главные модули при $s \leqslant k-1$. Поэтому найдется R-комплекс C_*, гомотопически эквивалентный $C_*(\widetilde{W}, t\widetilde{V})$, который до размерности $k-1$ имеет стандартный вид, а $C_k = Z_k \oplus T_k$ и $\partial|_{T_k}$ – мономорфизм (см. лемму С). Реализуем этот комплекс некоторой функцией Морса на кобордизме (V, W, tV) (см. [5]). Рассмотрим соответствующее разложение на ручки многообразия W, и пусть Y_0 есть объединение tV всех ручек индексов $\leqslant k-1$ и ручек, соответствующих T_k. Тогда вложение $tV^- \cup \cup Y_0 \to V^-$ индуцирует изоморфизм в гомологиях при $s \leqslant k-1$ (при $s < k-1$ это очевидно, а при $s = k-1$ вытекает из того, что в модуле $C_k(\widetilde{V}^-, (tV^- \cup Y_0)^\sim)$ граничный оператор нулевой). С другой стороны, вложение $tV^- \to tV^- \cup Y_0$ индуцирует изоморфизм в k-мерных гомологиях (ибо $H_{k+1}(\widetilde{Y}_0, t\widetilde{V}) = 0$ и в $C_k(\widetilde{Y}_0, t\widetilde{V})$ граничный оператор мономорфен), стало быть, $H_k((tV^- \cup Y_0)^\sim) \approx H_k(t\widetilde{V}^-) \approx H_k(\widetilde{V}^-)$. Рассмотрим элемент $\alpha \in H_k(t\widetilde{V}^-)$. Этот элемент лежит в образе ∂: $H_{k+1}(\widetilde{V}^-, (tV^- \cup Y_0)^\sim) \to H_k((tV^- \cup Y^0)^\sim)$. Заметим, что $H_s(\widetilde{V}^-, (tV^- \cup Y_0)^\sim) = 0$ при $s \leqslant k-1$, поэтому (сильная теорема Гуревича) $\pi_{k+1}(\widetilde{V}^-, (tV^- \cup Y_0)^\sim)$ эпиморфно отображается на $H_{k+1}(\widetilde{V}^-, (tV^- \cup Y_0)^\sim)$. Рассмотрим соответствующий элемент $\overline{\alpha} \in \pi_{k+1}(V^-, tV^- \cup Y_0)$. По теореме Хадсона [7] этот элемент может быть реализован вложением a: $(D^{k+1}, S^k) \subset (V^-, tV^- \cup Y_0)$ в случае $k+1 \leqslant n-3$ и $\pi_j(V^-, tV^- \cup Y_0) = 0$ при $j \leqslant 2(k+1) - n + 1$, что также имеет место при $k \leqslant n-4$. Приклеим теперь по вложению a ручку σ, рассмотрим $Z = tV^- \cup Y_0 \cup \sigma$. Ясно, что $V_0 = \partial Z$ – секущее многообразие и V_0^- обладает требуемыми свойствами.

Автор очень признателен Ж.–К. Сикораву за сообщение о его результатах (теоремы 1 и 3) и за ценные обсуждения.

Институт химической физики
Академии наук СССР
Москва

Поступило
29 XII 1987

ЛИТЕРАТУРА

1. *Новиков С.П.* – ДАН, 1981, т. 260, № 1, с. 31–35. 2. *Новиков С.П.* – УМН, 1982, т. 37, № 5, с. 3–49. 3. *Фарбер М.Ш.* – Функц. анализ и его прилож., 1985, т. 19, вып. 1, с. 49–59. 4. *Бурбаки Н.* Коммутативная алгебра. М.: Мир, 1971. 5. *Шарко В.В.* K-теория и теория Морса I. Препринт 86.39. Киев, ИМАН УССР, 1986. 6. *Farrell F.T.* – Indiana Univ. Math. J., 1971, vol. 21, № 4, p. 315–346. 7. *Hudson J.F.P.* – Proc. Cambr. Phil. Soc., 1972, vol. 72, № 1, p. 11–20.

