Математика и её приложения: ЖИМО. 2009. Вып. 1 (6). С. 3 – 10.

УДК 512.54

М. Е. Гришко[1]

Быстрый алгоритм кодирования и декодирования поля кратности и поля порядка одним числом при булевом сжатии файлов

Ключевые слова: булева алгебра, поле порядка, поле кратности.

Предложен алгоритм, позволяющий кодировать поле кратности и поля порядка не отдельно, а вместе одним числом. Для этого найден способ построения упорядоченной таблицы, содержащей все возможные типы повторов и их перестановки. Таблицу можно разбить на блоки. Это позволяет решать задачу кодирования путем вычисления длин блоков, предшествующих искомой строке. Основным преимуществом данного алгоритма является его быстрота. Также решена обратная задача: по номеру в упорядоченной таблице восстановить декодируемую строку.

We suggest the algorithm, which allows to code a multiplicity field and an order field by one number. We suggest the method for construction of the table, which contains all posible repetition types and their permutations. The table is devided on blocks; so the problem of coding is solved by calculation of block length that precedes to required line; this algorithm is quick. The return task is solved also: for any number in the ordered table we can restore an uncoding line.

В работе [1] к сжатию дискретной информации без потерь предложен подход, основанный на использовании булевых уравнений. Данный подход подразумевает разбиение исходного файла на кортежи по n битов и последующее объединение их в буферы. Код сжимаемого таким образом файла будет содержать три поля:

• поле принадлежности, состоящее из коэффициентов булева полинома и дающее информацию о том, какие из 2^n возможных кортежей входят в кодируемый буфер (алгоритм кодирования и декодирования поля принадлежности построен в работе [6]);

• поле кратности, содержащее числа повторов кортежей, входящих в буфер;

• поле порядка, дающее информацию о том, как кортежи расставлены в исходном буфере с учетом их кратности.

Алгоритму кодирования поля кратности посвящены работы [2, 4, 5, 9, 10]. Структура и процесс кодирования поля порядка описаны в работах [3, 8].

В [7] рассмотрена возможность кодирования поля кратности и поля порядка совместно одним числом, а не по отдельности, как предлагалось ранее.

Настоящая работа посвящена реализации данной возможности.

[1]Ивановский государственный университет; E-mail: mgrishko_37@mail.ru. Работа выполнена при финансовой поддержке РФФИ (проект 07-07-00155).

1. Постановка задачи

Рассмотрим исходный файл, разбитый на кортежи по n битов. Объединим кортежи последовательно по m штук. Данный набор кортежей будем называть буфером.

Для примера рассмотрим один буфер. Среди входящих в него кортежей могут быть повторяющиеся. Обозначим число повторяющихся кортежей через s. Очевидно, что выполнено соотношение

$$1 < s \leqslant m. \tag{1}$$

Рассмотрим последовательность n_1, n_2, …, n_s, где n_k — число, определяющее, сколько раз входит k-й кортеж в рассматриваемый нами буфер. Назовем ее *типом повторов*. Очевидно, справедливо равенство

$$\sum_{k=1}^{s} n_k = m. \tag{2}$$

Все типы повторов, возможные для заданного буфера, можно упорядочить в таблицу (3). Структура таблицы подробно описана в работах [9, 10].

Кроме поля кратности в код сжимаемого файла необходимо включить информацию не только о том, какие кортежи входят в буфер и в каком количестве, но и о том, как они расположены внутри рассматриваемого буфера. Для этого служит поле порядка. Оно представляет собой совокупность всех возможных перестановок чисел кратности для каждого типа повторов, входящего в таблицу (3). Назовем последовательность j_1, j_2, …, j_m, в которой s элементов различны, *перестановкой*.

В настоящей работе реализована возможность построения упорядоченной таблицы, содержащей и поле кратности, и поле порядка. Таким образом, можно сформулировать задачи, которые будут решаться:

• задача кодирования, заключающаяся в нахождении номера в упорядоченной таблице, соответствующего кодируемому буферу;

• задача декодирования, состоящая в необходимости поставить в соответствие номеру в упорядоченной таблице всех возможных типов повторов и их перестановок искомый буфер.

2. Структура таблицы

Таблица, содержащая все возможные типы повторов для заданного буфера, а также все их перестановки, однозначно определяется числами m и s. В работе предложен принцип построения, подразумевающий структурное деление таблицы на блоки. Таблица будет иметь следующий вид.

