ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 1999. Серия 1. Том 6, № 2, 70–81

УДК 519.176

ОЦЕНКИ ИЗБЫТОЧНОСТИ КОДИРОВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ АЛГОРИТМОМ ЛЕМПЕЛА — ЗИВА*)

В. Н. Потапов

Рассматривается задача неискажающего сжатия (кодирования) буквенных последовательностей. Для последовательностей с асимптотически нулевой эмпирической энтропией предложена модификация схемы кодирования Лемпела — Зива, для которой стоимость кодирования превышает энтропию не более чем в конечное число раз. Кроме того, предложено комбинаторное доказательство известной оценки избыточности схемы кодирования Лемпела — Зива для последовательностей с положительной энтропией.

Введение

В работах [19, 20] Дж. Зивом и А. Лемпелем были предложены методы кодирования (называемые в дальнейшем LZ77 и LZ78 соответственно), которые впоследствии получили широкое применение для решения задачи сжатия данных. В настоящее время известно множество модификаций этой схемы [2, 5, 7, 9, 11, 14–16]. Использование алгоритмов, основанных на схемах типа Лемпела — Зива, при разработке средств программного обеспечения вызывает неослабевающий интерес к теоретическим оценкам качества сжатия, которое обеспечивают данные схемы. В последние годы были получены точные по порядку оценки избыточности кодирования для различных модификаций алгоритма Лемпела — Зива [8, 10, 12, 13, 17, 18]. Особенно важной с практической точки зрения является оценка эмпирической избыточности $R(f, x_1^n)$ кодирования f произвольной последовательности x_1^n, состоящей из n букв некоторого конечного алфавита. Величина $R(f, x_1^n)$ определяется как разность между длиной кода $f(x_1^n)$ последовательности x_1^n и эмпирической энтропией $H(x_1^n)$ этой последовательности, когда длина кода и энтропия

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 99–01–00531) и Федеральной целевой программы «Интеграция» (проект 1997 г. № 473).

вычислены в расчете на одну букву исходной последовательности. Наилучшие оценки избыточности схемы Лемпела — Зива были получены С. Савари [12, 13]:

$$R(f_1, x_1^n) = O\Big(\frac{1}{\log n}\Big) \tag{1}$$

и

$$R(f_2, x_1^n) \leqslant \frac{CH(x_1^n)\log\log n}{\log n}(1 + o(1)) \tag{2}$$

при $n \to \infty$ и $\lim\limits_{n\to\infty} H(x_1^n) > 0$, где $C = 2$, а f_1 и f_2 построены по схемам кодирования LZ78 и LZ77 соответственно. Здесь и в дальнейшем символ log обозначает логарифм по основанию 2. Однако существуют примеры таких непериодических последовательностей x (x_1^n — первые n букв последовательности x), что

$$\lim_{n\to\infty} \frac{R(f, x_1^n)}{H(x_1^n)} = \infty,$$

где кодирование f построено по схеме LZ77 или LZ78.

В настоящей работе предложена схема кодирования, представляющая собой объединение алгоритмов LZ77 и LZ78. Для кодирования f, построенного по предлагаемой схеме, справедлива оценка избыточности (2), где $C = 1$, если $\lim\limits_{n\to\infty} H(x_1^n) > 0$, и оценка избыточности

$$R(f, x_1^n) = O(H(x_1^n))$$

для произвольной непериодической последовательности x. Таким образом, предлагаемый алгоритм в отличие от LZ77 и LZ78 гарантирует, что длина кода последовательности не более чем в конечное число раз превосходит эмпирическую энтропию последовательности.

Кроме того, в настоящей работе предлагается прямое, т. е. комбинаторное доказательство оценки (2), где $C = 1$ для схемы кодирования LZ78 и $C = 3$ для схемы кодирования LZ77. Полученные оценки несколько хуже известных оценок (1) и (2), доказанных более трудоемкими теоретико-вероятностными методами.

