UDC 519. 72

A.V. ANISIMOV

GENERATING (2,3)-CODES

Abstract. The (2,3)-representation of integers utilizes the mixed numeration base of the radix-2 and the auxillary radix-3. This representation yields a universal prefix-free binary encoding of all natural numbers with a variety of useful properties: robustness (self-synchronization), local error corrections, statistic regularities of code parameters, etc. The paper describes a procedure of monotonic generation of (2,3)-codewords in ascending order of their lengths.

Keywords: numeration system, radix, integer encoding, prefix encoding.

A form of integer representation plays an important role in computer architectures and algorithms. Universal integer representations are also of great importance in data transmission and compression especially when statistical properties of data are not known. There is a rich variety of well-known integer representations based on different numeration systems and properties: classic binary and byte, ternary, Fibonacci, Elias–Levenshtein [1, 2], two-base numerations [3]. For the first time the usefulness of integer representations for solving computational and data transmission problems have been demonstrated in [4, 5].

The (2,3)-numeration system has been introduced and thoroughly studied in [6]. It uses the representation of positive integers that are coprime with 2 and 3 in the form

$$x = 2^{n_1} + 3^{k_1} x_2, \tag{1}$$

where x_2 is also coprime with 2 and 3, n_1 is a maximum possible power of 2. The recursive use of representation (1) yields a binary representation of integers with many interesting properties including universality (in the sense of Elias [5]), existence of delimiters, local error corrections [7], block structured representations similar to classic Elias Δ, γ and ω representations [1] but with much smaller coefficients than that of Elias. It is worth to note that (2,3)-codes are well suited for correcting all types of errors investigated in [8]. The set of (2,3)-codewords arises as a set of words in the binary alphabet $\{0, 1\}$ obtained from integer representation (1) through the so-called delta approach. For encoding data by (2,3)-codes, for instance for data compression or transmission, it is necessary to use an independent generating procedure of codewords. The main purpose of the presented communication is to describe a monotonic procedure for generating (2,3)-codes in ascending order of their lengths. The generating procedure utilizes the structural property of a (2,3)-codeword as follows. If a codeword u of the length L has the structural form $u = 0^{\Delta}1^{k}v$, where v starts with 0, then v also is a shorter codeword for some integer. Therefore if a procedure monotonically generates all codewords v of lengths less than L then, for obtaining codewords of the length L, one should prepend a corresponding well-defined prefix $0^{\Delta}1^{k}$ to each v. The rules for correctness of such a prefix are given.

(2,3)-INTEGER REPRESENTATION AND CODING

Let $\mathbb{N}$ be the set of natural numbers extended by zero, $\mathbb{N}_{2,3}$ denotes the set of natural numbers that are coprime with 2 and 3. First, the (2,3)-representation is

defined on the set $\mathbb{N}_{2,3}$. Let $x \in \mathbb{N}_{2,3}$, $x \neq 1$. The number x can be uniquely represented in form (1), where $x_2 \in \mathbb{N}_{2,3}$, n_1 is a maximum power of 2 such that $x \equiv 2^{n_1} \bmod 3$, $x > 2^{n_1}$. It is easy to see that for any n integers 2^n and 2^{n-1} have different residues modulo 3. Thus, in (1) n_1 is equal to $\lfloor \log_2 x \rfloor$ or $\lfloor \log_2 x \rfloor - 1$. In the first case we call representation (1) maximal, and in the second case — minimal.

Applying splitting (1) recursively,we get the sequence of the remaining numbers:

$$x_1 = x, \;\; x_i = 2^{n_i} + 3^{k_i} x_{i+1}, \;\; i = 1, 2, \ldots, t, \;\; x_{t+1} = 1, \tag{2}$$

n_i is the maximum power of 2 such that $x_i \equiv 2^{n_i} \bmod 3$ and $x_i - 2^{n_i} > 0$, x_{i+1} is odd and not divisible by 3.

Examples.

$$5 = 2^1 + 3^1, \;\; 7 = 2^2 + 3, \;\; 11 = 2^3 + 3, \;\; 23 = 2^3 + 3(2^1 + 3^1);$$
$$20152015 = 2^{24} + 3(2^{19} + 3(2^{16} + 3(2^{14} + 3(2^{13} + 3(2^7 + 3(2^6 + 3(2^2 + 3^2)))))))).$$

Consider representation (2). With a number x from $\mathbb{N}_{2,3}$ the numerical sequence of integer pairs is associated

$$(n_1, k_1) \ldots (n_t, k_t). \tag{3}$$

We call (3) the (2,3)-representation of a number x. The pair (n_i, k_i) is called a block. In the sequel, $\bar{n}_{i+1}$ denotes $\lfloor \log_2 (x_{i+1}) \rfloor$.

Local properties between pairwise blocks (3) are defined by inequalities as follows: if the i-th block is maximal then the inequality

$$0 < n_i - \bar{n}_{i+1} - k_i \log_2 3 \tag{4}$$

holds. If the i-th block is minimal then the inequalities

$$-\log_2 3 < n_i - \bar{n}_{i+1} - k_i \log_2 3 < 1 \tag{5}$$

hold. For details see [6].

