UDC 681.3.04

DESCRIPTION AND APPLICATIONS OF BINOMIAL NUMERAL SYSTEMS

Olexiy Borysenko[1], Vyacheslav Kalashnikov[2], Nataliya Kalashnykova[3]

[1] Department of Electronics and Computing, Sumy State University, Rimsky-Korsakov str. 2, Sumy 40007, Ukraine;
electron@sumdu.edu.ua

[2] Department of Systems & Industrial Engineering, ITESM, Campus Monterrey,
Av. Eugenio Garza Sada 2501 Sur, Monterrey, N.L., 64849, Mexico;
kalash@itesm.mx

[3] Department of Physics and Mathematics (FICA), Universidad Autónoma de Nuevo León (UANL), Avenida Universidad S/N,
San Nicolás de los Garza, Monterrey, N.L., 66450, Mexico;
nkalash@einstein.fcfm.uanl.mx

Reviewer: Alexandr Kuznetsov Dr., Full Professor, V.N. Karazin Kharkiv National University, Svobody sq., 4,
Kharkov, 61022, Ukraine;
kuznetsov@karazin.ua

Received on April 2016

Abstract*. We develop a new class of positional numeral systems, namely the binomial ones, which form a subclass of generalized positional numeral systems (GPNS). The binomial systems have wide range of applications in the information transmission, processing, and storage due to their error-detection capabilities. In this paper, the binomial numeral systems are well-defined, their prefix and compactness properties are established. Algorithms of generating binomial coding words (non-uniform and uniform) are presented, as well as an enhanced procedure of construction of constant weight Boolean combinations based upon the non-uniform binomial coding words. The correctness of this procedure is established.*

Keywords*: generalized positional numeral systems, binomial numeral systems, constant weight codes.*

1 Introduction

Positional numeral systems are widely used in computing. More complicated numeral systems, in which the register's weight need not be equal to the power of the system's base (like, for example, in the binary or decimal system), have not been thoroughly studied as yet. Such generalized positional numeral systems (GPNS) may have quite useful properties, like being noise-proof, easy in generating permutations, etc. (see [1-3]). These properties allow one to exploit the GPNS to develop specialized digital de-vices with high computational speed, reliability, and very low size and weight parameters. Moreover, the GPNS may serve as a base for:

a) generation of codes and construction of coding devices for the thorough error-control when processing, transmitting, and storing information;

b) development of algorithms and devices used when information is compressed and/or coded;

c) efficient solution of combinatorial optimization problems.

When combinatorial objects are generated and numerated, researchers use to develop special methods for each individual problem, which can be characterized as a principal drawback of such an approach [4,5]. Therefore, a universal algorithm solving these problems at both the theoretical and practical levels would be very helpful. We propose a possible solution method based upon the GPNS. In particular, in this paper, a binomial numeral system is considered, which generates combinatorial objects making use of constant weight codes [6]. The total number of coding words in such codes is determined by binomial coefficients [7].

The generalized positional numeral systems (GPNS) allow one to develop efficient algorithms and specialized digital devices (based upon these algorithms) due to the similarity of their structures. Thus the device cost is saved, and the high computational speed is attained (due to the hardware implementation, up to ten times higher compared to the universal computers). Moreover, as the GPNS are noise-proof, their digital devices use to be much more reliable and easy in trouble-shooting.

It is worthwhile to develop digital devices using a GPNS and completing mainly logical and the simplest arithmetical operations with integer numbers, because these operations are realized by the GPNS in the most efficient way.

Certain parts of such specialized devices, e.g. noise-proof counters, registers, etc., are of interest for the universal computers as well [8,9].

