Этот код является префиксным и содержит слова 00, 01, 1, которые несут информационную нагрузку. Указанные буквы по каналу связи поступают на вход дешифратора приемника, который после их дешифрации возбуждает соответствующие входы шифратора, и тот восполняет недостающие в переданных словах буквы, восстанавливая таким образом слова, поступающие от простейшего источника информации.

Рассмотренная процедура характеризует важный метод сжатия, который, однако, не является единственным. На практике часто кодирующее устройство решает задачу преобразования генерируемого простейшим источником сообщения в другое, более короткое. Исходное сообщение длины n ставит в соответствие каждому разрешенному слову другое меньшей длины n' и в общем случае с другим числом m' букв в алфавите А. К такому способу сжатия, например, относится нумерационное кодирование [2]. В результате на приемном конце происходят дешифрация слов длины n' и выбор с его помощью из шифратора сообщений длины n .

Емкость шифратора для этого случая

$$G = N \log_m N . \quad (14)$$

Таким образом, проведенный анализ показывает, что для сжатия информации имеет значение характеристики не только источника информации, но и приемника. Причем эффект сжатия является результатом двустороннего взаимодействия источника и приемника информации.

SUMMARY

The paper has material on principles of information combinatory compression. For its analysis it is used combinatory information sources, each of which is of a finite predicate. The estimation of compression efficiency for codes, been formed different predicates, is carried out for the purpose of searching optimal values.

СПИСОК ЛИТЕРАТУРЫ

1. Березюк И.Т. и др. Кодирование информации (двоичные коды). – Харьков: Вища школа, 1978. - 252 с.
2. Амелькин В.А. Методы нумерационного кодирования. – Новосибирск: Наука, 1986.

Поступила в редакцию 11 мая 2006 г.

УДК 621.038

МЕТОД СЖАТИЯ ДВОИЧНЫХ КОДОВ НА ОСНОВЕ БИНОМИАЛЬНЫХ ЧИСЕЛ

В. Б. Чередниченко, ст. преподаватель
Харьковский национальный университет внутренних дел

В статье излагаются процедуры сжатия двоичных кодов, в которых используются методы двоичного биномиального счета. Приведены алгоритмы преобразований, которые отличаются простотой, надежностью и адаптивностью. Они иллюстрируются примерами и таблицами.

ВВЕДЕНИЕ

Непрерывный рост информационных потоков в системах связи и компьютерных сетях делает весьма важной задачу сжатия данных. Современные методы компрессии двоичных кодов достаточно эффективны, однако они не в полной мере отвечают все возрастающим требованиям. Такая задача остается в настоящее время актуальной, поскольку в ней играют роль не только коэффициент сжатия, но и простота алгоритмов, время сжатия, количество аппаратурных затрат, помехоустойчивость и надежность. Кроме того, сейчас к алгоритмам сжатия предъявляются дополнительные требования по реализации защиты информации от несанкционированного доступа. Поэтому активное продолжение работ в области сжатия двоичных кодов продолжает оставаться актуальным.

Для сжатия разработаны и используются различные методы сжатия – от применения методов Шеннона, Фано, Хаффмана, до арифметического кодирования и других. Однако существенную роль в методах преобразования кодов играет специфика сжимаемой информации. Например, задачи сжатия телевизионных изображений имеют существенное отличие от методов обработки текстовых сообщений, экономической информации, таблиц или графиков и т. д. Существующие теоретические подходы и методы решения не всегда достаточно эффективно справляются с вышеперечисленными и иными специфическими задачами сжатия.

В данной работе рассматриваются теория и практика сжатия двоичной информации без потерь на основе биномиальных чисел. Суть такого сжатия состоит в нумерации двоичных последовательностей с помощью биномиальных чисел [1]. Этот подход отличается довольно простыми алгоритмами и несложной аппаратурной реализацией, что повышает надежность преобразования кодов при сравнительно высоком коэффициенте сжатия. Одновременно он обладает потенциальной возможностью защиты информации от несанкционированного доступа. В рамках этого метода сжатия можно использовать любые современные способы шифрования такой информации.

