Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ORDINAL COMPUTING WITH 2-ARY CODE IN COMBINATORICS CODING METHOD

[1] RUSHENG ZHU, [2] ZHONGMING YONG,

[1] Department of Information Science and Technology, Heilongjiang University, Harbin,China

[2] Department of Computer Science and Technology, Heilongjiang University, Harbin,China

E-mail: [1] hlju2012zrs@163.com , [2] yzm517517@yahoo.com.cn

ABSTRACT

All characters in any character sequence can constitute a dictionary space by permutation and combination and the character sequence can be replaced by the ordinal of this sequence in the dictionary space. This coding method is named combinatorics coding method. Combinatorics coding method is expatiated firstly in this paper and ordinal computing in different data system is analysized. Combinatorics coding method can be done by different code modes. In these code modes, combinatorics coding method with 2-ary code has particular property. Ordinal computing method with 2-ary code is advanced and the characteristics of ordinal with 2-ary code are researched. Combinatorics coding technology with 2-ary code has reference to the intensive study of the combinatorics coding method.

Keywords: *Data Coding, Source Coding, Combinatorics Coding, Ordinal, Max Ordina, Whole Max Ordinal*

1. INTRODUCTION

Today, the information technology is developing very much. As one of the basic tools and researched problems, data coding technology plays an important role in all kinds of studies and applications. It is well known about Huffman coding, algorithm coding and dictionary coding[1-3].

There are three embranchments in coding technology: source coding, channel coding and secrecy coding. The main task of source coding is to compress data. Channel coding is used to improve reliability of information transmission. Secrecy coding prevents the information from being filched and the used technology is encryption.

Data compression technology is of great significance to information disposing, data storage, and data transmission.

Combinatorics coding is a new data coding method based on Combinatorics. At first, Combinatorics coding is only used in researching data compression [4-7]. In fact, this kind of coding technology can be used in many fields such as data encryption and error detection/error correction. It is of great significance to information disposal, information storage and information transmission.

2. THE BASIC THEORY OF COMBINATORICS CODING

To the sequence which length is n, if the number of the different elements is m and the number of the i^{th} element is n_i. Thus the sum of the different permutation sequences that the different element in each sequence has the same number with this sequence can be expressed by formula (1):

$$P = n!/(\prod_{i=1}^{m}(n_i!)) \qquad (1)$$

All these permutation sequences can constitute a permutation space. This space is an ordered dictionary space. Any permutation sequence in this dictionary space has its position number. This position number is named ordinal.

The relation between the character sequence space and the corresponding ordinal space is utilized in combinatorics coding method. It is to say that all characters in any character sequence can constitute a dictionary by full permutation. Then combinatorics coding method comes into being. In fact, the ordinal of the sequence indicates the number of the other sequences in front of this sequence in the dictionary space.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

To the same sequence, the different ordinal can be obtained based on the different element order which has different elements. It is to say that the ordinal of the sequence is not exclusive. This element sequence depended on is named benchmark sequence.

Max ordinal of the sequence space can be computed through which the number of the sequences in the ordinal space subtracts one. When the number of each element in the sequence is equal, whole max ordinal is obtained. In the optimizing procedure of ordinal computing, ordinal computing is depended on the value of max ordinal and max ordinal computing is depended on the value of whole max ordinal[ ].

In combinatorics coding method, a benchmark sequence needs to be agreed on in advance between the coder and the decoder. When the coder works, the ordinal of the sequence can be computed out based on this benchmark sequence. When the decoder works, the sequence can be deduced by the ordinal based on the same benchmark sequence.