To cope with the problem of noise-proof storage and transmission of information, a lot of various codes, both error-detecting and error-correcting, have been developed. Among those codes, it is worthy to distinguish the codes that detect errors not only during the information transmission and storage but also while it is processed. This class also includes the codes based upon the GPNS, whose strong sides are: (i) the simplicity of algorithms and devices for detecting errors, (ii) the structural regularity, (iii) the possibility to regulate the code's redundancy and hence its error-detecting capability depending upon the channel's adaptability. Such codes are quite applicable in specialized automatic controlling systems, as the information's downloading, processing, transmission, and development of controlling actions are all based upon the same GPNS code.

One of the important problems arising while storing and transmitting information is its compression, like for example by the optimal coding based on the Shannon-Fano and Huffman codes [10,11]. Nowadays, the coding theory can boast with a quite wide arsenal of other ways to compress information, which however cannot exclude the development of new methods and/or improvement of the existing ones. One of those is the numeration of messages, which has the following advantages: (a) an algorithmic coding structure allowing an easy implementation, and (b) no need in a dictionary.

Application of a GPNS permits one to expand the class of numerated messages and thus improve and simplify the algorithmic and technical realization of the information compression.

Both the numeration and de-numeration processes based upon the GPNS can be efficiently used to code the information. Thus we can obtain noise proof codes of high stability and with simple keys used for the information security.

Finally, problems of combinatorial optimization are of special importance. In the most general form, these problems may even not have an objective function but stated in some preference terms. Such problems are usually solved by an exhaustive search, or when it is impossible, by random search procedures [2]. In both cases, the GPNS can provide many efficient ways of generating the combinatorial objects in order to find a path to an optimal solution.

Therefore, the generalized positional numeral systems (GPNS) propose a unified approach allowing one to solve efficiently a series of practical problems of various natures. As an example of such a GPNS, our paper presents a binary binomial numeral system. The latter is characterized with the use of binomial coefficients as weights of the binary digits [12-14].

The rest of the paper is arranged as follows. In Section 2, we define the principal structure of the binomial calculus system. Section 3 presents the mains results establishing the key properties of the binomial systems, namely, the prefix and the compactness properties. Finally, Section 4 deals with the algorithms to generate and numerate binomial combinations of various lengths (non-uniform codes), the constant length (uniform codes), and the constant weight combinations. Conclusion, acknowledgement and the reference list complete the paper.

2 Binomial Systems

Now we describe one of the GPNS, namely the binomial system with the binomial weights and the binary alphabet {0,1} [12-14].

In a *k*-binomial system with n registers (k < n), the quantitative equivalent *QAi* of a code combination *Ai* = $(a_{j-1}, a_{j-2}, \dots, a_0)$, $i = 0,1,\dots, P-1$, with $P = C_n^k$, where $j = j(i)$ is the combination's length, is defined as follows:

$$QA_i = a_{j-1}C_{n-1}^{k-q_i} + \dots + a_\ell C_{n-j+\ell}^{k-q_\ell+1} + \dots + a_0 C_{n-j}^{k-q_1}, \quad (1)$$

where the following conditions must hold: either

$$\begin{cases} q_0 = k, \\ j < n; \\ a_0 = 1, \end{cases} \tag{2}$$

or

$$\begin{cases} n - k = j - q_0, \\ q_0 < k; \\ a_0 = 0, \end{cases} \tag{3}$$

Here q_0 is the quantity of units (ones) in the binomial number, P is the range of the system, j is the quantity of registers (positions) in the binomial number (or, its length), $\ell = 0,1,\ldots, j-1$ is the register's ordinal number, q_ℓ is the sum of the digits occupying the registers (j-1) through ℓ, inclusively, i.e.

$$q_\ell = \sum_{i=\ell}^{j-1} a_i , \tag{3}$$

with $q_j = 0$.

A positional numeral system must be finite, effective, and well-defined.
However, it is not enough for a generalized positional system. In addition, it has to be a prefix code system, i.e. with the "prefix property": there is no valid code word in the system that is a prefix (start) of any other valid code word in the set. With a prefix code, a receiver can identify each word without requiring a special marker between words. The generalized positional numeral system should be also continuous, which means that for any number s from the system's range (except for the maximal number), there exists a combination, whose quantitative equivalent is equal to (s + 1). All these properties of the binomial numeral systems will be established in the next section.