Целью данной работы является разработка алгоритмов обработки данных с анализом их эффективности на основе процедур счета биномиальных чисел.

ОСНОВНАЯ ЧАСТЬ

Биномиальные числа – это двоичные числа длиной n, имеющие количество единиц, не превышающее некоторую величину k. Они задаются системой кодообразующих ограничений, разбивающих кодовые комбинации на два непересекающихся класса. Комбинации первого класса содержат k единиц и заканчиваются на 1, в них количество нулей l находится в пределах $0 \le l \le (n-k-1)$, их длина r принимает значения $k \le r \le n-1$. Комбинации второго класса содержат $l = n \cdot k$ нулей и заканчиваются на 0, они имеют количество единиц q в пределах $0 \le q \le k-1$ и длину r в промежутке $n-k \le r \le n-1$. Общее количество кодовых комбинаций равно числу сочетаний C_n^k [2]. Например, при значениях $n = 6$, $k = 4$ максимальная длина биномиальных кодовых комбинаций $r_{\max} = n-1 = 5$, максимальное число единиц в кодах равно 4, а максимальное число нулей $l_{\max} = (n-k) = 2$. Этим условиям удовлетворяет кодовая комбинация 0110, т. к. она содержит $n-k = 2$ нуля и заканчивается нулем. Аналогично комбинация 10111 содержит $k = 4$ единицы, заканчивается единицей и также является биномиальной. Длина обеих комбинаций находится в пределах от 1 до 5. В таблице 1 приведены все биномиальные кодовые комбинации с параметрами $n=6$, $k=4$ в возрастающем порядке от 0 до 14 ($C_6^4 = 15$).

Был предложен метод нумерации двоичных кодовых комбинаций в два этапа: сначала переход от исходных кодов к биномиальным числам, а затем перевод последних в двоичные номера [1], что иллюстрируется алгоритмом (рис.1).

Риунок1 -- Преобразование равновесного кода в двоичный номер

Этап преобразования равновесных двоичных кодовых комбинаций в биномиальные состоит в отбрасывании в равновесной комбинации единиц справа до первого появления нуля или отбрасывание нулей справа до первой единицы. Например, дана равновесная кодовая комбинация 101011 с n = 6, k = 4, тогда переход к соответствующей ей биномиальной комбинации будет: 101011 _ 1010. Если при этих параметрах имеется равновесная комбинация 111010, то соответствующая ей биномиальная комбинация будет 11101.

Этап преобразования полученного биномиального кода в его двоичный номер выполняется следующим образом. Обнуляется ячейка S, в которой затем накапливается результат суммирования двоичного счета. Одновременно в ячейку D заносится значение биномиального числа N, которое будет переводиться в двоичный код. Из биномиального числа, записанного в ячейку D, вычитаются единицы согласно алгоритму биномиального вычитающего счета до появления в ней нуля, одновременно производится добавление единиц в ячейку S. В момент, когда число в ячейке D станет равным нулю, кодовая комбинация, которая получена в S, будет соответствовать номеру преобразуемой биномиальной комбинации.

Процедура биномиального вычитающего счета из переменной D состоит в выполнении следующих операций. Находится младшая единица в биномиальной кодовой комбинации и преобразуется в нуль. Затем подсчитывается число нулей в полученной кодовой комбинации, которые стоят левее полученного нуля. Если число этих нулей равно $n - k - 1$, то полученная кодовая комбинация, после отбрасывания всех нулей правее полученного нуля, является искомой. Если же количество нулей левее полученного нуля меньше $n - k - 1$, то правее полученного нуля добавляются единицы до тех пор, пока их общее число не станет равным k.

Процедура биномиального вычитающего счета иллюстрируется с помощью таблицы 1, где биномиальное число 10110 (с параметрами n = 6, k = 4) преобразуется в двоичное число 111, равное десятичному 7. Приведенный метод используется для преобразования равновесных кодов. Предлагается применять его для нумерации любых двоичных кодовых комбинаций длиной n с количеством единиц i в них: $0 \le i \le n$.