I. $n_k = 1$, $n_s = m - s + 1$, $k = 1, 2, \ldots, s - 1$;

1) $j_i = 1$, $i = 1, \ldots, n_1$,
$j_i = 2$, $i = n_1 + 1, \ldots, n_1 + n_2$,
$j_i = 3$, $i = n_1 + n_2 + 1, \ldots, n_1 + n_2 + n_3$,
…

2) … (3)

II. $n_k = 1$, $n_{s-1} = 2$, $n_s = m - s$, $k = 1, 2, \ldots, s - 2$,
1) ...
...
C_{m-1}^{s-1}. $n_1 = m - s + 1$, $n_k = 1$, $k = 2, \ldots, s$,
1) ...

Чтобы была понятна структура таблицы, приведем малоразмерный пример. Пусть m и s равны соответственно 5 и 3.

Таблица 1. Типы повторов для $m = 5$ и $s = 3$

I. 1 + 1 + 3

1. 1+2+3+3+3	8. 2+3+3+3+1	15. 3+3+1+2+3
2. 1+3+2+3+3	9. 3+1+2+3+3	16. 3+3+1+3+2
3. 1+3+3+2+3	10. 3+1+3+2+3	17. 3+3+2+1+3
4. 1+3+3+3+2	11. 3+1+3+3+2	18. 3+3+2+3+1
5. 2+1+3+3+3	12. 3+2+1+3+3	19. 3+3+3+1+2
6. 2+3+1+3+3	13. 3+2+3+1+3	20. 3+3+3+2+1
7. 2+3+3+1+3	14. 3+2+3+3+1	

II. 1 + 2 + 2

21. 1+2+2+3+3	31. 2+2+3+1+3	41. 3+1+3+2+2
22. 1+2+3+2+3	32. 2+2+3+3+1	42. 3+2+1+2+3
23. 1+2+3+3+2	33. 2+3+1+2+3	43. 3+2+1+3+2
24. 1+3+2+2+3	34. 2+3+1+3+2	44. 3+2+2+1+3
25. 1+3+2+3+2	35. 2+3+2+1+3	45. 3+2+2+3+1
26. 1+3+3+2+2	36. 2+3+2+3+1	46. 3+2+3+1+2
27. 2+1+2+3+3	37. 2+3+3+1+2	47. 3+2+3+2+1
28. 2+1+3+2+3	38. 2+3+3+2+1	48. 3+3+1+2+2
29. 2+1+3+3+2	39. 3+1+2+2+3	49. 3+3+2+1+2
30. 2+2+1+3+3	40. 3+1+2+3+2	50. 3+3+2+2+1

III. 1 + 3 + 1

51. 1+2+2+2+3	58. 2+2+1+2+3	65. 2+3+2+1+2
52. 1+2+2+3+2	59. 2+2+1+3+2	66. 2+3+2+2+1
53. 1+2+3+2+2	60. 2+2+2+1+3	67. 3+1+2+2+2
54. 1+3+2+2+2	61. 2+2+2+3+1	68. 3+2+1+2+2
55. 2+1+2+2+3	62. 2+2+3+1+2	69. 3+2+2+1+2
56. 2+1+2+3+2	63. 2+2+3+2+1	70. 3+2+2+2+1
57. 2+1+3+2+2	64. 2+3+1+2+2	

IV. 2 + 1 + 2

71. 1+1+2+3+3	81. 1+3+3+1+2	91. 3+1+2+1+3
72. 1+1+3+2+3	82. 1+3+3+2+1	92. 3+1+2+3+1
73. 1+1+3+3+2	83. 2+1+1+3+3	93. 3+1+3+1+2
74. 1+2+1+3+3	84. 2+1+3+1+3	94. 3+1+3+2+1
75. 1+2+3+1+3	85. 2+1+3+3+1	95. 3+2+1+1+3
76. 1+2+3+3+1	86. 2+3+1+1+3	96. 3+2+1+3+1
77. 1+3+1+2+3	87. 2+3+1+3+1	97. 3+2+3+1+1
78. 1+3+1+3+2	88. 2+3+3+1+1	98. 3+3+1+1+2
79. 1+3+2+1+3	89. 3+1+1+2+3	99. 3+3+1+2+1
80. 1+3+2+3+1	90. 3+1+1+3+2	100.3+3+2+1+1