To the j^{th} element in the sequence which has n elements, if this element is the same as the i^{th} element in benchmark sequence, it is predicated that there are existed permutation and combination sequences before the current sequence involved i-1 elements. To these i-1 elements, the permutation and combination number of the x^{th} element (the element position in these i-1 elements is x) occupied the j^{th} position can be expressed by $S_{j,x}$. It is shown in equation (2).

$$S_{j,x} = C_{n-j}^{w_1} * C_{n-j-w_1}^{w_2} * C_{n-j-(w_1+w_2)}^{w_3} * \ldots\ldots$$
$$* C_{n-j-(w_1+w_2+\ldots+w_{x-2})}^{w_{x-1}} * C_{n-j-(w_1+w_2+\ldots+w_{x-1})}^{w_x - 1} * C_{n-j-(w_1+w_2+\ldots+w_x)}^{w_{x+1}}$$
$$* \ldots\ldots * C_{n-j-\sum_{q=1}^{ii-2} w_q}^{w_{ii-1}} * C_{n-j-\sum_{q=1}^{ii-1} w_q}^{ii} \qquad (2)$$

w expresses the number of each element. To the j^{th} element in the sequence which has n elements, the sequence sum involved i-1 elements before j^{th} the element can be computed by the formula (3).

$$\sum_{x=1}^{i-1} S_{j,x} \qquad (3)$$

At last, the position (ordinal) of the sequence which has n elements in the whole space can be expressed by the formula (4).

$$\sum_{j=1}^{n} \sum_{x=1}^{i-1} S_{j,x} \qquad (4)$$

The basic idea of combinatorics decoding is that suppose the element in the inspected position is a certain element, the corresponding permutation and combination value p_1 is computed out based on benchmark sequence according to this suppose. Compared p_1 with ordinal, if ordinal is not less than p_1 , then the permutation and combination value p_2 of the next element is computed out based on benchmark sequence. It needs to judge that ordinal is less than $p_1 + p_2$ or not. ……Until it finds out that ordinal is less than $p_1 + p_2 + \ldots\ldots + p_r$. This moment, the element corresponding p_r should be the element in the inspected position. At last, the new ordinal needs to be computed in order to deduce the next position element according to the formula (5):

new ordinal= old ordinal-($p_1 + p_2 + \ldots\ldots + p_{r-1}$) (5)

When the ordinal is 0, the algorithm is over. If there are some remainder data, these remainder data can be appended to the decoded sequence based on benchmark sequence.

In fact, the process of combinatorics decoding is the inverse process to combinatorics coding. Probing step by step method is adopted to parse correlative character. The most important step of probing method is also ordinal computing.

III. THE RELATION BETWEEN THE SPEED OF ORDINAL COMPUTING AND DATA SYSTEM

In the procedure of ordinal computing, the speed of ordinal computing can be affected by different data system. It adopts 2^{16} system to computing ordinal in the figure 1. In this example, there are 256 different elements in benchmark sequence (m =256). The higher data are stored in the left nodes and the lower data are stored in the right nodes. There are 102265 nodes in the ordinal of the sequence which length is 200k in 2^{16} system, this ordinal can be expressed by the figure 1:

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

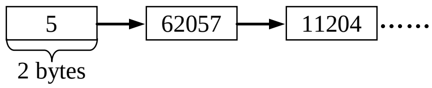

Fig.1 Ordinal Number in 2^{16} System

If it adopts 2^{32} system to computing the ordinal of the same sequence and the value range in each node of the ordinal is from 0~ 2^{16} -1 to 0~ 2^{32} -1, then the number of the nodes is halved. If the number is odd, for example, the nodes number of the ordinal in figure 1 is 102265 and it is an odd, so the value of the highest node is less than 256, this node can be extended to 4 bytes. The other nodes which the length of each node is 2 bytes can be merged into one node in pairs. Thus, in 2^{32} system, the number of the nodes of the ordinal is 51133. It is shown in figure 2.