Если подсчитать число единиц в такой двоичной последовательности, то ее можно считать равновесной кодовой комбинацией длиной n и с количеством единиц $k = i$. От такой кодовой комбинации можно с помощью описанных выше алгоритмов перейти к ее биномиальному коду, а от него - к номеру числа, что в совокупности является решением задачи сжатия.

Таблица 1 - Преобразование биномиального числа 10110 в двоичный код

Номер комбинации	Число в ячейке D	Число в ячейке S
7	10110	000
6	1010	001
5	100	010
4	01111	011
3	01110	100
2	0110	101
1	010	110
0	00	111

Для применения этого метода к любым двоичным числам необходимо на приемный вход передать информацию о количестве единиц в двоичной кодовой комбинации и ее номер, полученный в результате вышеописанных процедур преобразования и относящийся к i- множеству комбинаций, число элементов в которой равно C_n^i (длина комбинаций n задана заранее). Видоизмененный алгоритм преобразования двоичных кодовых комбинаций приведен на рис. 2.

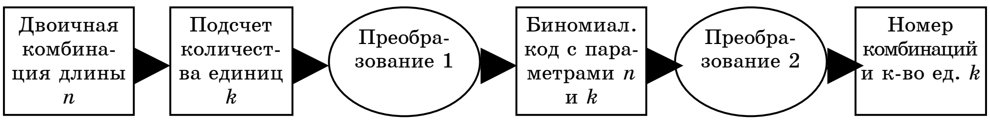

Рисунок. 2 - Преобразование двоичного кода длиной n *в двоичный номер*

Для оценки эффективности такого метода сжатия рассмотрим его характеристики. Очевидно, что после преобразования длина номера кодовой комбинации γ_i может быть определена по формуле

$$\gamma_i = \log_2 C_n^i \,. \tag{1}$$

Поскольку к каждой комбинации, кроме ее номера, нужно добавлять информацию о числе единиц q в ней, то ее длина после сжатия должна увеличиться на величину

$$q = \log_2 n \,. \tag{2}$$

Длина результирующей кодовой комбинации l_i равна

$$l_i = \gamma_i + q \,, \tag{3}$$

где слагаемые определены в формулах (1) и (2).

Например, имеется исходная двоичная кодовая комбинация длиной n = 64 разряда, в которой имеется i = 8 единиц. Применяем к ней два этапа преобразования, описанные выше, и получаем ее номер длиной $\gamma_i = \log_2 C_{64}^8$ = 32, данные о количестве единиц в ней $q = \log_2 8 =$ 3 и результирующую длину l_i =35.

На практике кодовые последовательности появляются с различной вероятностью p_i, тогда средняя длина номеров после сжатия двоичных кодовых комбинаций равна

$$\gamma_{cp} = \sum_{0}^{n} p_i \gamma_i \ . \qquad (4)$$

Средний коэффициент сжатия Y_{cp} в результате преобразования последовательности кодовых комбинаций вычислим по формуле

$$Y_{cp} = \frac{\gamma_{cp} + q}{n}, \qquad (5)$$

где слагаемые определяются по формулам (1), (2), (4).

Чем меньше Y_{cp}, тем лучше сжатие.

Выполнены расчеты среднего коэффициента сжатия Y_{cp} по формуле (5) для кодов длиной $n = 16$, 32, 64 и 128 разрядов и различном количестве единиц i в кодах. При этом задавались шестью вариантами распределения вероятностей появления кодовых комбинаций p_i (варианты одинаковы для каждого n). Примеры расчета коэффициента сжатия Y_{cp} для $n = 16$ и 128 приведены в табл. 2. В таблице 3 даны результирующие данные коэффициента сжатия Y_{cp} для n = 16, 32, 64 и 128 и величина обратной ему кратности сжатия $S_{cp} = \dfrac{1}{Y_{cp}}$.

