Математика и её приложения: ЖИМО. 2007. Выпуск 1 (4). С. 47 – 52.

УДК 512.54

А. А. Толстопятов[1]

Медленный алгоритм кодирования и декодирования поля кратности при булевом сжатии файлов

Ключевые слова: булева алгебра, сжатие информации.

Предложен алгоритм, позволяющий для разложения заданного натурального числа на сумму фиксированного числа натуральных слагаемых построить лексикографически упорядоченую таблицу всех таких разложений и определить номер в этой таблице для любого разложения. Алгоритм работает медленно, так как приходится последовательно генерировать всю таблицу, но прост для программной реализации.

We suggest the algorithm that construct for representation of any natural number as a sum of fixed quantity of natural addends the lexicographically ordered table of all such representations; this algorithm also determines the number of such representation in this table. The algorithm works slowly, but is simple for program realization.

В работе [1] рассмотрен подход к сжатию дискретной информации без потерь, основанный на разбиении файла на кортежи длины n и объединении этих кортежей в буферы разной длины. При этом код каждого кортежа состоит из следующих трех полей:

(1) *Поле принадлежности.* Это поле задается коэффициентами булева уравнения, решения которого дают кортежи, входящие в данный буфер.

(2) *Поле кратности.* Это поле содержит информацию о том, сколько раз каждый кортеж входит в данный буфер. Для построения кода этого поля нужно определить номер разложения числа кортежей в буфере в сумму, содержащую столько слагаемых, сколько разных кортежей содержит данный буфер в лексикографически упорядоченной таблице всех таких разложений.

(3) *Поле порядка.* Это поле содержит информацию о порядке расстановки кортежей внутри буфера. Его код задается номером в лексикографически упорядоченной таблице всех перестановок такого типа, который определяется полем кратности.

В настоящей работе рассматривается алгоритм, позволяющий построить код поля кратности. Если обозначить число кортежей в буфере через

[1]Ивановский государственный университет; E-mail: khash2@mail.ru. Работа выполнена при финансовой поддержке РФФИ (проект 07-07-00155).

m, а число разных кортежей через s, то длина кода поля кратности L_2, если задавать этот код номером в описанной ниже таблице, будет равна

$$L_2 = \log_2 D^s_{m-s}. \tag{1}$$

Как показано в [2], величина D^s_{m-s} может быть вычислена либо с использованием производящей функции

$$F(x) = \sum_{n=0}^{\infty} D^s_n x^n = \frac{1}{\prod\limits_{k=1}^{s}(1-x^k)}, \tag{2}$$

$$D^s_n = \frac{1}{n!} \lim_{x\to\infty} \frac{d^n}{dx^n} \frac{1}{\prod\limits_{k=1}^{s}(1-x^k)}, \tag{3}$$

либо с использованием рекуррентных формул:

$$\begin{array}{lll} 1.\ D^s_n & = D^s_{n-s} + D^{s-1}_n, & s \le n, \\ 2.\ D^s_n & = D^n_n, & s > n, \\ 3.\ D^s_0 & = 1, & \\ 4.\ D^1_n & = 1, & \\ 5.\ D^2_n & = \left[\frac{n}{2}\right] + 1. & \end{array} \tag{4}$$

Однако, в работе [2] были получены только асимптотические формулы для оценки L_2, но не был построен алгоритм получения указанного выше номера. В настоящей работе этот пробел будет заполнен.

1. Постановка задачи

Пусть буфер файла разбит на m кортежей, причем среди них s различных. Назовем типом кратности $n_1, n_2, \ldots, n_s$, где n_s — число вхождений в файл k-го кортежа, $k = 1, 2, \ldots, s$. При этом $1 \le s \le m$ и выполнено соотношение

$$\sum_{k=1}^{s} n_k = m. \tag{5}$$

Если считать, что n_k лексикографически упорядочены, т. е.

$$n_k \le n_{k+1},\ k = 1, 2, \ldots, s-1,$$

то при фиксированных m и s все такие типы кортежей (или типы повторов) образуют таблицу:

$$\begin{array}{lll} 1. & n_k = 1, & n_s = m - s + 1, \\ & & k = 1, 2, \ldots, s-1, \\ 2. & n_k = 1, & n_{s-1} = 2, n_s = m - s \\ & & k = 1, 2, \ldots, s-2, \\ & \ldots & \\ D^2_{m-s}. & n_k = \left[\frac{m}{s}\right], & n_{p+l} = \left[\frac{m}{s}\right] + 1, \\ & & k = 1, 2, \ldots, p, \\ & & p = \left(\left[\frac{m}{s}\right] + 1\right) s - m, \\ & & l = 1, 2, \ldots, s-p. \end{array} \tag{6}$$

Задача построения кода поля кратности заключается в том, чтобы по заданному типу повторов $n_1, n_2, \ldots, n_s$ вычислить номер этого типа в таблице (6). Задача декодирования заключается в том, чтобы при заданных m и s по номеру в таблице (6) восстановить тип повторов $n_1, n_2, \ldots, n_s$. В настоящей работе эти задачи будут решены с помощью алгоритма, генерирующего последовательно таблицу (6), и на каждом шаге сравнивающего заданный тип повтора $n_1, n_2, \ldots, n_s$ с соответствующей строкой таблицы (6).