1. Основные определения

Пусть $A = \{a_1, \ldots, a_{|A|}\}$ — некоторый конечный алфавит и $A^* = \bigcup\limits_{n=1}^{\infty} A^n$ — множество всех конечных последовательностей букв алфавита A. Если x, y — слова, то через xy обозначим конкатенацию слов x и y. Через x_l^r будем обозначать слово, состоящее из букв слова $x = a_{i_1} \ldots a_{i_n}$, начиная с l-й и кончая r-й, т. е. $x_l^r = a_{i_l} \ldots a_{i_r}$. Через $x(y) = a_{i_1} \ldots a_{i_m}$

обозначим слово, состоящее из букв слова x, непосредственно следующих за подсловом y, т. е. для каждой буквы a_{i_j}, $1 \leqslant j \leqslant m$, содержащейся в $x(y)$, найдутся такие подслова x_1^l и x_r^n, что $x = x_1^l y a_{i_j} x_r^n$. Через $|x|$ будем обозначать длину слова x.

Эмпирической энтропией (0-го порядка) слова $x_1^n \in A^n$ (см. [3]) называется величина

$$H(x_1^n) = \sum_{i=1}^{|A|} \frac{r_i}{n} \log \frac{n}{r_i}, \tag{3}$$

где r_i — число вхождений буквы a_i в слово x_1^n. Пользуясь формулой Стирлинга и (3), получаем

$$H(x_1^n) = \frac{1}{n} \log \frac{n!}{r_1! r_2! \dots r_{|A|}!} + \alpha(x_1^n), \tag{4}$$

где $\alpha(x_1^n) \geqslant 0$ и $\alpha(x_1^n) \leqslant \alpha'(n) \to 0$ при $n \to \infty$.

Эмпирической энтропией k-го порядка слова $x_1^n \in A^n$ (см. [1]) называется величина

$$H_k(x_1^n) = \sum_{y \in A^k} \frac{n(y)}{n} H(x_1^n(y)), \tag{5}$$

где $n(y) = |x_1^n(y)|$.

Кодированием будем называть инъективное отображение $f : A^* \to E^*$ ($E = \{0, 1\}$), ставящее в соответствие каждому слову в алфавите A двоичную последовательность (код этого слова). Кодирование f называется префиксным, если для любых двух различных слов x_1^n и y_1^n длины n в алфавите A кодовая последовательность $f(x_1^n)$ не является префиксом (началом) кодовой последовательности $f(y_1^n)$.

В дальнейшем будет использован префиксный код $\gamma(n)$ чисел натурального ряда, предложенный П. Элайесом [6]. Аналогичные коды были известны и ранее (см., например, [4]). Для каждого натурального n справедливо равенство

$$|\gamma(n)| = 2\lfloor \log(\lfloor \log n \rfloor + 1) \rfloor + \lfloor \log n \rfloor + 1. \tag{6}$$

Избыточностью (эмпирической) k-го порядка кодирования f для слова x_1^n называется величина

$$R_k(f, x_1^n) = \frac{1}{n} |f(x_1^n)| - H_k(f, x_1^n). \tag{7}$$

Рассмотрим множество $X(x_1^n) \subset A^n$, состоящее из слов, в которых имеется столько же вхождений буквы a_i, $1 \leqslant i \leqslant |A|$, что и в слове x_1^n. Тогда для произвольного префиксного кодирования f и $x \in A^\infty$ из (4) и (7) следует неравенство

$$\lim_{n \to \infty} \Big(\sup_{z \in X(x_1^n)} R(f, z) \Big) \geqslant 0.$$

Аналогично можно показать, что

$$\lim_{n\to\infty}\Big(\sup_{z\in X_k(x_1^n)} R_k(f,z)\Big)\geqslant 0,$$

где $X_k(x_1^n)\subset A^n$ — множество таких $z\in A^n$, что $z(y)$ и $x_1^n(y)$ имеют одинаковый частотный состав для всех $y\in A^k$.