3 The Binomial System is Finite, Effective, Prefix, and Well-Defined

Formula (1) shows that the binomial numeral system is *finite* and *effective*, because there exists a numeration algorithm, which, after a finite of number of steps, converts the coding combination A_i into its quantitative equivalent QA_i. Now the following theorem establishes the *prefix property* of the binomial numeral system. Although its proof was first given in [12], we repeat it here to make the paper self-contained.

Theorem 3.1. [12]. *The k-binomial numeral system with n registers (where k < n) is a prefix code system.*

Proof. Let a coding combination A_i satisfy conditions (2), i.e., let it contain exactly k units (ones). As the condition $j < n$ in constraint (2) implies that the length of such combinations cannot exceed (n-1), then we conclude that the length j of A_i lies between k and $n - 1$, that is, $k \le j \le n - 1$.

Therefore, the number of zeros z in A_i can be equal to $z = 0,1,\ldots, n\text{-}k\text{-}1$, while the combinations length equals $j = k+z$. The total number of distinct combinations of the same length containing k ones and z zeros coincides with the number of combinations of z (zeros) among the total quantity of $(j - 1)$ elements, namely, $P = C_{k+z-1}^{z}$. The distinct combinations cannot evidently be prefixes of the others having the same length, hence in this case, the desired property holds. As for the combinations of different lengths, their values of z are also different. However, as the combinations in question always have 1 at the extreme right position (i.e., $a_0 = 1$), and the total number of 1's is equal to the same number k, it is clear that the longer combination, in its prefix part of the length equal to the total length of the shorter combination, contains at least one 1 (unit) less than the shorter combination. Therefore, the prefix property is valid for all coding combinations satisfying conditions (2).

Now consider the coding combinations satisfying (3). As the constraint n-k=j-q_0 clearly implies that the total number of zeros in these combinations is constant and equal to n-k, then the combination's generation process stops when the last, the (n-k) -th zero appears at the right end position (i.e., $a_0 = 0$). The number of zeros (n-k) summed with the number $q_0 = 0,1,\dots,k-1$ of 1's defines the combination's total length as $j=n-k+q_0$. Therefore, the number of distinct combinations with q_0 units and (n-k) zeros (including the zero in the right end position) coincides with the number of all possible combinations of q_0 elements among the total of (j-1) positions, that is, $P = C^{q_o}_{n-k+q_o-1}$. Again, the prefix property for the combinations of the same length is evident. As for the combinations of the considered subclass having different length values, they also have different numbers of 0's. Consider two such coding combinations of length values equal to p and q, respectively, with, say, $p < q$. The shorter combination with the length p, which could be a prefix of the longer one, contains exactly (n-k) zeros, the same as the longer combination has. However, the right end position of the longer combination is occupied by zero, hence the number of zeros in the longer combination's prefix of length p cannot exceed (n-k-1), which clearly excludes the possibility for the shorter coding combination to be the prefix of the longer one.

Finally, it is straightforward that no coding combination satisfying (2) can be the prefix of a combination satisfying (3), and vice versa. This is due to the fact that the maximum number of 1's in any combination of the latter class is strictly less than that in every combination of the former class. Therefore, no combination of class (2) can be a prefix of a combination of class (3). In an analogous manner, it is easy to see that the maximum possible number of zeros in an arbitrary coding combination of subset (2) is strictly less than that in any combination of class (3), hence, no combination of subclass (3) can form a prefix of a combination from (2). Therefore, the prefix property is evidently valid for the whole set of combinations satisfying (2) or (3), which completes the proof of the theorem.

To show that the binomial system is well-defined, that is, two distinct coding combinations cannot be equivalent to the same numerical value, we prove the following result (again, see [12]).