2. Алгоритм кодирования

1. Обозначим через $\mathcal{N} = (n_1, n_2, \ldots, n_s)$ тип повторов с заданным s. При этом m определяется из формулы (5).

2. Назовем $\widetilde{\mathcal{N}} = (\tilde{n}_1, \tilde{n}_2, \ldots, \tilde{n}_s)$ стандартной формой типа повторов, если $\widetilde{\mathcal{N}}$ получается из $\mathcal{N}$ такой перестановкой, чтобы выполнялось условие

$$\tilde{n}_1 \le \tilde{n}_2 \le \cdots \le \tilde{n}_s. \tag{7}$$

3. Определим последовательную процедуру генерации стандартных форм, положив, что переход от i-ой стандартной формы

$$\widetilde{\mathcal{N}}_i = (\tilde{n}^i_1, \tilde{n}^i_2, \ldots, \tilde{n}^i_s) \tag{8}$$

к $(i+1)$-ой стандартной форме

$$\widetilde{\mathcal{N}}_{i+1} = (\widetilde{n}^{i+1}_1, \widetilde{n}^{i+1}_2, \ldots, \widetilde{n}^{i+1}_s) \tag{9}$$

проводится следующим образом.

3.1. Определяется число g первых a^i_j, $j = 1, 2, \ldots, g$, которые не будут меняться при переходе от $\widetilde{\mathcal{N}}_i$ к $\widetilde{\mathcal{N}}_{i+1}$.

3.2. Проверяется первое условие

$$\widetilde{n}^i_{s-1} - \widetilde{n}^i_{s-t-1} > 1, \tag{10}$$

когда параметру t последовательно придаются значения $t = 0, 1, \ldots, s-2$.

3.3. Находится минимальное значение t ($\min t$) из 0, 1, 2, ..., $s-2$, при котором выполнено условие (10).

3.4. Тогда

$$g_1 = s - 2 - \min t, \quad t = 0, 1, 2, \ldots, s-2. \tag{11}$$

3.5. Проверяется второе условие

$$\widetilde{n}^i_{s-1} - \widetilde{n}^i_{s-t} = 2 \tag{12}$$

при последовательно изменяемых значениях $t = 2, 3, \ldots, s-1$.

3.6. Находятся минимальное значение t из 2, 3, ..., $s-1$.

3.7. Тогда

$$g_2 = s - 1 - \min t, \quad t = 2, 3, \ldots, s-1. \tag{13}$$

3.8. Из g_1 и g_2 выбирается максимальное значение

$$g = \max(g_1, g_2). \tag{14}$$

3.9. Тогда алгоритм перехода от $\widetilde{\mathcal{N}}_i$ из (8) к $\widetilde{\mathcal{N}}_{i+1}$ из (9) имеет вид:

$$\begin{array}{ll} \widetilde{n}^{i+1}_k & = \widetilde{n}^i_k, \quad k = 1, 2, \ldots, g; \\ \widetilde{n}^{i+1}_k & = \widetilde{n}^i_{g+1}, \quad k = g+1, g+2, \ldots, s-1; \\ \widetilde{n}^{i+1}_s & = m - \sum\limits_{k=1}^{g} \widetilde{n}^i_k - (\widetilde{n}^i_{g+1} + 1)(s-g-1). \end{array} \tag{15}$$

4. Описание процедуры генерации стандартной формы $\widetilde{\mathcal{N}}_{i+1}$ (9) из стандартной формы $\widetilde{\mathcal{N}}_i$ (8) начинается с первой стандартной формы

$$\widetilde{\mathcal{N}_1} = (\tilde{n}^1_1, \tilde{n}^1_2, \ldots, \tilde{n}^1_s), \tag{16}$$

где

$$\begin{array}{ll} \widetilde{n}^i_k & = 1, \quad k = 1, 2, \ldots, s-1, \\ \widetilde{n}^1_s & = m - s + 1. \end{array} \tag{17}$$

5. Заканчивается процедура генерацией последней стандартной формы

$$\widetilde{\mathcal{N}_f} = (\tilde{n}^f_1, \tilde{n}^f_2, \ldots, \tilde{n}^f_s), \tag{18}$$

где

$$\begin{array}{ll} \widetilde{n}^f_k & = \left[\frac{m}{s}\right], \qquad k = 1, 2, \ldots, g_1, \\ \widetilde{n}^f_k & = 1 + \left[\frac{m}{s}\right], \quad k = g_1 + 1, \ldots, s, \end{array} \tag{19}$$

где числа g_1 и g_2 определяются следующим образом:

$$g_1 = s - m + s\left[\frac{m}{s}\right] \quad — \tag{20}$$

число первых $\widetilde{n}_k^f$, равных $\left[\frac{m}{s}\right]$.

$$g_2 = m - s\left[\frac{m}{s}\right] \quad —$$

число первых $\widetilde{n}_k^f$ равных $1 + \left[\frac{m}{s}\right]$.