2. Схема кодирования Лемпела — Зива и ее модификации

Алгоритм LZ77 [19] состоит в разделении кодируемого слова $x_1^n\in A^n$ на подслова σ_i, $1\leqslant i\leqslant m$, по следующему правилу. Пусть начало слова x_1^n уже разделено на подслова, т. е. это начало представляет собой конкатенацию подслов $\sigma_1,\sigma_2,\ldots,\sigma_i$ и $x_1^n=\sigma_1\ldots\sigma_i x_{l_i}^n$. Выберем наиболее длинную последовательность $x_{l_i}^{r_i}$, которая уже встречалась в $x_1^{r_i-1}$ — начале слова x_1^n, т. е. $x_{l_i}^{r_i}=x_{l_i-n_i}^{r_i-n_i}$, где $1\leqslant n_i<l_i$. Определим следующее подслово σ_{i+1} как $\sigma_{i+1}=x_{l_i}^{r_i}a_{p_i}$, где a_{p_i} — следующая за $x_{l_i}^{r_i}$ буква слова x_1^n. Кодом каждого подслова σ_{i+1} является тройка чисел (r_i-l_i,n_i,p_i). Например, последовательность $a_2a_1a_2a_1a_1a_2a_1a_1a_2a_2$ разделяется на подслова $a_2,a_1,a_2a_1a_1,a_2a_1a_1a_2a_2$ и кодируется последовательностью троек: $(0,0,2),(0,0,1),(2,2,1),(4,3,2)$. Первое число в тройке (r_i-l_i,n_i,p_i) запишем с помощью кодирования γ, второе и третье будем записывать в двоичном виде с использованием $\lfloor\log n\rfloor+1$ битов и $\lfloor\log|A|\rfloor+1$ битов соответственно. Тогда из (6) имеем

$$|f_1(x_1^n)|\leqslant\sum_{i=1}^{m}(\log n+\log|\sigma_i|+2\log(1+\log|\sigma_i|)+\log|A|+3),\qquad(8)$$

где кодирование f_1 построено по схеме LZ77, m — число подслов σ_i, на которые разбивается последовательность x_1^n алгоритмом LZ77. Из построения ясно, что f_1 — префиксное кодирование.

Алгоритм LZ78 [20] отличается от описанного выше тем, что на каждом шаге выбирается наиболее длинная начальная последовательность в остатке $x_{l_i}^n$, которая совпадает с некоторым уже выделенным подсловом σ_j, $j<i$, и к ней добавляется еще одна буква, т. е. $\sigma_{i+1}=\sigma_j a_{p_i}$. Код подслова σ_{i+1} определим как пару чисел (j,p_i). Например, последовательность $a_2a_1a_2a_1a_1a_2a_1a_2a_1$ разделяется на подслова $a_2,a_1,a_2a_1,a_1a_2,a_1a_2a_1$ и кодируется последовательностью пар чисел $(0,2),(0,1),(1,1),(2,2),(4,1)$. Построенное по схеме LZ78 кодирование f_2 определим как последовательность пар чисел (j,s), причем первое число i-й пары записано с использованием $\lfloor\log i\rfloor+1$, а второе — $\lfloor\log|A|\rfloor+1$ двоичных символов. Тогда

$$|f_2(x_1^n)|=\sum_{i=1}^{m}(\log i+\log|A|+2)\leqslant m(\log m+\log|A|+2),\qquad(9)$$

где m — число подслов σ_i, на которые разбивается последовательность x_1^n в соответствии с алгоритмом LZ78. Из построения ясно, что f_2 — префиксное кодирование.

Предлагаемая модификация алгоритма Лемпела — Зива (в дальнейшем LZP) основана на схемах кодирования LZ77 и LZ78. При каждом выборе очередного подслова используется один из двух способов: первый — как в алгоритме LZ78, второй — как в алгоритме LZ77, причем второй способ применяется только тогда, когда $n_i < r_i - l_i$. Выбор первого или второго способа определяется указателем (0 или 1) в зависимости от того, каким способом можно выделить более длинное подслово. Кодирование подслов осуществляется так же, как при использовании схем LZ78 и LZ77 соответственно с тем лишь отличием, что во втором случае для записи числа n_i используется $\lceil \log(r_i - l_i) \rceil$ битов. Например, последовательность $a_1a_2a_1a_1a_1a_1a_1a_1a_2$ будет разделена на подслова $a_1, a_2, a_1a_1, a_1a_1a_1a_1a_2$ и кодируется тремя тройками и четверкой чисел $(0,0,1),(0,0,2),(0,1,1),(1,4,2,2)$. Ясно, что длина кода $|f_3(x_1^n)|$, построенного по схеме LZP, оценивается следующим образом:

$$|f_3(x_1^n)| \leqslant m_1 \log m + 2 \sum_{\sigma_i \in B_2} (\log|\sigma_i| + \log(1 + \log|\sigma_i|)) + m(\log|A| + 3), \quad (10)$$

где m_1 — число подслов, выделенных первым способом, B_2 — множество слов, выделенных вторым способом, и $m = m_1 + |B_2|$ — число всех подслов, на которые разбивается слово x_1^n. Кодирование f_3 является префиксным по построению, как и кодирования f_1 и f_2.

