Коррекция данных при декомпрессии поврежденных архивов

Data correction in decompression of damaged archives

Ключевые слова: декомпрессия – decompression; архив – archive; искажение – corruption, код источника – source code; контекст – context; коррекция – correction.

В статье рассматривается проблема декомпрессии поврежденных архивов. Представлено аналитическое описание процедур сжатия и декомпрессии информации. Разработан алгоритм коррекции искажений при декомпрессии поврежденных архивов.

This article deals with the problem of damaged archive decompression. It provides an analytical description of the compression and decompression procedures. An algorithm for corrupted data correction in decompression of damaged archives is developed.

ПРОНКИН / PRONKIN A.
Алексей Александрович
(pron_rzhew@mail.ru)
Академия ФСО России,
адъюнкт.
г. Орел

АНАЛИТИЧЕСКОЕ ОПИСАНИЕ ОПЕРАЦИИ СЖАТИЯ СООБЩЕНИЯ

Суть методов сжатия информации состоит в представлении последовательности символов из алфавита источника сообщений как можно меньшим числом бит. Обобщенная структурная схема формирования кодовой последовательности представлена на рис. 1.

Пусть: $A = \{a_0, a_1, ..., a_{(N_A-1)}\}$ – алфавит некоторой последовательности символов, $S = \{s_0, s_1, ..., s_{(N_S-1)}\}$, $s_i \in A$ – некоторая последовательность (строка) символов, c^H – множество кодовых слов Хаффмана, $c^{LZ} = \{c^s, c^o, c^l\}$ – множество кодовых слов LZSS, c^S – кодовое отображение LZSS символа $s \in A$, c^l – кодовое отображение LZSS длины $l_s = j_2 - j_1$ подстроки $s_{j_1}^{j_2}$, c^o – кодовое отображение LZSS смещения $s_{j_1}^{j_2}$ в буфере поиска LZSS.

В процессе сжатия компрессор осуществляет преобразование входной последовательности символов $s_0, s_1, s_2, ... \in A$, поступающей от источника сообщений, в последовательность кодовых слов $c_0^H, c_1^H, c_2^H, ...$. Для получения последовательности различных подстрок компрессором выполняется грамматический разбор входного потока на отдельные подстроки [1].

Определение 1. Подпоследовательностью (подстрокой) последовательности S называется любая подпоследовательность символов $s_{j_1}^{j_2}$, $1 \le j_1 \le j_2 \le N_S$, ограниченной длины $l_s = j_2 - j_1$, принадлежащая строке S.

В данной статье рассматривается только сжатие LZSS. Основная идея метода LZSS состоит в использовании ранее прочитанной части входного файла в качестве словаря (рис. 2), где SW – скользящее окно, SB – буфер поиска, PB – упреждающий буфер, PC – текущая позиция указателя кодера. Очевидно, что $N_{SW} = N_{SB} + N_{PB}$.

Формально процедуру кодирования можно представить следующим выражением (1):

$$C^{LZ} = \begin{cases} C^{LZ} \bigcup [s_{k+N_{SB}}^{k+N_{SB}} \underset{\text{code}}{\rightarrow} c_i^s], & s_{k+N_{SB}}^{k+N_{SB}+l_s} \notin SB, l_s = 0 \\ & s_{k+N_{SB}}^{k+N_{SB}+l_s} \in SB, 0 \le l_s < 3 \\ C^{LZ} \bigcup \{[l_s \underset{\text{code}}{\rightarrow} c_i^l], [\text{ofs}(SB, s_{k+N_{SB}}^{k+N_{SB}+l_s}) \underset{\text{code}}{\rightarrow} c_{i+1}^o\}, & \\ s_{k+N_{SB}}^{k+N_{SB}+l_s} \in SB, \ 3 \le l_s \le N_{PB} & \end{cases}, \quad (1)$$

где $x \underset{\text{code}}{\rightarrow} y$ – операция отображения символа x в кодовую комбинацию y, $\text{ofs}(Y, y_{j1}^{j2})$ – процедура поиска подстроки y_{j1}^{j2} размером $l_s = j_2 - j_1$ символов в строке Y, $\bigcup$ – операция добавления элемента (нескольких элементов) в множество.

