕРЕМЕННОЙ ДЛИНЫ ДЛЯ СЖАТИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

*Учреждение образования «Белорусский государственный технологический университет», г. Минск, Республика Беларусь, доцент, кандидат технических наук, доцент*

Коды переменной длины принято обозначать как Variable Length Coding (VLC). К данной категории относится Universal Variable Length Code (UVLC), обеспечивающий альтернативный способ кодирования (ALT). UVLC может строиться несколькими способами: с использованием длины последнего кодового блока, посредством динамического переупорядочения символов DSR (Dynamic Symbol Reordering), применяемого при небольшом размере алфавита источника, а также на основе дополнительной конфигурации кодов ACC (Additional Code Configuration).

В рамках данной работы используется RVLC (Reversible Variable Length Code) – обратимый код переменной длины, относящийся к семейству VLC. По эффективности кодирования RVLC сопоставим с другими кодами переменной длины, однако отличается от них симметричной структурой кодовых слов. Благодаря этому каждое слово может считываться как в прямом, так и в обратном направлении. Такая особенность позволяет восстанавливать больший объем данных при возникновении ошибок в процессе передачи или хранения файла по сравнению с традиционными VLC-кодами. Кроме того, RVLC удобно реализовывать программно с использованием стандартной структуры данных «очередь» и базовых операций над ней.

Алгоритмы декодирования должны исключать неоднозначность интерпретации кодовой последовательности. В разработанном алгоритме это обеспечивается соблюдением принципов префиксности и симметричности. Корректно сформированные RVLC-коды соответствуют обоим требованиям.

Процесс декодирования заключается в последовательном побитовом чтении закодированной информации. На каждом этапе проверяется, является ли сформированная последовательность симметричной. Если условие выполняется, полученная комбинация сопоставляется со словарем кодовых слов, после чего распознанная последовательность удаляется из обрабатываемого потока. Описанные действия повторяются до тех пор, пока не будет обработан весь закодированный файл.

Для иллюстрации рассмотрим последовательность «0001011», полученную в результате RVLC-кодирования в предыдущем примере. Первый бит считывается отдельно, поскольку минимальная длина кодового слова равна двум битам. После добавления второго бита формируется строка «00». Она является симметричной, поэтому выполняется поиск соответствующего слова в словаре. Устанавливается, что комбинация «00» соответствует символу a. В потоке остается последовательность «01011».

Далее считываются два первых бита – «01». Полученная комбинация не обладает симметрией, поэтому к ней добавляется следующий бит. Последовательность «010» уже является симметричной и, согласно словарю, соответствует символу c. После этого считываются следующие два бита – «11». Данная комбинация также симметрична и соответствует символу b. Таким образом, результатом декодирования последовательности «0001011» является строка «acb».

Рассмотрим реализацию декодирования в разработанном программном обеспечении. Процесс запускается после возникновения события нажатия кнопки «Декодировать». Сначала из закодированного файла извлекаются параметры, необходимые для последующей обработки: минимальная длина кодового слова и словарь кодовых комбинаций. Полученные сведения преобразуются и сохраняются в списке структур MSWord. В результате формируется словарь в формате, удобном для использования программой [1, 2].

После подготовки всех необходимых данных начинается непосредственное декодирование. Вначале из файла считывается один байт, содержащий информацию о начале блока и количестве битов, занятых данными этого блока. На основании указанного значения вычисляется количество байтов, которое необходимо дополнительно прочитать. Считанные байты преобразуются в битовую строку, после чего обрабатываются в соответствии с описанным алгоритмом декодирования.

Последовательно выполняются три операции: считывание заголовка блока, чтение содержащихся в нем данных и их декодирование. Данный цикл повторяется до достижения конца закодированного файла.

Симметричная структура RVLC позволяет также частично восстанавливать информацию при наличии ошибок внутри блока. Если при последовательном декодировании в прямом направлении обнаруживается ошибка, обработка блока продолжается в обратную сторону. В случае возникновения ошибки при обратном

декодировании уже распознанные данные записываются в выходной файл, а оставшаяся необработанная часть блока отбрасывается. Благодаря этому удается сохранить часть информации, расположенную как в начале, так и в конце поврежденного блока.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Жиляк, Н. А. Безопасная форма хранения паролей в базе данных / Н. П. Цыганенко, Н. А. Жиляк // Материалы XVI Республ. науч. Конф. студентов и аспирантов (Гомель, 25–27 марта 2013 года) – С. 147–148.

2. Жиляк, Н. А. Шифрование отдельных элементов в формате TEXML / Н. А. Жиляк, Н. П. Цыганенко // 64-я НТК студентов и магистрантов: сб. науч. работ: в 3-х ч. – Минск : БГТУ, 2013. – Ч. 3 – С. 381–383.