Theorem 3.2. [12] *The k-binomial system with n registers (where k < n) is well-defined.*

Proof. The previous result (Theorem 3.1 with the prefix property) implies that any two distinct coding combinations have different digits (0 and 1) at least in one of the registers (counted from left to right). The digits in the registers (if any) preceding the first such register are common for both combinations, whereas the remaining (succeeding) part is called the proper part of each combination in this pair. If we prove that the proper parts of these two coding combinations cannot represent the same number, the binomial system is well-defined. Consider the proper parts of two coding combinations (without affecting generality, assume that the combinations have no coinciding preceding parts):

$$Aw = (a_\alpha ,\dots, a_0) \text{ and } As = (b_\beta ,\dots, b_0);$$

where

$$a_\alpha = 0;\ b_\beta = 1;\quad 0 \le \alpha ,\ \le n - 1;\ 0\ \le w;\ s\ \le P - 1; \text{ and } w \ne s.$$

It is not difficult to demonstrate (see the decription of the algorithm generating non-uniform binomial numbers in Section 4) that if in the coding combination A_w all the digits to the right from a_ were 1's (i.e., $a_m = 1$ for $m = 0, 1, \dots, a -1$), whereas in A_s, vice versa, all the entries to the right from b_β were zero, that is, $b_t = 0$ for $t = 0, 1;\dots , \beta-1$, then the distance between the numbers QA_w and QA_s represented by the combinations A_w and A_s, respectively, would be the minimum possible one. Now we establish that this minimum distance is not zero. Indeed, by definition (1) and by the above assumptions, one has:

$$QA_\omega = 0\cdot C^{k-q_{\alpha+1}}_{n-j_\alpha+\alpha} + 1\cdot C^{k-q_{\alpha+1}}_{n-j_\alpha+\alpha-1} + 1\cdot C^{k-q_{\alpha+1}-1}_{n-j_\alpha+\alpha-2} + \dots + 1\cdots C^{k-q_{\alpha+1}-\alpha+1}_{n-j_\alpha+\alpha-\alpha},$$

and

$$QA_s = 1\cdot C^{k-q_{\beta+1}}_{n-j_\beta+\beta} + 0\cdot C^{k-q_{\beta+1}-1}_{n-j_\beta+\beta-1} + 0\cdot C^{k-q_{\beta+1}-1}_{n-j_\beta+\beta-2} + \dots + 0\cdots C^{k-q_{\beta+1}-1}_{n-j_\beta+\beta-\beta}.$$

Now since

$$QA_{\varpi} = \sum_{i=1}^{\alpha} C_{n-j_{\alpha}+i-1}^{k-q_{\alpha+1}-\alpha+i} = C_{n-j_{\alpha}+\alpha}^{k-q_{\alpha+1}} - 1$$

and hence

$$QA_{s} = C_{n-j_{\beta}+\beta}^{k-q_{\beta+1}} = C_{n-j_{\alpha}+\alpha}^{k-q_{\alpha+1}} = QA_{\varpi} + 1,$$

the latter relationships make it possible to conclude that $QA_{\varpi} \neq QA_{s}$ and thus the minimum distance between them is 1, which completes the proof.

Theorems 3.1 and 3.2 have the following important corollary, which proves the compactness of the binomial numeral systems.

Corollary 3.1. The k-binomial system with n registers ($k < n$) *is compact, that is, its range is complete and covers all the integers between 0* and ($C_n^k - 1$).