УДК 519.2 МАТЕМАТИКА

Б.Я. РЯБКО

БЫСТРЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ КОД

(Представлено академиком А.П. Ершовым 10 XII 1987)

1. Рассматривается задача побуквенного кодирования слов $x = x_1 x_2 \ldots x_N$, $N \geqslant 1$, в некотором конечном алфавите A. Наилучшим побуквенным кодом слова x является код Хаффмена [1], однако для его построения кодируемое слово x необходимо просмотреть и вычислить частоту встречаемости букв из A в x. В работе предлагается код, средняя длина кодового слова которого лишь на константу превышает длину слова кода Хаффмена. Однако предлагаемый код не требует пред-

варительного просмотра слова x. Время кодирования и декодирования одной буквы у этого кода близко к минимальному и на порядок меньше, чем у кода Галлагера–Кнута [2, 3]. Эффективность предлагаемого кода на порядок выше, чем у других быстрых методов, разработанных автором [4], Дж. Бентли и Р. Тарьяном с соавторами [5] и П. Элайесом [6].

2. В статье рассматриваются последовательные коды, которые по определению должны удовлетворять следующему условию: любое слово $x = x_1 x_2 \ldots x_N$, $N \geqslant 1$, необходимо кодировать и декодировать побуквенно за один просмотр, т.е. так, что при $i = 1, 2, \ldots, N$ код буквы x_i не зависит от $x_{i+1} x_{i+2} \ldots x_N$, но, возможно, зависит от $x_1 x_2 \ldots x_{i-1}$ *. Рассматриваются только последовательные коды, ставящие в соответствие буквам $x_1, x_2, \ldots$ слова из двоичного алфавита; перенесение результатов на общий случай очевидно.

Эффективность сжатия последовательных кодов оценивается двумя характеристиками [5, 6]: средней избыточностью, определяемой как разность между средней длиной кодового слова и энтропией Шеннона в случае, когда буквы $x_1, x_2, \ldots$ порождаются бернуллиевским источником, и комбинаторной избыточностью. Она определяется как разность между средним числом двоичных символов в кодовом слове, приходящихся на одну букву слова x, и эмпирической энтропией x, определяемой равенством

$$\hat{H}(x) = - \sum_{a \in A} (p(a, x)/N) \log (p(a, x)/N),$$

где $p(a, x)$ – число встреч $a \in A$ в слове $x = x_1 \ldots x_N$ (здесь и ниже $\log x \equiv \log_2 x$, $0 \log 0 = 0$). Сложность последовательного кода оценивается размером (в битах) программы кодера и декодера при реализации их на вычислительной машине, а также максимальным временем кодирования и декодирования одной буквы слова x. Время измеряется числом операций над однобитовыми словами. Формально время и объем программы оцениваются при реализации кодера и декодера на вычислительной машине с произвольным доступом к памяти при логарифмическом весовом критерии [9].

3. Одним из первых последовательных кодов был метод "стопка книг", предложенный в работе автора [4] в 1980 г. В 1986–1987 гг. он был переоткрыт в работах [5, 6]**, причем в [5] предложен алгоритм кодирования и декодирования существенно более простой, чем в [4].

Характеристики кода "стопка книг" при организации кодирования и декодирования по методу из [5] приведены в первой строке табл. 1. В [6] П. Элайес предложил также так называемый интервальный код, асимптотические характеристики которого те же, что и у стопки книг (см. табл. 1). Д. Кнут [2], основываясь на конструкции Р. Галлагера [3], разработал последовательный код, избыточность которого асимптотически меньше, чем у кодов "стопка книг" и интервального. В методе Галлагера–Кнута буквы слова $x_1 x_2 \ldots x_N$, $N \geqslant 1$, кодируются последовательно, причем перед кодированием i-й буквы, $1 < i \leqslant N$, подсчитывается частота встречаемости всех букв $a \in A$ в слове $x_1 x_2 \ldots x_{i-1}$ и строится код

---

* Формально такие коды описаны в [2, 6] и названы там "on-line"; близкие коды – автоматные и последовательные – рассматривались в [7, 8] соответственно. Рассматриваемые коды реализуются не конечными автоматами, а термин "линейный код", близкой к "on-line", широко используется в теории помехоустойчивого кодирования. Поэтому отдано предпочтение названию "последовательный".