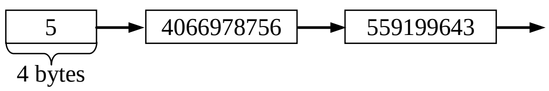

Fig.2 Ordinal Number in 2^{32} System

The experiment manifests that the speed of ordinal computing is advanced after adopting 2^{32} system, because the number of the nodes decreases and the length of the ordinal is halved, these mean that the computing times are halved in the computing procedure. To the main procedure of the computing, the assistant computing in the algorithm can be ignored. So the computing speed is advanced double.

IV. ORDINAL COMPUTING WITH 2-ARY CODE IN COMBINATORICS CODING METHOD

Sequence ordinal can be computed by the different code system. At present, the research on combinatorics coding method is mainly with 256-ary code. Because the smallest stored unit of the file information is byte. Then the number of the different elements in benchmark sequence is 2^8 . It is 256-ary code.

In this paper, combinatorics coding is researched with 2-ary code. It is to say that the number of the different elements in benchmark sequence is only two (the value of m is two and the elements are 0 and 1). Thus there are only two benchmark sequences: 0, 1 and 1, 0.

To the sequence 101011, there are two different elements: the number of 0 is 2 and the number of 1 is 4. The number of sequences which are all composed by 0(the number is 2) and 1(the number is 4) can be computed by the formula (1):

$$P = n!/(\prod_{i=1}^{m}(n_i!))$$

$$= 6!/(2!*4!)$$

$$=15$$

According to the order of 0 first and 1 last, the space of dictionary can be obtained as follows:

0) 001111 1) 010111 2) 011011 3) 011101 4) 011110

5) 100111 <u>6) 101011</u> 7) 101101 8) 101110 9) 110011

10)110101 11)110110 12) 111001 13) 111010 14)111100

In above instance, the number of the whole permutations about 0(the number is 2) and 1(the number is 4) is 15. Because it is counted from 0, the max ordinal is 14.

The ordinal of 101011 can be computed as follows:

Step 1: The first element of the sequence is 1, the number of sequences that have the form of 0* and that are in front of this sequence can be computed by the formula (2). Of course, the number of 0 is 2 and the number of 1 is 4 in these sequences.

$$C_5^1 C_4^4 = 5$$

Because the ordinal computing is based on 2-ary code, it only needs to compute once. This means that the formula (3) becomes vestigial with 2-ary code.

Step 2: The second element of the sequence is 0. There is no other element in front of this element in benchmark sequence. So there is no computation.

Step 3: The 3rd element is 1, then each sequence in front of this sequence has the form of 100*. In each sequence, the number of 0 is 2 and the number of 1 is 4 too. The number of these sequences is $C_3^3 = 1$.

Step 4: The fourth element is 0. There is no element in front of this element in benchmark sequence. So there is no computation.

Step 5: The fifth element and the sixth element are 1. But now there is no 0 element in the sequence. So it doesn't need to handle.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

At last, all data are added and the ordinal of the sequence is obtained. The ordinal is 6.

The ordinal of the sequence 101011 is computed out based on the benchmark sequence (0 first and 1 last). If the order of the benchmark sequence is contrary, 15 kinds of permutations can be obtained still. Of course, the order of these permutations is contrary compared with the case that the benchmark sequence is 0 and 1. It is shown as follows.

0) 111100 1) 111010 2) 111001 3) 110110 4) 110101

5) 110011 6) 101110 7) 101101 <u>8) 101011</u> 9)100111

10) 011110 11) 011101 12) 011011 13) 010111 14) 001111

In above instance, the ordinal of the sequence 101011 is 8 in the set of the whole permutations based on the benchmark sequence with '1' and '0'. Its computing procedure is similar to the case that the benchmark sequence is '0' and '1'.

To the same sequence 101011, the ordinal is 6 based on the benchmark sequence of 0 and 1, but the ordinal is 8 based on the benchmark sequence of 1 and 0. Luckily the sum of two ordinals equals to the max ordinal 14.