6. На каждом шаге переходов от $\widetilde{\mathcal{N}_1}$ к $\widetilde{\mathcal{N}_2}$, от $\widetilde{\mathcal{N}_2}$ к $\widetilde{\mathcal{N}_3}$ и т. д. нужно сравнивать получаемую стандартную форму с исходной стандартной формой $\widetilde{\mathcal{N}}$, номер которой нужно вычислить.

7. Если $\widetilde{\mathcal{N}_i}$ не совпадает с $\widetilde{\mathcal{N}}$, то нужно вычислить соответствующие p_i и z_i^j, где p_i — количество различных чисел в форме $\widetilde{\mathcal{N}_i}$, а z_i^j — кратность каждого из p_i чисел в стандартной форме $\widetilde{\mathcal{N}_i}$.

8. Тогда каждой стандартной форме $\widetilde{\mathcal{N}_i}$ ставится в соответствие число

$$\sigma_i = \frac{s!}{\prod\limits_{j=1}^{p_i} z_i^j!}, \tag{21}$$

причем, если $\widetilde{\mathcal{N}_i}$ совпадает с $\widetilde{\mathcal{N}}$, то σ_i вычислять не нужно.

9. Зная все $\sigma_1, \sigma_2, \ldots, \sigma_{i-1}$, найдем номер N_1 равный

$$N_1 = \sum_{k=1}^{i-1} \sigma_k. \tag{22}$$

10. Номер N типа повторов $\mathcal{N}$ в полученной таблице будет равен

$$N = N_1 + N_2. \tag{23}$$

11. Для определения номера N_2 строится следующая процедура. Берется исходная форма $\mathcal{N} = (n_1, n_2, \ldots, n_s)$.

12. Находится в $\mathcal{N} = (n_1, n_2, \ldots, n_s)$ минимальное значение $n'_1, n'_2, \ldots, n'_\nu$, где $n'_1 = n'_2, \ldots, n'_\nu$.

13. Производится замена

$$n'_1 \to 1,\ n'_2 \to 1,\ \ldots,\ n'_\nu \to 1. \tag{24}$$

14. Удаляются из $\mathcal{N} = (n_1, n_2, \ldots, n_s)$ все $n'_1, n'_2, \ldots, n'_\nu$.

15. В оставшемся множестве находятся минимумы значений $n_1^2, n_2^2, \ldots, n_{s-\nu}^2$.

16. Производится замена

$$n_1^2 \to 2,\ n_2^2 \to 2,\ \ldots,\ n_{s-\nu}^2 \to 2. \tag{25}$$

17. Удаляются из списка все $n_1^2, n_2^2, \ldots, n_{s-\nu}^2$.

18. Процедура, описанная в пп. 12–17 повторяется до тех пор, пока $\mathcal{N} = (n_1, n_2, \ldots, n_s)$ не превратится в пустое множество.

19. В $\mathcal{N} = (n_1, n_2, \ldots, n_s)$ вместо n_k подставляются присвоенные им значения и получается перестановка $j_1, j_2, \ldots, j_s$.

20. По известному алгоритму [3] по перестановке $j_1, j_2, \ldots, j_s$ находится ее номер N_2.

21. Тогда, в соответствии с (23):

$$N = N_1 + N_2.$$

3. Алгоритм декодирования

Алгоритм декодирования должен по известным m, s, N восстановить тип повторов $\mathcal{N} = (n_1, n_2, \ldots, n_s)$. По существу он уже содержится в алгоритме кодирования. Если сгенерированная при кодировании таблица (6) сохраняется, то достаточно просто найти в этой таблице строку с номером N. Она будет содержать искомый тип повторов $\mathcal{N} = (n_1, n_2, \ldots, n_s)$. Тогда декодирование будет работать быстро, но потребует много памяти. При больших m этот вариант может оказаться невозможен технически. В этом случае можно при декодировании заново сгенерировать таблицу (6) согласно пп. 3–5 предыдущего раздела этой работы, закончив генерацию не последней стандартной формой $\widetilde{\mathcal{N}_f}$ из (18), (19), а формой с номером N. Тогда для декодирования никакой дополнительной памяти, кроме памяти для хранения номера N, не потребуется, но потребуется больше времени для декодирования, чем в первом варианте.

В заключение отметим, что тестирование рассмотренного в этой работе алгоритма показало, что при $m \sim 100$, когда среди n_k, $k = 1, 2, \ldots, s$, есть много чисел, больших единицы, алгоритм работает недостаточно быстро, поэтому он и был назван медленным.

Список использованной 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.

Поступила в редакцию 21.11.2007.