Все подслова, выделяемые алгоритмами LZ77, LZ78 и LZP из слова x_1^n, попарно различны, за исключением, возможно, последнего подслова, которое может совпадать с одним из предыдущих. В дальнейшем для упрощения вычислений будем полагать, что все подслова, включая последнее, различны.

3. Основные результаты

Следующая лемма будет использована для оценки избыточности кодирования алгоритмом LZP последовательностей с асимптотически нулевой энтропией.

Лемма 1. *Пусть* $x_1^n \in A^n$ *и* $x_1^n = \sigma_1 \ldots \sigma_m$, *где* σ_i, $1 \leqslant i \leqslant m$, — *подслова, выделенные алгоритмом* LZP. *Тогда*

$$nH_k(x_1^n) \geqslant \sum_{|\sigma_i| > |A|^k} \max(\log(|\sigma_i|/|A|^k), 1).$$

Доказательство. Введем обозначение $\sigma_i' = z_i\sigma_i$, где z_i — подслово, состоящее из k букв, непосредственно предшествующих подслову σ_i в слове x_1^n (для первых подслов σ_i подслово z_i может состоят из меньшего числа букв). Пусть $y \in A^k$ и $\sigma_i'(y)$ содержит различные буквы. Тогда из (3) следует, что

$$|\sigma_i'(y)|H(\sigma_i'(y)) \geqslant \log|\sigma_i'(y)|. \tag{11}$$

Поскольку $\sum\limits_{y\in A^k} |\sigma_i'(y)| = |\sigma_i|$ и $|\sigma_i| > |A|^k$, то найдется $y \in A^k$ такое, что

$$|\sigma_i'(y)| \geqslant |\sigma_i|/|A|^k, \quad |\sigma_i'(y)| \geqslant 2. \tag{12}$$

Если $\sigma_i'(y)$ содержит последнюю букву подслова σ_i, то из алгоритма LZP следует, что $\sigma_i'(y)$ содержит не менее двух различных букв и неравенство (11) справедливо. Пусть $\sigma_i'(y)$ не содержит последнюю букву подслова σ_i и состоит из повторений единственной буквы $a_{i(y)}$. Пусть $y = a_{i_1}a_{i_3}\ldots a_{i_k}$. Рассмотрим слово $y_1 = a_{i_2}a_{i_3}\ldots a_{i_k}a_{i(y)}$. Тогда из определения y_1 следует, что $|\sigma_i'(y_1)| \geqslant |\sigma_i'(y)| \geqslant \max(|\sigma_i|/|A|^k, 2)$. Если y_1 не состоит из повторений одной буквы, то неравенство (11) справедливо. В противном случае определим y_2 аналогично y_1, т. е. $y_2 = a_{i_3}\ldots a_{i_k}a_{i(y)}a_{i(y_1)}$, y_3, y_4 и так далее. Поскольку $y_j \in A^k$, то последовательность $y_1, y_2, y_3, \ldots$ либо прерывается, либо является циклической. Если $y_1, y_2, y_3, \ldots$ — циклическая, то все k-элементные блоки подслова σ_i являются членами последовательности $y_1, y_2, y_3, \ldots$, что противоречит алгоритму LZP выбора подслов σ_i. Таким образом, если $|\sigma_i| > |A|^k$, то найдется слово $y \in A^k$ такое, что справедливы неравенства (11) и (12).

Поскольку функция $\log x$ выпукла вверх, то из неравенства Йенсена и определения энтропии (3) следует неравенство

$$|x_1^n(y)|H(x_1^n(y)) \geqslant \sum_{i=1}^{m} |\sigma_i'(y)|H(\sigma_i'(y)).$$

Тогда из (5), (11), (12) и последнего неравенства следует утверждение леммы 1.

Оценка избыточности схемы кодирования Лемпела — Зива основывается на следующем утверждении.