V. 2 + 2 + 1

101. 1+1+2+2+3	111. 1+3+2+1+2	121. 2+2+3+1+1
102. 1+1+2+3+2	112. 1+3+2+2+1	122. 2+3+1+1+2
103. 1+1+3+2+2	113. 2+1+1+2+3	123. 2+3+1+2+1
104. 1+2+1+2+3	114. 2+1+1+3+2	124. 2+3+2+1+1
105. 1+2+1+3+2	115. 2+1+2+1+3	125. 3+1+1+2+2
106. 1+2+2+1+3	116. 2+1+2+3+1	126. 3+1+2+1+2
107. 1+2+2+3+1	117. 2+1+3+1+2	127. 3+1+2+2+1
108. 1+2+3+1+2	118. 2+1+3+2+1	128. 3+2+1+1+2
109. 1+2+3+2+1	119. 2+2+1+1+3	129. 3+2+1+2+1
110. 1+3+1+2+2	120. 2+2+1+3+1	130. 3+2+2+1+1

VI. 3 + 1 + 1

131. 1+1+1+2+3	138. 1+2+1+3+1	145. 2+1+3+1+1
132. 1+1+1+3+2	139. 1+2+3+1+1	146. 2+3+1+1+1
133. 1+1+2+1+3	140. 1+3+1+1+2	147. 3+1+1+1+2
134. 1+1+2+3+1	141. 1+3+1+2+1	148. 3+1+1+2+1
135. 1+1+3+1+2	142. 1+3+2+1+1	149. 3+1+2+1+1
136. 1+1+3+2+1	143. 2+1+1+1+3	150. 3+2+1+1+1
137. 1+2+1+1+3	144. 2+1+1+3+1	

На приведенном примере хорошо видна структура таблицы. Теперь подробнее объясним принцип построения. Вся таблица делится на блоки различного уровня. Обратим внимание на блоки, пронумерованные римскими цифрами. Если рассматривать их отдельно, то они представляют собой совокупность всех возможных типов повторов для заданных изначально m и s. Каждый такой блок распадается на отдельную таблицу, представляющую собой все перестановки соответствующего типа повторов. Нетрудно заметить, что среди всех перестановок также можно провести разбиение на подблоки. Так первое разделение даст подблоки, различающиеся значением j_1. Также каждый такой подблок можно разделить на части в соответствии со значениями j_2 и т. д. Основываясь на предложенном структурном построении, в следующем разделе мы найдем формулу для вычисления длины подблоков.

3. Вычисление длин блоков

Для вычисления длины всей таблицы, а также длин отдельных блоков воспользуемся формулами комбинаторики.

В работе [9] приведено выражение для вычисления количества блоков, обозначенных выше римскими цифрами. Из [3] воспользуемся формулой, определяющей количество возможных перестановок отдельного типа повторов. Таким образом, несложно увидеть, что длина таблицы (3) будет определяться с помощью выражения

$$N = \sum_{i=1}^{C_{m-1}^{s-1}} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!}\,. \tag{4}$$

Если вычислить длину таблицы при $m = 5$, $s = 3$ по формуле (4), то полученный результат в точности совпадет с приведенным выше примером. Таким образом, теперь мы можем вычислить длину необходимого количества блоков, пронумерованных римскими цифрами, изменяя пределы у суммы в формуле (4).

Далее подробнее рассмотрим структуру блоков, где нумерация произведена арабскими цифрами. Обратим внимание на столбец j_1. В нем можно выделить подблоки, начинающиеся с $j_1 = 1$, $j_1 = 2$ и т. д. Если зафиксировать j_1 и рассмотреть соответствующий ему подблок, то несложно заметить, что в нем аналогично производится деление на более мелкие подблоки в соответствии со значением j_2. Таким образом, деление на подблоки можно производить до тех пор, пока подблок не будет представлять собой единственную перестановку.

Теперь определим способ вычисления длины подблоков, начинающихся с определенного числа фиксируемых j_i. Введем обозначения:

L_{j_1} – длина подблока с фиксированным значением j_1;

$L_{j_1 j_2}$ – длина подблока с фиксированным значением j_1 и j_2;

...

Тогда длина подблока с p фиксированными первыми элементами перестановки будет обозначаться как $L_{j_1 j_2 \ldots j_p}$. Очевидно, что среди фиксированных j_i могут быть повторяющиеся. Учитывая данную возможность, введем обозначение числа кратности фиксированного элемента как n_k^f. Например, для подблока вида $L_{31123432}$ кратности j_i (i пробегает значения от 1 до 4 (столько различных элементов зафиксировано для данного подблока)) будут равны соответственно n_1^f=2, n_2^f=2, n_3^f=3, n_4^f=1.