Proof. According to formula (1), the maximal number represented in the *k*-binomial system with *n* registers is as follows:

$$QA_{P-1} = 1 \cdot C_{n-1}^{k-q_j} + 1 \cdot C_{n-2}^{k-q_j-1} + \cdots + 1 \cdot C_{n-j}^{k-q_1} = C_n^k - 1.$$

The minimal represented value is zero, hence the total number of the integers between the lower and upper bounds of the range is C_n^k. Meanwhile, it is not difficult to establish that the total number of coding combinations constructed by formula (1) and ending with 1 (i.e. satisfying (2)) is equal to

$$N_1 = \sum_{i=0}^{n-k-1} C_{n-2-i}^{n-k-1-i} = C_{n-1}^{n-k-1} = C_{n-1}^{k}.$$

Similarly, it can be proved that the total number of combinations generated by (1) and ending with 0, i.e. with condition (3), is:

$$N_0 = \sum_{i=0}^{k-1} C_{n-2-i}^{k-1-i} = C_{n-1}^{k-1}.$$

Therefore, the total number of distinct coding combinations in the *k*-binomial system equals

$$N = N_1 + N_0 = C_{n-1}^{k} + C_{n-1}^{k-1} = C_n^k.$$

By Theorem 3.2, the correspondence between the coding combinations and the represented integers is one-to-one, and the compactness of the *k*-binomial system with *n* registers is proved.

Remark 3.1. *It is straightforward that for the k-binomial calculus system with n registers, the range parameter P is equal to* C_n^k.

4 Algorithms Generating Binomial Combinations

Table 4.1 contains the binomial combinations and their quantitative equivalents for the *k*-binomial system with *n* registers, where $n = 6$ and $k = 4$.

They are generated by the following algorithm:

Step 1. An initial combination A_0 consisting of (*n-k*) zeros is composed and referred to as a *key-word*.

Step 2. The digit 1 is put into the right end register, and zero is added to the right side of it.

Step 3. Step 2 is repeated while the number of 1's in the coding word is less than *k-1*. If the number of 1's is equal to *k-1*, then go to Step 4.

Step 4. If the right end position contains zero, we replace it with 1. Go to Step 5.

Step 5. Check the number of 1's in the coding combination: if it equals k but the 1's do not occupy the first *k* registers counted from left to right, go to Step 6. Otherwise, i.e. if the 1's occupy the first *k* registers counted from left to right, then **STOP**: all the combination have been generated.

Table 4.1 Binomial coding combinations of non-constant length *(non-uniform code)*

Binomial word	**Its quantitative equivalent**
00	$0C_5^4 + 0C_4^4 = 0$
010	$0C_5^4 + 1C_4^4 + 0C_3^3 = 1$
0110	$0C_5^4 + 1C_4^4 + 1C_3^3 + 0C_2^2 = 2$
01110	$0C_5^4 + 1C_4^4 + 1C_3^3 + 1C_2^2 + 0C_1^1 = 3$
01111	$0C_5^4 + 1C_4^4 + 1C_3^3 + 1C_2^2 + 1C_1^1 = 4$
100	$1C_5^4 + 1C_4^3 + 0C_3^3 = 5$
1010	$1C_5^4 + 0C_4^3 + 1C_3^3 + 0C_2^2 = 6$
10110	$1C_5^4 + 0C_4^3 + 1C_3^3 + 1C_2^2 + 0C_1^1 = 7$
10111	$1C_5^4 + 0C_4^3 + 1C_3^3 + 1C_2^2 + 1C_1^1 = 8$
1100	$1C_5^4 + 1C_4^3 + 0C_3^2 + 0C_2^2 = 9$
11010	$1C_5^4 + 1C_4^3 + 0C_3^2 + 1C_2^2 + 0C_1^1 = 10$
11011	$1C_5^4 + 1C_4^3 + 0C_3^2 + 1C_2^2 + 1C_1^1 = 11$
11100	$1C_5^4 + 1C_4^3 + 1C_3^2 + 0C_2^1 + 0C_1^1 = 12$
11101	$1C_5^4 + 1C_4^3 + 1C_3^2 + 0C_2^1 + 1C_1^1 = 13$
1111	$1C_5^4 + 1C_4^3 + 1C_3^2 + 1C_2^1 = 14$

Step 6. Update the keyword A_0 by putting 1 as a prefix before the beginning of the keyword (i.e., its left end). If the total number of 1's in the keyword is less than *k*, go to Step 2.