Лемма 2. *Пусть $x_1^n \in A^n$ и $x_1^n = \sigma_1\sigma_2\ldots\sigma_m$, где $\sigma_i \neq \sigma_j$ при $i \neq j$. Тогда для каждого целого $k \geqslant 0$ справедливо неравенство*

$$nH_k(x_1^n) \geqslant m\log m - \sum_{i=1}^{m}\log|\sigma_i| - 2\sum_{i=1}^{m}\log(1+\log|\sigma_i|) - Cm,$$

где константа $C > 0$ зависит только от k и $|A|$.

Доказательство. Пусть $a_0 \neq a_i$, $1 \leqslant i \leqslant |A|$. Введем обозначения $\hat{A} = A \cup a_0$ и $z_1^k = a_0 a_0 \dots a_0$, а также $\hat{\sigma}_i = z_1^k \sigma_i$ и $\hat{x}_1^n = \hat{\sigma}_1 \dots \hat{\sigma}_m$. Пусть S_m — множество перестановок длины m. Пусть $\tau \in S_m$. Введем обозначение $\tau(\hat{x}_1^n) = \hat{\sigma}_{\tau(1)} \dots \hat{\sigma}_{\tau(m)}$. Тогда $\tau(\hat{x}_1^n) \neq \tau'(\hat{x}_1^n)$ при $\tau \neq \tau'$, поскольку $a_0 \in \hat{A} \backslash A$ и слова $\tau(\hat{x}_1^n)$, $\tau'(\hat{x}_1^n)$ однозначно разделяются на подслова σ_i, которые попарно различны по условию.

Рассмотрим $y \in \hat{A}^k$ и слово $\hat{\sigma}_1(y)\hat{\sigma}_2(y) \dots \hat{\sigma}_m(y)$. Если $\tau \in S_m$, то $\hat{\sigma}_{\tau(1)}(y)\hat{\sigma}_{\tau(2)}(y) \dots \hat{\sigma}_{\tau(m)}(y)$ — некоторая перестановка $\delta_\tau(y)$ букв слова $\hat{\sigma}_1(y)\hat{\sigma}_2(y) \dots \hat{\sigma}_m(y)$. Пусть $\Delta(y)$ — множество всех таких перестановок $\delta_\tau(y)$, где $\tau \in S_m$. Если $y \in A^k$, то $\hat{\sigma}_1(y)\hat{\sigma}_2(y) \dots \hat{\sigma}_m(y) = \sigma_1(y)\sigma_2(y) \dots \sigma_m(y)$ и $|\sigma_1(y)\sigma_2(y) \dots \sigma_m(y)| \leqslant |x_1^n(y)|$. Кроме того, число вхождений буквы $a_i \in A$, $1 \leqslant i \leqslant |A|$, в $\sigma_1(y)\sigma_2(y) \dots \sigma_m(y)$ не превосходит числа ее вхождений в $x_1^n(y)$. Поэтому

$$|\Delta(y)| \leqslant \frac{n(y)!}{r_1(y)!r_2(y)! \dots r_{|A|}(y)!}, \tag{13}$$

где $n(y) = |x_1^n(y)|$, $r_i(y)$ — число вхождений буквы $a_i \in A$ в слово $x_1^n(y)$. Если $y \in \hat{A}^k \backslash A^k$, то $|\hat{\sigma}_1(y)\hat{\sigma}_2(y) \dots \hat{\sigma}_m(y)| \leqslant m$ и

$$|\Delta(y)| \leqslant |\hat{A}|^m. \tag{14}$$

Каждое слово $\tau(\hat{x}_1^n)$ из S_m можно однозначно задать, указав число букв между соседними подсловами z_1^k в слове $\tau(\hat{x}_1^n)$: $|\hat{\sigma}_{\tau(1)}|, |\hat{\sigma}_{\tau(2)}|, \dots, |\hat{\sigma}_{\tau(m-1)}|$, а также указав перестановки $\delta_\tau(y) \in \Delta(y)$ для всех $y \in \hat{A}^k$. Отсюда, а также из (6) и равенства $|S_m| = m!$ получаем

$$\log m! \leqslant \sum_{y \in \hat{A}^k} \lceil \log |\Delta(y)| \rceil + \sum_{i=1}^{m-1} \lfloor \log |\sigma_i| \rfloor + 2 \sum_{i=1}^{m-1} \lfloor \log(1 + \log |\sigma_i|) \rfloor + m. \tag{15}$$