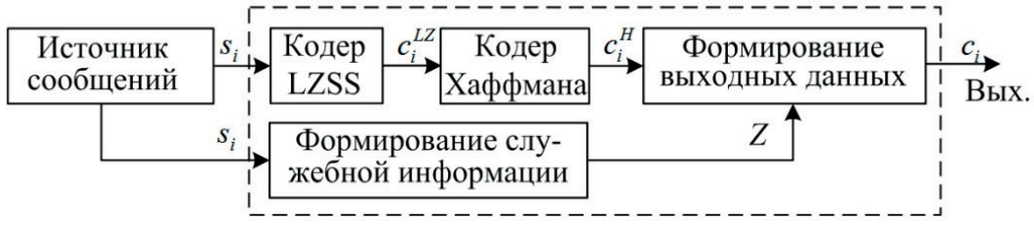

Рис. 1. Обобщенная структурная схема формирования сжатой последовательности

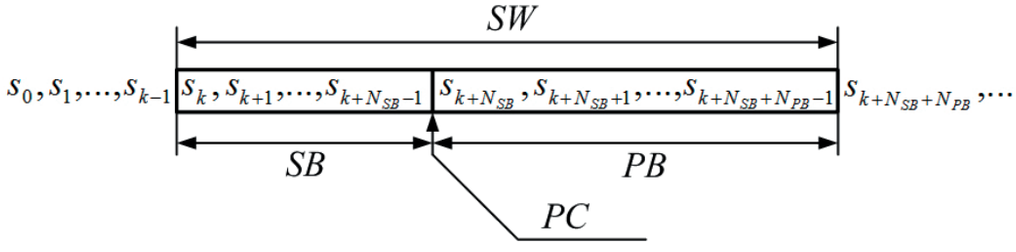

Рис. 2. Скользящее окно кодера LZSS

Если i – текущая итерация кодирования подстроки $s_{k_i+N_{SB}}^{k_i+N_{SB}+N_{PB}-1}$ в скользящем окне $SW = \{s_{k_i}^{k_i+N_{SB}+N_{PB}-1}\}$, тогда последующая ($i$+1)-ая итерация (сдвиг скользящего окна) будет определяться выражением (2):

$$k_{i+1} = \begin{cases} k_i + 1, \ 0 \le l_s < 3 \\ k_i + l_s, \ 3 \le l_s \le N_{PB} \end{cases}, \quad (2)$$

где k – позиция скользящего окна SW в строке S. Процедуру преобразования кодовых слов LZSS в кодовые слова Хаффмана можно представить выражением [2]:

$$C^H = C^{LZ} \underset{\text{code}}{\rightarrow} C^H.$$

АНАЛИТИЧЕСКОЕ ОПИСАНИЕ ОПЕРАЦИИ ДЕКОМПРЕССИИ СООБЩЕНИЯ

В каждый момент времени i=1, 2, ... на вход декомпрессора поступает один бит. Сформировав из нескольких подряд следующих бит кодовое слово, декомпрессор декодирует его $c^H \underset{\text{decode}}{\rightarrow} c^{LZ} \underset{\text{decode}}{\rightarrow} s_{j1}^{j2}$ и выдает на выход один или несколько символов (количество символов – $l_s = j_2 - j_1$) алфавита источника сообщений. При этом, в силу особенностей алгоритма компрессии LZSS, на вход декомпрессора в каждый момент времени поступает либо c^s, либо $\{c^l, c^o\}$. Процесс декодирования принятой кодовой последовательности $C = \{c_0^s, c_1^s, \ldots, c_{i-1}^l, c_i^o, c_{i+1}^s, \ldots\}$ можно представить в следующем виде (рис. 3).

Формально процедуру декодирования принятой кодовой последовательности $\hat{C}$ можно представить следующим выражением (3):

$$S = \begin{cases} S \bigcup [c_{i+1}^s \underset{\text{decode}}{\rightarrow} s_{k+N_{SB}}^{k+N_{SB}}], l_s = 0 \\ S \bigcup [\{c_{i-1}^l, c_i^o\} \underset{\text{decode}}{\rightarrow} s_{k+o}^{k+o+l_s}], \ s_{k+o}^{k+o+l_s} \in SB, \\ l_s \ge 3 \end{cases}, \quad (3)$$

По аналогии с выражением (3), сдвиг скользящего окна будет определяться выражением (4):

$$k_{i+1} = \begin{cases} k_i + 1, \ l_s = 0 \\ k_i + l_s, \ l_s \ge 3 \end{cases}, \quad (4)$$

где k – позиция буфера поиска SB в строке S [2, 3].