The binomial systems find various important applications, in which the following useful features are exploited: (*i*) the binomial systems are noise-proof in the information transmission, processing, and storage; (*ii*) they are able to search, generate and numerate coding combinations with a constant weight; (*iii*) they can be used to construct noise-proof digital devices. To detect errors with the aid of binomial coding combinations, they should be completed with zeros to obtain uniform *(n - 1)*-digital binomial coding words given in Table 4.2.

Table 4.2 Binomial coding combinations of a constant length (*uniform code*)

NN	**Binomial word**	**Binomial uniform word**
0	00	00000
1	010	01000
2	0110	01100
3	01110	01110
4	01111	01111
5	100	10000
6	1010	10100
7	10110	10110
8	10111	10111
9	1100	11000
10	11010	11010
11	11011	11011
12	11100	11100
13	11101	11101
14	1111	11110

The main tokens of errors in a binomial coding combination are either the number of 1's being greater than *k*, or the number of zeros exceeding (*n*-*k*). The principal feature of the binomial noise-proof code is its ability to detect errors while processing information. This feature allows one to arrange the throughout control in the information processing channels involving the digital devices.

4.1. *Generation of binomial coding combinations with a constant weight*

Next, Table 4.3 shows a transformation of binomial coding combinations to coding words with a constant weight: this is done by adding (to the right end) either 1's if the binomial combination contains (*n*-*k*) zeros, or adding zeros if the combination comprises *k* digits 1, until the combination's length reaches *n*.

Table 4.3 Binomial coding combinations of a constant weight

NN	**Binomial word**	**Binomial constant weight word**
0	00	001111
1	010	010111
2	0110	011011
3	01110	011101
4	01111	011110
5	100	100111
6	1010	101011
7	10110	101101
8	10111	101110
9	1100	110011
10	11010	110101
11	11011	110110
12	11100	111001
13	11101	111010
14	1111	111100

Each binomial combination (column 2 of Table 4.3) has the corresponding combination with the constant weight (column 3 of Table 4.3), hence the former is a compressed image of the latter. If one needs to label a combination with the constant weight by some traditional numeral system number (e.g., decimals of column 1 in Table 4.3), formula (1) has to be used. In the latter case, a compression of binomial numbers is completed.

Algorithms of search and generation of binomial combinations and those with constant weights can be also found in [14]. Now we describe one of modifications of such algorithms and prove its efficiency as follows. This method is based upon the fact that the range of binomial numbers of length n and with parameter *k* ($k < n$) coincides with the range of the constant weight coding combinations with k units among n registers. Therefore, the formal description of the algorithm is as follows:

Step 1. Select an arbitrary non-uniform binomial coding combination.

Step 2. If the coding combination ends with the digit 1, then put zeros into all registers up to the right end (register n), which is considered as auxiliary. The thus obtained combination ending with 0 will be the combination with the constant weight.

Step 3. If the coding combination ends with the digit 0, then set units (ones) into all registers up to the right end (register n, or the auxiliary register). The thus created combination ending with 1 will be the combination with the constant weight.

Step 4. Verify that the thus obtained combination is indeed with the constant weight by counting the total number of ones (units). If this number is k then the combination is indeed a desired one. Select another non-uniform binomial coding combination and go to Step 2. If all the non-uniform binomial coding combinations have been already selected, then **STOP**: all the constant weight combinations of this range have been generated.

The above algorithm generates the complete range of the corresponding combinations of the constant weight, which is confirmed by the following theorem.