Из определения слова $\hat{\sigma}_i$ следует, что число различных слов $y \in \hat{A}^k \backslash A^k$, содержащихся во всех $\hat{\sigma}_i$, не превосходит $k|A|$. Из (4), (13), (14) получаем

$$\sum_{y \in \hat{A}^k} \lceil \log |\Delta(y)| \rceil \leqslant \sum_{y \in A^k} \left\lceil \log \frac{n(y)!}{r_1(y)!r_2(y)! \dots r_{|A|}(y)!} \right\rceil + k|A| \lceil \log(|\hat{A}|^m) \rceil$$
$$\leqslant \sum_{y \in A^k} n(y) H(x_1^n(y)) + km|A| \log |\hat{A}| + (|A| + 1)^k.$$

Тогда из (5), (15), неравенства $\log m! \geqslant m \log m - m / \ln 2$ и предыдущего неравенства следует утверждение леммы 2.

В следующей лемме содержится нижняя оценка числа различных подслов, на которые можно разделить слово.

Лемма 3. *Пусть* $\sigma_1, \sigma_2, \ldots, \sigma_m \in A^*$, *где* $\sigma_i \neq \sigma_j$ *при* $i \neq j$. *Тогда справедливо неравенство*

$$\frac{1}{m}\sum_{i=1}^{m}|\sigma_i| \geqslant \frac{\log m}{2\log|A|} - 1.$$

Доказательство. Так как число слов σ_i длины k не превышает $|A|^k$, то

$$\sum_{k=1}^{\lfloor \log_{|A|} m\rfloor - 2} \sum_{|\sigma_i|=k} 1 \leqslant \sum_{k=1}^{\lfloor \log_{|A|} m\rfloor - 2} |A|^k \leqslant \frac{m}{|A|}.$$

Поэтому

$$\sum_{k=\lfloor \log_{|A|} m\rfloor - 1}^{\infty} \sum_{|\sigma_i|=k} 1 \geqslant m - \frac{m}{|A|} \geqslant \frac{m}{2}$$

и

$$\frac{1}{m}\sum_{i=1}^{m}|\sigma_i| \geqslant \frac{1}{m}\frac{m}{2}(\lfloor \log_{|A|} m\rfloor - 1) \geqslant \frac{\log m}{2\log|A|} - 1.$$

Лемма 3 доказана.

Оценим избыточность кодирования последовательности с асимптотически ненулевой энтропией.

Теорема 1. *Пусть* $A = \{a_1, \ldots, a_{|A|}\}$, $x \in A^\infty$, $k \geqslant 0$ — *целое и* $\lim\limits_{n\to\infty} H_k(x_1^n) > 0$. *Тогда*

$$R_k(f, x_1^n) \leqslant \frac{CH_k(x_1^n)\log\log n}{\log n}(1 + o(1)),$$

где $C = 1$ *для кодирований, построенных по схемам* LZ78 *и* LZP, *и* $C = 3$ *для кодирования, построенного по схеме* LZ77.

Доказательство. Пусть алгоритм LZP разбивает слово $x_1^n \in A^n$ на подслова $\sigma_1, \sigma_2, \ldots, \sigma_m$. Поскольку $\sum\limits_{i=1}^{m}|\sigma_i| = n$ и $\log x$ — выпуклая вверх функция, то из неравенства Йенсена следует, что

$$\sum_{i=1}^{m}\log|\sigma_i| \leqslant m\log\left(\sum_{i=1}^{m}|\sigma_i|/m\right) = m\log\frac{n}{m}. \tag{16}$$

Аналогично, учитывая выпуклость вверх функции $\log\log x$, имеем

$$\sum_{i=1}^{m}\log(1 + \log|\sigma_i|) \leqslant m\Big(1 + \log\log\frac{n}{m}\Big). \tag{17}$$

Тогда из (10), (16) и (17) для кодирования f, построенного по схеме LZP, получаем

$$|f(x_1^n)| \leqslant m_1 \log m + 2m_2 \log \frac{n}{m_2} + 2m \log\log \frac{n}{m} + m(\log|A| + 5), \quad (18)$$

где m_1 и m_2 — число подслов, выделенных алгоритмом LZP из слова x_1^n первым и вторым способами соответственно.