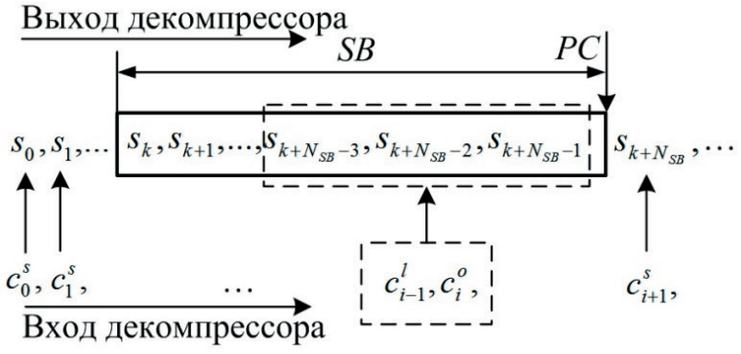

Рис. 3. Процесс декодирования принятой кодовой последовательности

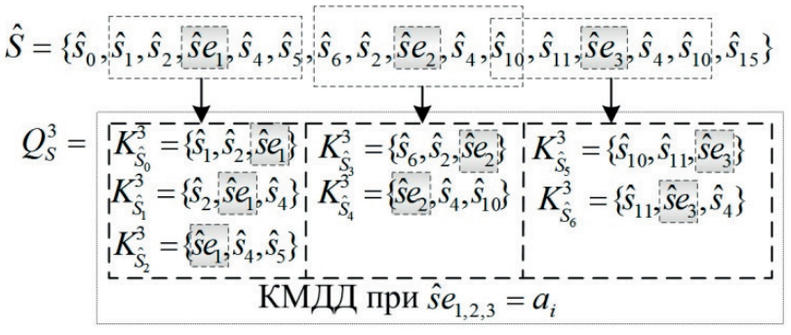

Рис. 4. Построение контекстной модели декодированных данных после декомпрессии кода источника

КОРРЕКЦИЯ ИСКАЖЕНИЙ ПРИ ДЕКОМПРЕССИИ СООБЩЕНИЯ

При передаче по дискретному каналу в результате воздействия помех кодовая последовательность C может быть искажена: $C \to \hat{C}$. Если искажений нет и $\hat{c}_i = c_i$ для всех $1 \le i \le N$, тогда в результате операции декодирования получаем $\hat{s}_i = [\hat{c}_i \underset{\text{decode}}{\to} s_i] = s_i$ для всех i, где $[\hat{c}_i \underset{\text{decode}}{\to} s_i]$ – процедура декодирования символа s_i. Таким образом, сообщение воспроизводится в точном соответствии с оригиналом. В условиях воздействия помех возникают искажения, которые оказывают существенное влияние на декодированное сообщение $\hat{S}$, при этом существуют i, при которых $\hat{s}_i = [\hat{c}_i \underset{\text{decode}}{\to} s_i] \ne s_i$.

Пусть:

– $A = \{a_0, a_1, a_2, \ldots\}$ – алфавит источника;

– $S = \{s_0, s_1, s_2, \ldots\}$ $s_i \in A$ – исходное сообщение;

– $S_O = \{so_0, so_1, so_2, \ldots\}$, $so_i \in A$, $S_O \ne S$ – сообщение-эталон (словарь);

– $K^m_{S_0} = \{s_0, s_1, \ldots, s_{m-1}\}$, $K^m_{S_1} = \{s_1, s_2, \ldots, s_m\}$, ... – контексты (m-граммы) сообщения S;

– $K^m_{SO_0} = \{so_0, so_1, \ldots, so_{m-1}\}$, $K^m_{SO_1} = \{so_1, so_2, \ldots, so_m\}$, ... – контексты (m-граммы) сообщения S_O;

– $Q^m_{SO} = \{K^m_{SO_0}, K^m_{SO_1}, K^m_{SO_2}, \ldots\}$ – множество неповторяющихся контекстов сообщения S_O;

– $Q^m_S = \{K^m_{S_0}, K^m_{S_1}, K^m_{S_2}, \ldots\}$ – множество неповторяющихся контекстов сообщения S;

Процесс коррекции искажений предлагается рассмотреть на примере. Исходное сообщение S представлено выражением (5) при условиях (6):

$$S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}, \quad (5)$$

$$s_2^4 = s_7^9, \; s_8^{10} = s_{12}^{14}. \quad (6)$$

С учетом выражений (5) и (6) после сжатия сообщения S кодовая последовательность на выходе кодера представлена выражением (7):

$$C = \{c_0^{s=s0}, c_1^{s=s1}, c_2^{s=s2}, c_3^{s=s3}, c_4^{s=s4}, c_5^{s=s5}, c_6^{s=s6}, [c_7^{l=3}, c_8^{o=5}], c_9^{s=s10}, c_{10}^{s=s11}, [c_{11}^{l=3}, c_{12}^{o=4}], c_{13}^{s=s15}\}. \quad (7)$$

При передаче кодовой последовательности (7) по дискретному каналу в результате воздействия помех возникает ошибка, представленная выражением (8):

$$\hat{C} = \{\hat{c}_0^{s=s0}, \hat{c}_1^{s=s1}, \hat{c}_2^{s=s2}, \boxed{\hat{c}e_3^{s=sel}}, \hat{c}_4^{s=s4}, \hat{c}_5^{s=s5}, \hat{c}_6^{s=s6}, [\hat{c}_7^{l=3}, \hat{c}_8^{o=5}], \hat{c}_9^{s=s10}, \hat{c}_{10}^{s=s11}, [\hat{c}_{11}^{l=3}, \hat{c}_{12}^{o=4}], \hat{c}_{13}^{s=s15}\} \quad (8)$$