Таблица 2 - Расчет среднего коэффициента сжатия Y_{cp} при различных вариантах вероятностей p_i

	Y cp	Вар. \ k	0	1	2	3	4	5	6	7	8	9	10
16		$\log_2 C_n^k$	0,0	4,0	6,9	9,1	10,8	12,1	13,0	13,5	13,7	13,5	13,0
	0,275	**p=var 1**	**0,9**	**0,1**									
q=0,250		p*log C_n^k	0,0	0,4									
	0,318	**p=var 2**	**0,8**	**0,1**	**0,1**								
		p*log C_n^k	0,0	0,4	0,7								
	0,400	**p=var 3**	**0,6**	**0,2**	**0,1**	**0,1**							
		P *log C_n^k	0,0	0,8	0,7	0,9							
	0,493	**p=var 4**	**0,4**	**0,3**	**0,1**	**0,1**	**0,1**						
		p*log C_n^k	0,0	1,2	0,7	0,9	1,1						
	0,675	**p=var 5**	**0,2**	**0,2**	**0,15**	**0,15**	**0,1**	**0,1**	**0,1**				
		p*log C_n^k	0,0	0,8	1,0	1,4	1,1	1,2	1,3				
	0,853	**p=var 6**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	
		p*log	0	0,4	0,7	0,9	1,1	1,2	1,3	1,3	1,4	1,3	
128		$\log_2$	0,0	7,0	13,0	18,4	23,3	28,0	32,3	36,5	40,4	44,1	47,7
	0,058	**p=var 1**	**0,9**	**0,1**									
q=0,055		p*log	0,0	0,4									
	0,063	**p=var 2**	**0,8**	**0,1**	**0,1**								
		p*log C_n^k	0,0	0,4	0,7								
	0,073	**P=var 3**	**0,6**	**0,2**	**0,1**	**0,1**							
		p*log C_n^k	0,0	0,8	0,7	0,9							
	0,085	**p=var 4**	**0,4**	**0,3**	**0,1**	**0,1**	**0,1**						
		p*log C_n^k	0,0	1,2	0,7	0,9	1,1						
	0,108	**p=var 5**	**0,2**	**0,2**	**0,15**	**0,15**	**0,1**	**0,1**	**0,1**				
		p*log C_n^k	0,0	0,8	1,0	1,4	1,1	1,2	1,3				
	0,130	**p=var 6**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	**0,1**	
		p*log C_n^k	0,0	0,4	0,7	0,9	1,1	1,2	1,3	1,3	1,4	1,3	

Таблица 3 - Средние коэффициенты сжатия Y_{cp} и кратности сжатия S_{cp} при различных вариантах вероятностей p_i

Средний коэффициент сжатия Y_{cp}				
n	**16**	**32**	**64**	**128**
p=var 1	0,275	0,169	0,100	0,058
p=var 2	0,318	0,190	0,111	0,063
p=var 3	0,400	0,231	0,131	0,073
p=var 4	0,493	0,278	0,154	0,085
p=var 5	0,675	0,369	0,200	0,108
p=var 6	0,853	0,458	0,245	0,130
q	*0,250*	*0,156*	*0,094*	*0,055*

Средняя кратность сжатия $S_{cp} = 1/Y_{cp}$				
n	**16**	**32**	**64**	**128**
p=var 1	3,64	5,93	10,00	17,30
p=var 2	3,14	5,25	9,03	15,82
p=var 3	2,50	4,32	7,62	13,61
p=var 4	2,03	3,60	6,47	11,76
p=var 5	1,48	2,71	5,00	9,28
p=var 6	1,17	2,18	4,09	7,69

Из таблицы 2 можно сделать вывод, что чем больше вероятности появления кодовых комбинаций с малым количеством единиц i, тем лучше сжатие, и наоборот. Таблица 3 показывает, что сжатие тем эффективнее, чем длиннее кодовые комбинации.

Анализ формулы среднего коэффициента сжатия Y_{cp} (5) позволяет сделать вывод, что не исключены такие параметры кодовых комбинаций, при которых результирующая двоичная последовательность будет длиннее исходной, т. е. $Y > 1$. Для определения интервалов сжатия были вычислены коэффициенты сжатия равномерных кодов длиной n в зависимости от количества в них единиц k $(0 \le k \le n)$ по формуле (5), результаты приведены в таблице 4 и на графике рис. 3.