Поскольку $-t\log(t/c) \leqslant c$ при $c > 0$ и $t > 0$, то $2m_2 \log \frac{n}{m_2} \leqslant m_2 \log m + 2(n/\sqrt{m})$. Так как все σ_i попарно различны по построению, то из леммы 2, соотношений (7), (16)–(18) и последнего неравенства имеем

$$R_k(f, x_1^n) \leqslant \frac{m}{n} \log \frac{n}{m} + 4\frac{m}{n} \log\log \frac{n}{m} + C'\frac{m}{n} + \frac{2}{\sqrt{m}}, \quad (19)$$

где $C' > 0$ — некоторая константа. Из леммы 3 следует, что $m \to \infty$ и $\frac{m}{n} \to 0$ при $n \to \infty$. Тогда из (19) получаем, что $\limsup\limits_{n\to\infty} R_k(f, x_1^n) \leqslant 0$. Без ограничения общности можно считать, что $\limsup\limits_{n\to\infty} R_k(f, x_1^n) = \lim\limits_{n\to\infty} R_k(f, x_1^n)$, переходя при необходимости к соответствующей подпоследовательности. Если $\lim\limits_{n\to\infty} R_k(f, x_1^n) < 0$, то справедливость теоремы очевидна. Иначе из $\lim\limits_{n\to\infty} H_k(x_1^n) > 0$ и $\lim\limits_{n\to\infty} R_k(f, x_1^n) = 0$ следует, что $H_k(x_1^n) \sim \frac{1}{n}|f(x_1^n)| \sim \frac{m}{n}\log m$ и $\frac{H_k(x_1^n)}{\log m} \sim \frac{m}{n}$, $\frac{\log m}{H_k(x_1^n)} \sim \frac{n}{m}$ при $n \to \infty$. Из перечисленных эквивалентностей и (19) следует утверждение теоремы для кодирования, построенного по схеме LZP. Для кодирований, построенных по схемам LZ77 и LZ78, утверждение теоремы можно доказать аналогичным образом, используя лемму 2 и неравенства (8), (9). Теорема 1 доказана.

Рассмотрим пример такой последовательности x, что

$$|f(x_1^n)|/nH_1(x_1^n) \to \infty \quad (20)$$

при $n \to \infty$, где кодирование f построено по схеме LZ78.

Пусть последовательность x составлена из повторений двух букв: $x_i = a_1$, если $i = 2^k$ для некоторого целого k, и $x_i = a_2$ в остальных случаях. Тогда из (3) и (5) следует, что $H_1(x_1^n) = \frac{\log^2 n}{n}(1 + o(1))$. Алгоритм LZ78 разделяет эту последовательность на не менее чем $\sqrt{n}$ подслов, так как длины подслов не могут возрастать быстрее, чем члены арифметической прогрессии. Тогда из (9) следует, что для кодирования f справедливо равенство

$$|f(x_1^n)| = \frac{\sqrt{n}}{2} \log n(1 + o(1)),$$

т. е.

$$|f(x_1^n)|/nH_1(x_1^n) = \frac{\sqrt{n}}{2\log n}(1+o(1)) \to \infty$$

при $n \to \infty$.

Для схемы кодирования LZ77 также можно привести примеры последовательностей, код которых удовлетворяет соотношению (20). Следующая теорема показывает, что для кода, построенного по схеме LZP, предел в (20) всегда конечен.

Теорема 2. *Пусть* $A = \{a_1, \ldots, a_{|A|}\}$, $x \in A^\infty$, $\lim\limits_{n\to\infty} H_k(x_1^n) = 0$ *и для каждого целого* $i > 0$ *последовательность* X_i^∞ *непериодическая. Тогда*

$$\frac{1}{n}|f(x_1^n)| = O(H_k(x_1^n)),$$

где кодирование f *построено по схеме* LZP.

Доказательство. Пусть алгоритм LZP разбивает слово $x_1^n \in A^n$ на подслова $\sigma_1, \sigma_2, \ldots, \sigma_m$. Поскольку для каждого целого $i > 0$ последовательность X_i^∞ непериодическая, то процедура выделения алгоритмом LZP следующего подслова всегда конечна. Пусть кодирование $f : A^* \to E^*$ построено по схеме LZP. Согласно лемме 2 и (10) для каждого целого $k \geqslant 0$ имеем

$$|f(x_1^n)| - nH_k(f, x_1^n) \leqslant 6\sum_{i=1}^{m} \log|\sigma_i| + Cm, \tag{21}$$

где $C > 0$ зависит только от k и $|A|$. Из леммы 1 имеем

$$nH_k(f, x_1^n) \geqslant \sum_{i=1}^{m} \log|\sigma_i| - mk\log|A| - |A|^{2k}\log|A|$$

и

$$m \leqslant nH_k(f, x_1^n) + |A|^k \log|A|.$$

Из (21) и двух последних неравенств следует утверждение теоремы 2.