Ошибка

В результате декодирования кодовой последовательности (8) сообщение, представленное выражением (9), воспроизводится с искажениями:

$$\hat{S} = \{\hat{s}_0, \hat{s}_1, \hat{s}_2, \boxed{\hat{s}e_1}, \hat{s}_4, \hat{s}_5, \hat{s}_6, \hat{s}_2, \boxed{\hat{s}e_2}, \hat{s}_4, \hat{s}_{10}, \hat{s}_{11}, \boxed{\hat{s}e_3}, \hat{s}_4, \hat{s}_{10}, \hat{s}_{15}\} \quad (9)$$

где $\hat{s}e_1, \hat{s}e_2, \hat{s}e_3$ – искаженные (неопределенные) символы из-за воздействия ошибки и влияния словаря LZSS (эффект размножения ошибок).

Процесс коррекции ошибок заключается в построении контекстной модели декодированного сообщения (9), подстановки вместо неизвестных (искаженных) символов символы из алфавита источника методом полного перебора и проверки построенной контекстной модели на запрещенные комбинации путем сравнения с контекстной моделью сообщения-эталона (рис. 4).

Пусть N_A – размер алфавита источника и $0 \le i < N_A$, тогда если существует символ $a_i = \hat{s}e_{1,2,3}$ такой, что выполняется условие(10):

$$K^m_{\hat{S}_j} \in Q^m_{SO}, \quad (10)$$

где $0 \le j < N_{QS}$, N_{QS} – размер Q^m_S, $a_i \in A$.

При этом в данном случае (рис. 4), если выполняется условие (10), то, в зависимости от количества пересечений множеств Q^3_S и Q^3_{SO},возможно несколько вариантов коррекции ошибок в сообщении $\hat{S}$:

$$-0 \le j \le 6 \Rightarrow \boxed{\hat{s}e_1 = \hat{s}e_2 = \hat{s}e_3 = a_i},$$

т.е. все семь контекстов сообщения $\hat{S}$ содержатся в контекстной модели сообщения-эталона $K^m_{\hat{S}_0}, K^m_{\hat{S}_1}, ..., K^m_{\hat{S}_6} \in Q^m_{SO}$, следовательно, могут быть исправлены все три ошибки;

$$-0 \le j \le 4 \Rightarrow \boxed{\hat{s}e_1 = \hat{s}e_2 = a_i}, \hat{s}e_3 = ?,$$

т.е. контекст $K^m_{\hat{S}_5}$ и (или) $K^m_{\hat{S}_6} \notin Q^m_{SO}$, следовательно, могут быть исправлены две ошибки;

$$-0 \le j \le 2 \Rightarrow \boxed{\hat{s}e_1 = a_i}, \hat{s}e_2 = ?, \hat{s}e_3 = ?,$$

т.е. контексты $K^m_{\hat{S}_0}, K^m_{\hat{S}_1}, K^m_{\hat{S}_2} \in Q^m_{SO}$, а остальные являются запрещенными комбинациями для Q^m_{SO}, следовательно, может быть исправлена только одна ошибка.

Если условие (10) не выполняется, т.е. $K^m_{\hat{S}_0}, K^m_{\hat{S}_1}, ..., K^m_{\hat{S}_6} \notin Q^m_{SO}$, то ошибки в сообщении не могут быть исправлены [4].

Таким образом, в данной статье получены выражения для формального описания процедур кодирования и декомпрессии информации, а так же рас-смотрен алгоритм коррекции искажений при декомпрессии сообщения, учитывающий особенности формирования сжатой кодовой последовательности и позволяющий исправлять ошибки в декодированном сообщении.

Литература

1. *Пронкин, А.А.* Восстановление искаженных сжатых сообщений // Интернет-журнал "Науковедение". – 2014. – № 1 (20) [Электронный ресурс]. – М. : Науковедение, 2014. – Режим доступа: http://naukovedenie.ru, свободный. – Загл. с экрана.

2. *Ziv J.,* Lempel A. A Universal Algorithm for Sequential Data Compression // IEEE Transaction on Information Theory, 1977, vol.23, №3. – P.337–343.

3. *Ralf D. Brown.* Reconstructing corrupt DEFLATEd files // Digital Investigation. – 2011. – № 8. – P. 125–131.

4. Патент 2510957 Российская федерация, МПК7 H 03 M 007/40. Способ восстановления искаженных сжатых файлов [Текст] / Пронкин А. А. [и др.]; заявитель и патентообладатель Академия ФСО России. – № 2012155650/08; заявл. 25.12.2012 ; опубл. 10.04.2014, Бюл. № 10. – 10 с.: ил.