Таким образом, с учетом принятых обозначений, длину подблока вида $L_{j_1 j_2 \ldots j_p}$ можно вычислять с помощью формулы

$$L_{j_1 j_2 \ldots j_p} = \frac{(m-p)!}{\prod\limits_{k=1}^{s} n_k'!}, \tag{5}$$

где n_k'=$n_k - n_k^f$.

4. Алгоритм кодирования

Как было сказано выше, задача кодирования заключается в том, чтобы по заданному буферу определить его номер в таблице (3).

Итак, пусть задан вид буфера, код которого нужно найти. Это значит, что нам известна исходная перестановка кортежей. Обозначим ее j_1^*, j_2^*, ..., j_m^*. Далее по этим данным вычисляем m, s, а также тип повторов, к которому принадлежит исходная перестановка. Чтобы найти номер j_1^*, j_2^*, ..., j_m^* в таблице (3), необходимо вычислить длину всех блоков, предшествующих типу повторов, к которому принадлежит исходная перестановка, и количество строк, предшествующих самой перестановке в подблоке для данного типа повторов.

Для решения поставленной задачи применим формулу

$$N = \sum_{i=1}^{N^*-1} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} + \sum_{j_1=1}^{j_1^*-1} L_{j_1} + \sum_{j_2=1}^{j_2^*-1} L_{j_1^* j_2} + \cdots + \sum_{j_{m-1}=1}^{j_{m-1}^*-1} L_{j_1^* j_2^* \ldots j_{m-2}^* j_{m-1}} + 1, \quad (6)$$

где N^* – номер типа повторов, содержащего исходную перестановку. Последовательность n'_k, где $k = 1, 2, \ldots, s$, для каждой суммы вычисляется отдельно. Заметим, что j_i может принимать только такие значения, которые удовлетворяют условию: $n_k^f \leq n_k$. Также следует учесть, что если верхний предел суммы оказывается меньше нижнего, то сумму следует считать равной нулю.

Произведя все вышеописанные вычисления, мы получим номер исходной перестановки в таблице (3).

5. Алгоритм декодирования

Задача декодирования заключается в следующем: по заданным m, s, а также номеру $\tilde{N}$ в упорядоченной таблице (3) необходимо восстановить перестановку $j_1^*, j_2^*, \ldots, j_m^*$.

Разобьем процедуру восстановления искомой перестановки на следующие этапы.

1. Нахождение типа повторов, к которому принадлежит декодируемый буфер.

2. Восстановление перестановки.

Алгоритм декодирования выглядит следующим образом.

1.1. Найдем номер типа повторов, к которому принадлежит искомая перестановка. Для этого проверим условие

$$\sum_{i=1}^{N} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} > \tilde{N} \quad (7)$$

последовательным перебором $N = 1, 2, \ldots, C_{m-1}^{s-1}$. Последовательности чисел кратности n_k^i можно найти, зная m и s, используя алгоритмы, описанные в работах [9, 10].

1.2. Найдем минимальное N, для которого выполнено условие (7). Это и будет номер того типа повторов, который содержит в себе декодируемую перестановку. Обозначим его как N^*.

2.1. Теперь будем последовательно определять элементы искомой перестановки $j_1^*, j_2^*, \ldots, j_m^*$.

2.2. Для нахождения j_1^* проверим условие

$$\sum_{i=1}^{N^*-1} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} + \sum_{j_1=1}^{q_1} L_{j_1} > \tilde{N} \quad (8)$$

последовательным перебором $q_1 = 1, 2, \ldots, s$.

2.3. Найдем минимальное q_1, удовлетворяющее условию (8).

2.4. Положим $j_1^* = \min\{q_1\}$.

2.5. Для нахождения j_2^* проверим условие

$$\sum_{i=1}^{N^*-1} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} + \sum_{j_1=1}^{j_1^*-1} L_{j_1} + \sum_{j_2=1}^{q_2} L_{j_1^* j_2} > \tilde{N} \tag{9}$$

последовательным перебором $q_2 = 1, 2, \ldots, s$.

2.6. Найдем минимальное q_2, удовлетворяющее условию (9).