** В [6] П. Элайес указывает, что этот код был известен ему с начала 80-х годов и использовался при занятиях со студентами Гарвардского университета.

Таблица 1

Характеристики известных последовательных кодов при большой мощности алфавита

№	Название кода	Избыточность (средняя и комбинаторная)	Объем памяти кодера и декодера (в битах)	Максимальное время кодирования и декодирования одной буквы
1	Стопка книг [4–6]	$\log\log \mid A \mid + O(1)$	$O(\mid A \mid \log \mid A \mid)$	$O(\log^2 \mid A \mid)$
2	Интервальный код [6]	$\log\log \mid A \mid + O(1)$	$O(\mid A \mid \log \mid A \mid)$	$O(\log^2 \mid A \mid)$
3	Код Галлагера–Кнута [2, 3]	$O(1)$	$O(\mid A \mid \log \mid A \mid)$	$O(\mid A \mid \log \mid A \mid)$
4	Частотный код	$O(1)$	$O(\mid A \mid \log \mid A \mid)$	$O(\log^2 \mid A \mid)$

Хаффмена [1] для алфавита A, оптимальный для этих частот; он и используется при кодировании x_i. В [2] указывается также, что данный метод можно модифицировать: при кодировании x_i оценивать частоты буквы $a \in A$ не по всему слову $x_1 x_2 \ldots x_i$, а только по его части, предшествующей x_i, т.е. по $x_{i-k} x_{i-k+1} \ldots x_{i-1}$, где $k > 0$ – некоторое целое число, параметр метода. Характеристики этого метода приведены в табл. 1. Наконец, необходимо отметить ряд методов, возникших в теории кодирования, которые можно использовать и как последовательные коды. К ним относятся "арифметический код" [10, 11] и "обобщенный код Шеннона" [12]. Эти коды зависят от ряда параметров. При значениях параметров, дающих среднюю избыточность $O(1)$ и минимизирующих время кодирования, характеристики этих методов такие же, как у кода Галлагера–Кнута.

В данной работе предлагается код, названный частотным, соединяющий достоинства методов из [2] и [4–6]: его избыточность минимальна, как у метода Галлагера–Кнута, а время обработки одной буквы $O(\log^2 |A|)$, как у "стопки книг", что близко к $O(\log |A|)$ – очевидной нижней границе.

В конструкции частотного кода используется ряд известных методов и идей теории кодирования. Ключевую роль играет результат Р.Е. Кричевского [13, 14], показывающий, что для эффективного кодирования неизвестного источника достаточно иметь выборку наблюдений над источником объема $O(|A|)$ (при больших $|A|$). В частотном коде, так же как в методе Галлагера–Кнута, оценивается частота встречаемости букв алфавита A по некоторой обновляющейся выборке, но для построения кода используются смещенные оценки частот, что позволяет за счет небольшого увеличения избыточности существенно понизить трудоемкость кодирования. Смещенные оценки введены в работах [13, 14] при построении кодов на основе наблюдений. При построении частотного метода используется алфавитный код Гильберта–Мура [15], применявшийся также в [8, 12]. Для построения частотного метода разработан алгоритм быстрой реализации алфавитного кода, время работы которого меньше по порядку, чем в [12]: у предлагаемого алгоритма $O(\log |A|)$ операций суммирования, для алгоритма из [12] – $O(|A|)$ операций суммирования (при больших $|A|$).