Theorem 4.1. *With the aid of the above algorithm, for every non-uniform binomial combination of length n with parameter k (k < n), one obtains the unique corresponding coding combination with (the constant) weight k and length n.*

Proof. First, consider the case when the selected non-uniform binomial coding combination ends with the unit (i.e., with digit 1). According to the definition of the non-uniform binomial coding words, it implies that this combination has already had *k* units (digits 1). Making use of the above-described algorithm (Step 2), we need only to add several zeros into the registers to the right from the rightest 1 till the auxiliary register is filled, thus having obtained the combination with (the constant) weight *k*. It is clear that two different non-uniform binomial coding words cannot generate (with the aid of the above algorithm) the same constant weight combination: indeed, if otherwise, it would imply that one (the shorter) of these non-uniform binomial coding words is the prefix of the second (the longer) one, which would contradict Theorem 3.1.

Next, if the selected non-unifom binomial coding combination ends with 0, then, due to the description of the verified algorithm (see Step 3), we will insert 1's into all the registers to the right from the rightest zero, including the auxiliary register. According to the definition of the non-uniform binomial combination ending with zero, the total number of zeros in it is equal to (n-k); therefore, the constructed new combination will contain $n-(n-k)=k$ digits 1, i.e. it will have the (constant) weight *k*. Repeating exactly the proof for the first case (Step 2) given above, we conclude that different non-uniform binomial combinations ending with 0 will produce different combinations of (the constant) weight *k*. Finally, two constant weight combinations produced by different steps (Step 2 and Step 3) of the above algorithms cannot coincide due to the different digits in their auxiliary registers (0 for Step 2 and 1 for Step 3). The proof is complete.

5 Conclusion

In this paper, we have described the error-detecting binomial numeral systems capable of transmitting, processing and storing information. The systems can also generate and numerate combinatorial configurations, like, for example, coding words with a constant weight, as well as compositions, combinations with repetitions, etc. Moreover, the binomial systems can be applied to produce efficient information compression and defense. The latter is the goal of our further research.

6 Acknowledgements

The research activity of the second author was financially supported by the R&D Department (Cátedra de Investigación) CAT-174 of the Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, and by the SEP-CONACYT projects CB-2008-01-106664 and I0010-2009-01-122315, Mexico. Also, the work of the second and third authors was supported by the Russian Humanitarian Research Foundation (RGNF) within the project RGNF 08-02-00271, while the third author was supported by the PAICYT project No. CE250-09 and by the SEP-CONACYT project I0010-2009-01-118057.

References

[1] Stakhov A.P. Golden Section Codes / A.P. Stakhov. – Moscow: Radio and Communication, 1984. (In Russian).
[2] Stoyan Yu.G. Solution of Some Extremum Problems by the Method of Contracting Neighborhoods, Scientific Thinking / Yu.G. Stoyan, V.Z. Sokolovsky. – Kiev: Naukova Dumka, 1980. (In Russian).
[3] Borisenko A.A. Numerical coding based upon combinatorial calculus systems/ A.A. Borisenko // Theoretical and Applied Aspects of Program Systems Development (TAAPSD'2010): Proceedings of the International Conference, Kiev, Ukraine, October 04-08, 2010 / ed. M.S. Nikitchenko. – pp.98 – 104. (In Russian).
[4] Amel'kin V.A. Numerical Coding Methods / V.A. Amel'kin. – Novosibirsk: Nauka, 1986. (In Russian).
[5] Reingold E.M. Combinatorial Algorithms: Theory and Practice / E.M. Reingold, J. Nievergelt, N. Deo. – Prentice Hall [New Jersey]: Englewood Cliffs, 1977.
[6] Borisenko A.A. Binomial Coding / A.A. Borisenko, I.A. Kulik . – Sumy: Sumy State University, 2010. (In Russian).