ЛИТЕРАТУРА

1. **Гоппа В. Д.** Коды и информация // Успехи мат. наук. 1984. Т. 39, вып. 1. С. 77–120.
2. **Кадач А. В.** Эффективные алгоритмы неискажающего сжатия текстовой информации: Дис. ... канд. физ.-мат. наук. Новосибирск: Ин-т систем информатики им. А. П. Ершова, 1997.

3. **Кричевский Р. Е.** Сжатие и поиск информации. М.: Радио и связь, 1989.

4. **Левенштейн В. И.** Об избыточности и замедлении разделимого кодирования натуральных чисел // Проблемы кибернетики. М.: Наука, 1968. Вып. 20. С. 173–179.

5. **Brent R. P.** A linear algorithm for data compression // Austral. Comput. J. 1987. V. 19, N 2. P. 64–68.

6. **Elias P.** Universal codeword sets and representations of integers // IEEE Trans. Inform. Theory. 1975. V. 21, N 2. P. 194–203.

7. **Kieffer J., Finamore W., Nunes P.** A class of noiseless data compression algorithms based on Lempel — Ziv parsing trees // Proc. IEEE Intern. Symp. on Inform. Theory. Piscataway, NJ, 1994. P. 6.

8. **Lounchard G., Szpankowski W.** On the average redundancy rate of the Lempel — Ziv code // IEEE Trans. Inform. Theory. 1997. V. 43, N 1. P. 1–7.

9. **Miller V. S., Wegman M. N.** Variations on a theme by Ziv and Lempel // Combinatorial Algorithms on Words. Berlin: Springer-Verl., 1985. P. 131–140.

10. **Plotnick E., Weinberger M. J., Ziv J.** Upper bounds on the probability of sequences emitted by finite-state source and on the redundancy of the Lempel — Ziv algorithm // IEEE Trans. Inform. Theory. 1992. V. 38, N 1. P. 66–72.

11. **Rodeh M., Pratt V. R., Even S.** Linear algorithm for data compression via string matching // J. Assoc. Comput. Mach. 1981. V. 28, N 1. P. 16–24.

12. **Savari S. A.** Redundancy of the Lempel — Ziv incremental parsing rule // IEEE Trans. Inform. Theory. 1997. V. 43, N 1. P. 8–16.

13. **Savari S. A.** Redundancy of the Lempel — Ziv string matching code // IEEE Trans. Inform. Theory. 1998. V. 44, N 2. P. 787–792.

14. **Storer J. A., Szymansky T. G.** Data compression via textual substitution // J. Assoc. Comput. Mach. 1982. V. 25, N 4. P. 928–951.

15. **Welch T. A.** A techcnique for high-perfomance data compression // IEEE Comput. 1984. V. 17, N 6. P. 8–19.

16. **Wyner A. D., Ziv J.** Fixed data base version of the Lempel — Ziv algorithm // IEEE Trans. Inform. Theory. 1991. V. 37, N 3. P. 723–731.

17. **Wyner A. D., Wyner A. J.** Improved redundancy of a version of the Lempel — Ziv algorithm // IEEE Trans. Inform. Theory. 1995. V. 41, N 3. P. 723–731.

18. **Wyner A. J.** The redundancy and distribution of phrase lengths of the fixed-database Lempel — Ziv algorithm // IEEE Trans. Inform. Theory. 1997. V. 43, N 5. P. 1452–1464.

19. Ziv J., Lempel A. A universal algorithm for sequential data compression // IEEE Trans. Inform. Theory. 1977. V. 23, N 3. P. 337–343.

20. Ziv J., Lempel A. Compression of individual sequences via variable-length coding // IEEE Trans. Inform. Theory. 1978. V. 24, N 5. P. 530–536.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия

Статья поступила
2 февраля 1999 г.