4. Перейдем к описанию частотного кода. Пусть буквы из A как-нибудь занумерованы: $A = \{a_1, \ldots, a_n\}$, $n = |A|$, дано $N \geqslant 1$ и $x = x_1 x_2 \ldots x_N \in A^N$. Определим $l = \ulcorner \log n \urcorner$, $\mathcal{L} = 2^l$. Для описания частотного кода удобно продлить x на $\mathcal{L}$ букв влево: $x_0 = a_1$, $x_{-1} = a_2, \ldots, x_{-n+1} = \ldots = x_{-L+1} = a_n$. Перед кодированием каждой буквы x_i, $i = 1, \ldots, N$, оцениваются частоты всех букв из A в подслове $x_{i-1} \ldots$

$\dots x_{i-1}$; обозначим их через $p(a, i)$, $a \in A$. Затем находятся величины

$$\hat{P}(a, i) = p(a, i) + 1, \quad Q(a_j, i) = 2 \cdot \sum_{k=1}^{j-1} \hat{P}(a_k, i) + \hat{P}(a_j, i),$$
$$(1) \qquad m(a_j, i) = 2 + l - \lfloor \log \hat{P}(a_j, i) \rfloor, \quad j = 1, 2, \dots, n.$$

Буква x_i кодируется словом, образованным первыми $m(x_i, i)$ знаками двоичного разложения числа $Q(x_i, i)$. Рассмотрим пример. Пусть $A = \{a_1, \dots, a_8\}$, $x = x_1 x_2 \dots \dots x_{10} = a_6 a_6 a_6 a_7 a_6 a_6 a_6 a_6 a_6 a_6$. Это x малоинформативно, явно преобладает буква a_6. Частотный код это "учтет": согласно (1) первая буква $x_1 = a_6$ кодируется словом длины 4, а последняя буква $x_{10} = a_6$ кодируется словом длины 2. Непосредственное построение кода по формулам (1) требует при вычислении $Q(a_L, i)$ L операций суммирования l разрядных чисел, т.е. всего $O(L \cdot l) = O(n \log n)$ операций над битами. Для уменьшения сложности вычисления величины $Q(a_j, i)$, $j = 1, \dots, n$, разработан специальный алгоритм, использование которого и делает код "быстрым". При этом в памяти машины хранятся не только величины $\hat{P}(a_j, i)$, $j = 1, 2, \dots, L$, но и суммы их пар $(\hat{P}(a_1, i) + \hat{P}(a_2, i))$, $(\hat{P}(a_3, i) + \hat{P}(a_4, i))$, $\dots \dots$, $(\hat{P}(a_{L-1}, i) + \hat{P}(a_L, i))$, суммы четверок:

$$\sum_{j=1}^{4} \hat{P}(a_j, i), \quad \sum_{j=5}^{8} \hat{P}(a_j, i), \dots, \sum_{j=L-3}^{L} \hat{P}(a_j, i),$$

суммы восьмерок и т.д.; всего $L + L/2 + L/4 + \dots = 2L - 1$ чисел, каждое из которых записано в ячейке из $l + 1$ бит. Используя эти величины, любую сумму вида $\sum_{j=1}^{k} \hat{P}(a_j, i)$ можно вычислить, затратив не более чем $l - 1 = \lceil \log n \rceil - 1$ суммирование. Например,

$$\sum_{j=1}^{7} \hat{P}(a_j, i) = [\hat{P}(a_1, i) + \dots + \hat{P}(a_4, i)] + [\hat{P}(a_5, i) + \hat{P}(a_6, i)] + [\hat{P}(a_7, i)],$$

где каждая величина в квадратных скобках хранится в памяти машины, т.е. вместо шести операций суммирования необходимы только две. После кодирования i-й буквы необходимо изменить две частоты: $\hat{P}(x_i, i+1) = \hat{P}(x_i, i) + 1$, $\hat{P}(x_{i-L}, i+1) = \hat{P}(x_{i-L}, i) - 1$. Кроме того, необходимо также изменить две пары, содержащие эти частоты, две четверки и т.д. Всего требуется $l - 1$ раз прибавить единицу и $l - 1$ раз отнять 1. Таким образом, для кодирования и пересчета частот требуется $O(l) = O(\log n)$ операций суммирования чисел, длина каждого из которых равна $\lceil \log n \rceil + 1$ знаков в двоичной системе счисления, т.е. требуется всего $O(\log^2 n)$ операций над битовыми словами.