[7] Anderson J.A. Discrete Mathematics with Combinatorics / J.A. Anderson. – Prentice Hall [New Jersey]: Upper Saddle River, 2001.
[8] Borisenko A.A. Binomial Calculus and Counters / A.A. Borisenko. – Sumy: Sumy State University, 2008. (In Russian).
[9] Oberman R.M.M. Counting and Counters / R.M.M. Oberman. – Hoboken [New Jersey]: Wiley & Sons, 1981.
[10] Tsymbal V.P. Information and Coding Theory / V.P. Tsymbal. – Kiev: Vyshaya Shkola, 1977. (In Russian).
[11] Kuz'min I.V. A Basic Theory of Information and Coding / I.V. Kuz'min, V.A. Kedrus. – Kiev: Vyshaya Shkola, 1977. (In Russian).
[12] Borisenko A.A. Binomial calculus: Advantages and prospects / A.A. Borisenko, V.V. Kalashnikov, N.I. Kalashnykova // ICIC Express Letters. – 2006. – №2. – pp.123-130.
[13] Borisenko A.A. Introduction to the Theory of Binomial Calculus / A.A. Borisenko. – Sumy: Universitetskaya Kniga, 2004. (In Russian).
[14] Borisenko A.A. Binomial Calculus: Theory and Applications / A.A. Borisenko. – Sumy: Universitetskaya Kniga, 2004. (In Russian).

Рецензент: Олександр Кузнецов, д.т.н., проф., Харківський національний університет імені В. Н. Каразіна, Харків, Україна. E-mail: kuznetsov@karazin.ua

Надійшло: квітень 2016.

Автори:
Олексій Борисенко, д.т.н., проф., Сумський державний університет, Суми, Україна. E-mail: electron@sumdu.edu.ua
В'ячеслав Калашников, д.ф.-м.н., проф., департамент систем і промислового виробництва Технологічного університету Монтеррея, Монтеррей, Мексика. E-mail: kalash@itesm.mx
Наталя Калашникова, к.т.н., доц., автономный университет Нуэво-Леон, Сан Николас де Лос Гарса, Монтеррей, Мексика. E-mail: nkalash@einstein.fcfm.uanl.mx

Опис та застосування біноміальних систем числення.
Анотація. Розробляється новий вид позиційних систем числення, що отримали назву біноміальних, які утворюють підклас узагальнених позиційних систем числення (УПСЧ). Вони мають широку область застосування при передачі, обробці та зберіганні інформації завдяки забезпеченню можливості виявлення помилок і генерування різних комбінаторних конфігурацій. Наведені алгоритми формування біноміальних кодових слів (рівномірних і нерівномірних) та побудови на цій основі рівноважних кодових комбінацій з постійною вагою. Показано коректність цієї процедури.

Ключові слова: позиційні системи числення, кодування, комбінаторика, біноміальні числа.

Рецензент: Александр Кузнецов, д.т.н., проф., Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина. E-mail: kuznetsov@karazin.ua

Поступила: апрель 2016.

Авторы:
Алексей Борисенко, д.т.н., проф., Сумской государственный университет. Сумы, Украина. E-mail: electron@sumdu.edu.ua
Вячеслав Калашников, д.ф.-м.н., проф., департамент систем и промышленного производства Технологического университета Монтеррея, Монтеррей, Мексика. E-mail: kalash@itesm.mx
Наталья Калашникова, доц., к.т.н., автономный университет Нуэво-Леон, Сан Николас де Лос Гарса, Монтеррей, Мексика. E-mail: nkalash@einstein.fcfm.uanl.mx

Описание и применение биномиальных систем счисления.
Аннотация. Разрабатывается новый вид позиционных систем счисления, называемых биномиальными, который образует подкласс обобщенных позиционных систем счисления (ОПС). Они имеют широкую область применения при передаче, обработке и хранении информации благодаря обеспечению возможности обнаружения ошибок и генерирования различных комбинаторных комбинаций. Представлены алгоритмы формирования биномиальных кодовых слов (равномерных и неравномерных) и построения на их основе равновесных кодовых комбинаций с постоянным весом. Показана корректность этой процедуры.

Ключевые слова: позиционные системы счисления, кодирование, комбинаторика, биномиальные числа.