Таблица 4 - Интервалы сжатия равновесных кодов для

n = 16, 32, 64 и 128

n \ k		***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***20***	***23***	***32***	***44***
8	C_n^k	1	8	28	56	70	56	28	8	1												
	$\log C_n^k$	0	3	4,81	5,81	6,13	5,81	4,81	3,00	0,00												
	Y	0,38	0,75	**0,98**	1,10	***1,14***	1,10	**0,98**	0,75	0,38												
16	***Y***	0,25	0,50	0,68	0,82	**0,93**	1,01	1,06	1,09	***1,10***	1,09	1,06	1,01	**0,93**	0,82	0,68	0,50	0,25				
32	***Y***	0,16	0,31	0,44	0,54	0,63	0,71	0,77	0,83	0,89	**0,93**	0,97	1,00	1,02	1,04	1,06	1,06	***1,07***	….	**0,93**	….	.....
64	***Y***	0,09	0,19	0,27	0,33	0,39	0,45	0,50	0,55	0,59	0,64	0,67	0,71	0,74	0,77	0,80	0,83	0,86	**0,94**	.....	***1,04***	**0,94**
128	***Y***	0,05	0,11	0,16	0,20	0,24	0,27	0,31	0,34	0,37	0,40	0,43	0,45	0,48	0,50	0,53	0,55	0,57				
128	***k***	***18***	***20***	***22***	***24***	***26***	***28***	***30***	***32***	***34***	***36***	***38***	***40***	***42***	***44***	***46***	***48***	.....	***64***	.....	***86***	.....
	Y	0,61	0,65	0,69	0,72	0,76	0,78	0,81	0,84	0,86	0,88	0,90	0,92	**0,94**	0,95	0,97	0,98		***1,02***		**0,94**	

Как видно из таблицы 4, при различных n сжатие кодовых комбинаций наблюдается в интервале от начальной величины до младшего порогового значения, а затем симметрично от старшего порогового значения до конечной величины k (интервалы сжатия и их пороговые значения выделены в таблице 4).

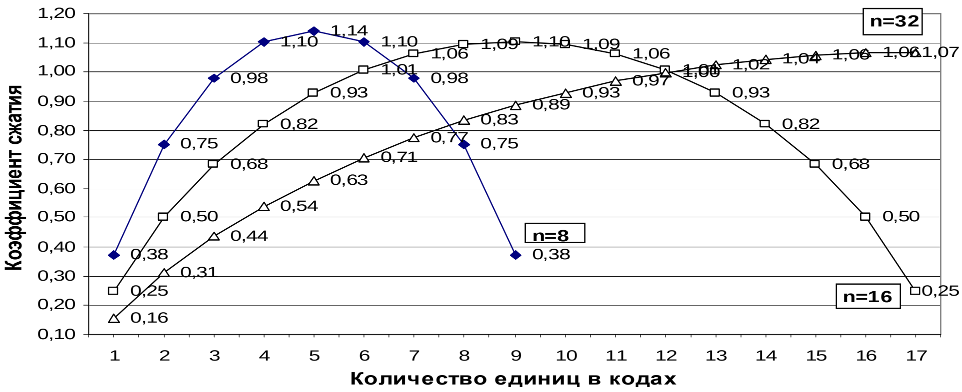

Рисунок 3 - Коэффициенты сжатия равновесных кодов для n = 16, 32, 64

Между пороговыми значениями находится внутренний интервал, в котором после преобразования происходит не сжатие, а удлинение кодовых комбинаций. Посередине этого интервала выделено максимальное значение длины преобразованного кода. Относительная величина интервалов наличия и отсутствия сжатия в зависимости от длины кодов приведена в таблице 5.