При декодировании используются те же величины $\hat{P}(a_j, i)$, $j = 1, 2, \dots, L$, и те же суммы их пар, четверок и т.п., что и при кодировании. Пусть буквы $x_1 x_2 \dots \dots x_{i-1}$ уже декодированы. Декодер считывает $l + 2$ очередных двоичных символа (пусть они $\alpha_1 \alpha_2 \dots \alpha_{l+2}$) и вычисляет число $W = \sum_{j=1}^{l+2} \alpha_j \cdot 2^{l+2-j}$, а затем находит k, удовлетворяющее неравенства

$$(2) \qquad 2 \sum_{j=1}^{k-1} \hat{P}(a_j, i) \leqslant W < 2 \sum_{j=1}^{k} \hat{P}(a_j, i).$$

Буква a_k и будет равна x_i. При помощи таблицы пар, четверок и т.д. значение k, удовлетворяющее (2), может быть найдено не более чем за l сравнений $(l+2)$-знаковых чисел, т.е. за $O(l^2) = O(\log^2 n)$ операций над однобитовыми словами. Таким образом, время кодирования и декодирования одной буквы для частотного кода равно $O(\log^2 n)$.

Пусть $\varphi(a/x)$ – кодовое слово, которое соответствует букве a при кодировании частотным кодом, если a следует за словом x. Обозначим через $E_\mu(f)$ среднее по мере μ функции f. Избыточность частотного кода характеризует следующая

Теорема. *Пусть $x = x_1x_2\ldots$ – бесконечное слово, μ – бернуллиевская мера. Тогда длина частотного кода на слове x превосходит эмпирическую энтропию не более чем на $3 + (1 + {}^1/_{12})\log e$, а средняя по мере μ длина кодового слова превосходит энтропию Шеннона $h(\mu)$ не более чем на $2 + \log e$:*

$$\lim_{n\to\infty}\left(\left(\frac{1}{n}\sum_{i=1}^{n}|\varphi(x_i/x_1x_2\ldots x_{i-1})|\right) - \hat{H}(x_1\ldots x_n)\right) \leqslant 3 + (1 + {}^1/_{12})\log e,$$

$$\lim_{n\to\infty} E_\mu\left(\frac{1}{n}\sum_{i=1}^{n}|\varphi(x_i/x_1\ldots x_{i-1})|\right) - h(\mu) \leqslant 2 + \log e.$$

Новосибирский электротехнический
институт связи
им. Н.Д. Псурцева

Поступило
30 XII 1987

ЛИТЕРАТУРА

1. *Huffman D.A.* – Proc. IRE, 1952, vol. 40, № 9, p. 1098–1101. 2. *Knuth D.E.* – J. algorithms, 1985, vol. 6, p. 163–180. 3. *Gallager R.G.* – IEEE Trans. Inform. Theory, 1978, vol. 24, p. 668–674. 4. *Рябко Б.Я.* Пробл. передачи информ., 1980, т. 16, № 1, с. 16–21. 5. *Benthy J.L., Sleator D.D., Tarjan R.E., Wei V.K.* – Commun. ACM, 1986, vol. 29, № 4, p. 320–330. 6. *Elias P.* – IEEE Trans. Inform. Theory, 1987, vol. 33, № 1, p. 3–10. 7. *Левенштейн В.И.* В кн.: Дискретная математика и математические вопросы кибернетики. М.: Наука, 1974, с. 207–305. 8. *Штарьков Ю.М.* – Пробл. передачи информ., 1987, т. 23, № 1, с. 3–17. 9. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с. 10. *Rissanen J.* – Acta Polytechn. Scand., 1979, vol. 31, p. 44–51. 11. *Witten I.H., Neal R.M., Cleary J.G.* – Commun. ACM, 1987, vol. 30, № 6, p. 520–540. 12. *Штарьков Ю.М.* – Пробл. передачи информ., 1984, т. 20, № 3, с. 3–16. 13. *Кричевский Р.Е.* – Там же, 1968, т. 4, № 3, с. 48–57. 14. *Кричевский Р.Е.* – Там же, 1975, т. 11, № 4. 15. *Gilbert E.N., Moore E.F.* – Bell Syst. Tech. J., 1959, vol. 38, № 4, p. 933–967.