Thus it can be seen that, to the same sequence, its ordinal value is not unique. There are the close relation to the ordinal and the selection of benchmark sequence. To the same sequence, if the orders of two benchmark sequences are contrary, the sum of their ordinals is max ordinal.

The original sequence can be deduced based on frequency(such as the number of 0 is 2 and the number of 1 is 4)、benchmark sequence(such as 0 and 1) and ordinal (such as 6). The procedure of decoding is described as follow:

Step 1: Decode the first element of the sequence.

(1) If the first element of the sequence is 0, then there are one '0' and four '1' in the follow five elements. So the number of thus sequence can be computed as follow:

$$C_5^1 C_4^4 = 5$$

Because 5 is less than 6, the supposition is not right based on decoding idea. It is to say that the first element is not 0.

(2) If the first element of the sequence is 1, then there are two '0' and three '1' in the follow five elements. So the number of thus sequence can be computed as follow:

$$C_5^2 C_3^3 = 10$$

5+10=15，the sum is more than 6, so the first element of the sequence is 1.

Adjust ordinal: 6-5=1.

In fact, it can be directly judged that the first element must be '1' based on the result of (1) in the step 1. Because there are only two elements in benchmark sequence, if it is not '0', it must be '1'. It is to say that each step needs only computing once in 2-ary code of combinatiorics coding technology. It means that the speed of decoding in 2-ary code can be advanced.

Step 2: Decode the second element of the sequence.

(1) If the second element of the sequence (its form is 1*) is '0', then there are one '0' and three '1' in the follow four elements. So the number of thus sequence can be computed as follow:

$$C_4^1 C_3^3 = 4$$

4 is more than 1, the supposition is right based on decoding idea. It is to say that the second element is 0.

This moment the ordinal doesn't need to be adjusted.

Step 3: Decode the third element of the sequence.

(1) If the third element of the sequence (its form is 10*) is '0', then there are no '0' and three '1' in the follow three elements. So the number of thus sequence can be computed as follow:

$$C_3^3 = 1$$

1 is not more than 1, the supposition is not right based on decoding idea. It is to say that the third element is not 0.

It can be directly judged that the third element must be '1' based on the result of (1) in the step 3.

Adjust ordinal: 1-1=0.

This moment the ordinal is 0. Three elements in the front of the sequence are 101, there are remained elements: one '0' and two '1'. Based on the order of benchmark sequence, '011' is filled to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the tail of the decoded sequence. So the final decoded sequence can be obtained: 101011.

Thus the original sequence can be successfully decoded based on frequency、benchmark sequence and ordinal.

V. CHARACTERISTIC OF ORDINAL WITH 2-ARY CODE IN COMBINATORICS CODING METHOD

At present, the research on combinatorics coding is mainly aimed at 256-ary code [8-10]. But in relation to 256-ary code or other coding method, 2-ary code has its distinctive character.

At first, to the computing speed in combinatorics coding method, decoding speed is slower than coding speed. The more the number of different elements is, the slower the speed of decoding (relation to coding) is. It is apparent that the speed difference of coding and decoding is the smallest with 2-ary code.

Coding technology can be optimized in other existed coding method. The paper [11] and the paper [12] expatiate how to make combinatorics coding reduce computing speed from exponent level to n^2 level. In this paper, combinatorics computing becomes proportion computing. But at present, there is no good optimization method for decoding in combinatorics coding technology. Because decode method is to probe step by step. It means that if one element doesn't be decoded, then all elements follow this element can't be decoded. Thus merging computing can't be done. Compared the code/decode procedure with 256-ary code to the code/decode procedure with 2-ary code in this paper, it can be seen that the speed of decoding is almost equal to the speed of coding with 2-ary code, while the speed of decoding is slower more than the speed of coding with other code method.

Secondly, the result of existed experiment makes clear that the difference of the ordinal length and the sequence length with 2-ary code is smaller than the difference of the ordinal length and the sequence length with other code. This conclusion can be estimated through the formula (1).