2.7. Положим $j_2^* = \min\{q_2\}$.

...

2.8. Для нахождения j_i^* проверим условие

$$\sum_{i=1}^{N^*-1} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} + \sum_{j_1=1}^{j_1^*-1} L_{j_1} + \sum_{j_2=1}^{j_2^*-1} L_{j_1^* j_2} + \ldots + \sum_{j_i=1}^{q_i} L_{j_1^* j_2^* \ldots j_i} > \tilde{N} \tag{10}$$

последовательным перебором $q_i = 1, 2, \ldots, s$.

2.9. Найдем минимальное q_i, удовлетворяющее условию (10).

2.10. Положим $j_i^* = \min\{q_i\}$.

...

2.11. Для нахождения j_{m-1} проверим условие

$$\sum_{i=1}^{N^*-1} \frac{m!}{\prod\limits_{k=1}^{s} n_k^i!} + \sum_{j_1=1}^{j_1^*-1} L_{j_1} + \sum_{j_2=1}^{j_2^*-1} L_{j_1^* j_2} + \ldots + \sum_{j_i=1}^{j_i^*-1} L_{j_1^* j_2^* \ldots j_i} + \ldots + \sum_{j_{m-1}=1}^{q_{m-1}} L_{j_1^* j_2^* \ldots j_{m-1}} > \tilde{N} \tag{11}$$

последовательным перебором $q_{m-1} = 1, 2, \ldots, s$.

2.12. Найдем минимальное q_{m-1}, удовлетворяющее условию (11).

2.13. Положим $j_{m-1}^* = \min\{q_{m-1}\}$.

2.14. Последний элемент перестановки j_m^* определяется исходя из типа повторов с учетом восстановленных элементов $j_1^*, j_2^*, \ldots, j_{m-1}^*$. Таким образом, итогом всех произведенных операций будет восстановленный вид декодируемого буфера.

6. Заключение

Предложенный в данной работе алгоритм кодирования поля кратности и поля порядка одновременно одним числом обеспечивает длину кода, меньшую, чем при кодировании вышеназванных полей по отдельности. Благодаря структурному разбиению таблицы (3) на блоки и возможности

вычислять длины целых блоков и подблоков без их построчной генерации, реализация алгоритма занимает сравнительно малый промежуток времени.

Автор выражает глубокую благодарность А. А. Толстопятову, доценту кафедры ТФМКМ физического факультета ИвГУ, за постановку задачи и полезные консультации.

Список литературы

1. *Толстопятов А. А.* О структуре дискретной информации и общих условиях ее сжатия // Вестник ИвГУ. – 2002. – Вып. 3. – С. 80–82.
2. *Толстопятов А. А.* Вычисление длины поля кратности при булевом сжатии файлов // Вестник ИвГУ. – 2004. – Вып. 3. – С. 71–76.
3. *Толстопятов А. А.* Быстрый алгоритм кодирования и декодирования поля порядка при булевом сжатии файлов // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2007. – Вып. 1 (4). – С. 35–46.
4. *Толстопятов А. А.* Медленный алгоритм кодирования и декодирования поля кратности при булевом сжатии файлов // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2007. – Вып. 1 (4). – С. 47–52.
5. *Толстопятов А. А.* Быстрый алгоритм кодирования и декодирования поля кратности при булевом сжатии файлов // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2007. – Вып. 1 (4). – С. 53–78.
6. *Толстопятов А. А.* Алгоритм кодирования и декодирования поля принадлежности // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2008. – Вып. 1 (5). – С. 53–76.
7. *Толстопятов А. А.* Возможность кодирования поля кратности и поля порядка одним числом // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2009. – Вып. 1 (6). – С. 121–128.
8. *Толстопятов А. А., Хашин С. И.* Алгоритм построения поля порядка при булевом сжатии // Вестник ИвГУ. – 2004. – Вып. 3. – С. 139–143.
9. *Толстопятов А. А., Гришко М. Е.* Быстрый алгоритм кодирования и декодирования поля кратности одним числом при булевом сжатии файлов // Вестник ИвГУ. – 2009. – Вып. 2. – С. 45–52.
10. *Толстопятов А. А., Гришко М. Е.* Медленный алгоритм кодирования и декодирования поля кратности одним числом при булевом сжатии файлов // Вестник ИвГУ. – 2009. – Вып. 2. – С. 53–55.

Поступила в редакцию 19.06.2009.