Таблица 5 - Пороги сжатия и интервал удлинения

Относительные интервалы сжатия				
n	Наличие сжатия			Отсутствие сжатия
8	0 - 0,25	0,75 – n		0,5
16	0 - 0,25	0,75 - n		0,5
32	0 - 0,31	0,69 - n		0,38
64	0 - 0,34	0,66 – n		0,31
128	0 - 0,36	0,64 - n		0,28

Из этой таблицы видно, что относительный размер начального и конечного интервалов сжатия увеличивается с удлинением кодов. Одновременно уменьшается относительная длина внутреннего интервала, где наблюдается увеличение длины преобразованного кода.

ВЫВОДЫ

Методы биномиального счета позволяют сжимать не только равновесные, но и любые двоичные коды длиной n. При этом используются простые алгоритмы, что позволяет реализовать их в аппаратном виде и повысить надежность работы. При таком методе к сжатой кодовой комбинации нужно добавить информацию о количестве единиц в ней. Эта информация является ключом к последующему восстановлению исходного сообщения, поэтому при ее шифровании можно надежно закрыть данные от несанкционированного доступа.

Сжатие тем лучше, чем выше вероятность появления кодовых комбинаций с малым количеством единиц или малым количеством нулей.

В кодовых комбинациях имеется внутренний интервал с центром в точке $k = n/2$, в котором сжатия не происходит, а его относительная длина изменяется от $0{,}5n$ при $n = 8$, до $0{,}28n$ при $n = 128$. Если в исходном коде количество единиц k соответствует этому внутреннему интервалу, то целесообразно передавать такую кодовую комбинацию без обработки, что улучшает характеристики сжатия. При этом весьма значительно уменьшается суммарное время преобразования, поскольку оно максимально в точке $k = n/2$ и быстро уменьшается при удалении от нее.

SUMMARY

The paper expounds compression of binary codes, in which use methods of binary binomial count. Algorithms of transformation are differ by simplicity and capability to adaptation. They are illustrated by examples and tables.

СПИСОК ЛИТЕРАТУРЫ

1. Борисенко А.А., Чередниченко В.Б. Нумерация равновесных кодов на основе биномиальных чисел // Право і безпека. – 2004. - № 3 – 4. - С.194 -197.
2. Борисенко А.А Биномиальный счет. Теория и практика: Монография. – Сумы: ИТД «Университетская книга», 2004. – 170 с.
3. В. Б. Чередниченко. Оценка быстродействия нумерации биномиальных чисел методом последовательного счета //АСУ и устройства автоматики. – 2005. -№ 131. - С.25- 30.
4. Чередниченко В.Б. Оценка времени преобразования равновесных кодов в биномиальные // Вісник СумДУ. – 2004 - № 12(71). - С. 113- 117.

Поступила в редакцию 12 мая 2006г.

УДК 519.17:519.816:004.715

РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ FLASH И JAVA APPLETS

Б.О. Кузиков, студ.; С.П. Шаповалов, доц.
Сумский государственный университет

Интеграция различных технологий является достаточно сложной проблемой. В статье исследованы возможности интеграции Flash и Java Applets, описаны механизмы возможного взаимодействия, рассмотрена их практическая реализация.

На рынке веб-приложений давно закрепились такие технологии, как Macromedia Flash и Java Applets. К безусловным преимуществам первого относится простота использования в сфере дизайна. Java же предоставляет богатые функциональные возможности, в частности в области вычислений и работы с различными серверами. В частности, в рамках лаборатории дистанционного обучения СумГУ, при подготовке учебного материала, Flash применяется для создания виртуальных лабораторных столов по ряду предметов, а на базе Java разрабатываются интерактивные тренажеры, типичными операциями для которых являются отображение и проверка формул, поддержка нескольких языков интерфейса, сохранение промежуточных результатов, связь с преподавателем.

Иногда возникают ситуации, когда нужно объединить наглядность представления данных со сложными математическими расчетами и гибким взаимодействием с сервером. Рассмотрим варианты возможного взаимодействия указанных технологий.

ПОСТАНОВКА ЗАДАЧИ

Следует понимать, что Java Applets, как и Flash, в разных системах имеет разную реализацию контейнеров-интерпретаторов, а их выполнение в контексте браузера налагает определенные ограничения в плане безопасности. Значит, прямое взаимодействие объектов на разных