For example, with 2-ary code, when the sequence length n is 8, it is to say that each element occupies 1 bit and the sequence occupies one byte, the whole max ordinal of the sequence can be computed by the follow procedure:

$$P = n!/((n/2)!*(n/2)!)$$

$$= 8!/(4!*4!)$$

$$= 8!/(4!*4!) = 70$$

But with 4-ary code, each element of the sequence occupies 2 bits, while the sequence itself still occupies one byte. Now, the length of the sequence is 4 and its whole max ordinal can be computed:

$$P' = n'!/((n'/4)!*(n'/4)!*(n'/4)!*(n'/4)!)$$

$$= 4!/((4/4)!*(4/4)!*(4/4)!*(4/4)!) = 24$$

$p > p'$, $\log(p) >= \log(p')$, so the length of p is not less than the length of p'. In the most case, the length of p is more than the length of p'.

VI. CONCLUSION

The ordinal computing method with 2-ary code in combinatorics coding technology is advanced and the characteristic of 2-ary code ordinal is researched in this paper. The research makes clear that the difference between coding speed and decoding speed is the smallest in binary combinatorics coding. Furthermore, the difference between ordinal length and sequence length with 2-ary code is smaller than that with other code. 2-ary code/decode technology is significant to research intensively on combinatorics code method.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] D. A. Huffman. A Method for the Construction of Minimum Redundancy Codes. Proceedings of the Institute of Radio Engineers, vol. 40, pp.1098-1101, Sept 1952,

[2] J.Rissanen, G.Langdon. Arithmetic Coding. IBM Journal of Research and Development,vol. 23,pp. 149-162,Feb. 1979.

[3] J. Ziv , A. Lempel. Compression of Individual Sequences via Variable Rate Coding. IEEE Transactions on Information Theory. Vol,24,pp.530-536, May 1978.

[4] Lu Jun, Liu Da-xin, Gao Yang, Xie Xin-qiang, Zhang wei-ran, Ma Yan-bin, "Study on frequency table of stochastic subsection file in constant grade compression method," Computer Engineering and Applications, vol. 44, pp. 175-177, Mar. 2008.

[5] Jun Lu, Daxin Liu, and Xinqiang Xie, "Selection of the Smallest Compression Subsection in Constant Grade Compression," Journal of Information & Computational Science, vol. 5, pp. 1545-1550, Apr. 2008.

[6] LU Jun, LIU Da-xin, CHEN Li-yan. "Compression method with constant degree based on permutation and combination,"Journal of Dalian Maritime University,vol 34(4), pp.28-32, 2008.

[7] Lu Jun, Liu DaXin,"Optimization of frequency table storage in Constant Grade Compression," 2009 International Forum on Information Technology and Applications (IFITA 2009), ChengDu, P. R. China, May, 2009, pp. 72–74.

[8] Lu Jun, Liu Da-Xin,"Study on Information Entropy of Combinatorics Coding," 2010 International Conference on Computer Design and Applications (ICCDA 2010), Qinhuangdao, P. R. China, June, 2010, pp. 105–108.

[9] Lu Jun, Liu Da-Xin,"Research on Parallel Technology within Section in Combinatorics Coding," The 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, P. R. China, October,2010, pp.252-255.

[10] Lu Jun, Liu Da-Xin,"Research on Whole Max Ordinal in Combinatorics Coding Technology," 2010 International Forum on Information Technology and Applications (IFITA 2010), Kunming, P. R. China, July, 2010, pp. 167–170.

[11] Lu Jun, Liu Daxin, "Optimization of Constant Grade Compression Method," Journal of Jiangsu University(Natural Science Edition). vol. 31, pp. 78-81. Jan 2010.

[12] Lu Jun, Wang Tong, Li Yibing, "Study on Optimization Technology in Computing Ordinal Number," Journal of Computers, vol. 5, pp. 210-217, Feb 2010.