Using sequence (3), we can construct a Δ-representation of x

$$(\Delta_1, k_1)(\Delta_2, k_2) \ldots (\Delta_t, k_t), \tag{6}$$

where $\Delta_i = n_i - \bar{n}_{i+1} - k_i$. Considering (4) and (5) it is not difficult to prove that for all Δ_i the inequality $\Delta_i \geq 0$ holds [6]. The next step is constructing a binary representation of x using (6). We want to represent pairs (Δ, k) as blocks of the form $0^{\Delta} 1^k$. But in this case some difficulties arise with the case $\Delta = 0$. To overcome the "$\Delta = 0$ problem" we must use some other excessive encoding. The simplest variant is to use $\Delta + 1$ instead of Δ. So, instead of (6) we use

$$(\Delta_1^1, k_1) \ldots (\Delta_t^1, k_t), \tag{7}$$

where $\Delta_i^1 = \Delta_i + 1$. In its turn (6) can be encoded by the concatenation of binary block codes

$$0^{\Delta_1^1} 1^{k_1} 0^{\Delta_2^1} 1^{k_2} \ldots 0^{\Delta_t^1} 1^{k_t} = 0^{n_1 - \bar{n}_2 - k_1 + 1} 1^{k_1} 0^{n_2 - \bar{n}_3 - k_2 + 1} \ldots 0^{n_t - k_t + 1} 1^{k_t}. \tag{8}$$

From local inequalities (4), (5) we can derive that in representation (8) there is no block of the form $0111 \, (\Delta^1 = 1, k = 3)$. Therefore, we can use this string as a delimiter denoted herein as $\# = 0111$.

Table 1. The structural correspondence between $\mathbb{N}$ an $\mathbb{N}_{2,3}$

$\mathbb{N}$	a in binary	$\mathbb{N}_{2,3}$
$6k$	01	$6k+1$
$6k+1$	00	$6k+1$
$6k+2$	11	$6k+5$
$6k+3$	10	$6k+5$
$6k+4$	01	$6k+5$
$6k+5$	00	$6k+5$

Table 2. The first 14 codes

Number	Prefix a	Main body $x+a$	Delimiter
1	00	*blank*	0 111
2	11	01	0 111
3	10	01	0 111
4	01	01	0 111
5	00	01	0 111
6	01	001	0 111
7	00	001	0 111
8	11	0001	0 111
9	10	0001	0 111
10	01	0001	0 111
11	00	0001	0 111
12	01	011	0 111
13	00	011	0 111
14	11	0011	0 111

In order to extend the (2,3)-representation for all natural numbers we use a functional correspondence between $\mathbb{N}$ and $\mathbb{N}_{2,3}$ given by the mapping $x \to x+a$, where a is the smallest positive number such that $x+a \in \mathbb{N}_{2,3}$. It is easy to see that a can only be one of the numbers from the set {0, 1, 2, 3}. The structural correspondence between $\mathbb{N}$ and $\mathbb{N}_{2,3}$ given by the mapping $x \to x+a$ is depicted in Table 1. Thus, there exists a bijective correspondence between sets $\mathbb{N}$ and $\{0,1\} \times \{6k+1\} \cup \{0,1,2,3\} \times \times \{6k+5\}$, $k = 0,1,2,\ldots$.

For the (2,3)-binary representation of an integer x from $\mathbb{N}$, we reserve two prefix bits that characterize a number a in the correspondence $x \to x+a$ and use the (2,3)-representation of a corresponding number $x+a$ from $\mathbb{N}_{2,3}$, $C^+_{2,3}(x) = b_1 b_2 C^+_{2,3}(x+a)$, where b_1 and b_2 are two bits from the binary representation of a.

In [6], representation (8) appended by # is denoted by $C^+_{2,3}(x)$. The code $C^+_{2,3}$ is a set of all words $C^+_{2,3}$, $x \in \mathbb{N}$. $C^+_{2,3}$ codes of the first 14 natural numbers are given in Table 2. For instance, 12 is divisible by 3 and 2. Thus, we must add 1 to it and use the (2,3)-code of 13 as a main body,

$$13 = 2^2 + 3^2,\ \Delta = 0,\ k = 2,\ \Delta^1 = 1,\quad C^+_{2,3}(12) = 01\ 011\ 0111.$$

The code $C^+_{2,3}$ is excessive because we use one extra bit for encoding Δ. Nevertheless, it gives good metric properties. In [6] we proved that the average statistical codeword length of $C^+_{2,3}(x)$ does not exceed the value $1.16 \log_2 x$. Moreover, for some numbers, their (2,3)-representation could be shorter than the traditional binary form. For comparison, it is worth to note that the length of the well-known Fibonacci code Fib2 is always equal to $\approx 1.44 \log_2 x$.

RELATIONSHIPS BETWEEN Δ, k AND $\log_2 x$

Hereinafter, we restrict our considerations only for numbers from $\mathbb{N}_{2,3}$. In this case in a codeword we ignore the constant prefix part 00.

Let x has the representation

$$x = 2^{n_1} + 3^{k_1} x_2 \tag{9}$$

for some n_1. We call this representation (2,3)-canonical if it coincides with (1), i.e., n_1 is a maximum possible power of 2 and $x_2 \in \mathbb{N}_{2,3}$. The next theorem gives the internal characteristics of the (2,3)-canonical representation.

Theorem 1. Representation (9) is (2,3)-canonical if and only if the inequalities

$$1 \le k_1 \le \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil$$

hold. The equality $k_1 = \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil$ holds true if and only if block (9) is minimal.

Proof. Necessity. In (9) $n_1 = \lfloor \log_2 x \rfloor$ or $n_1 = \lfloor \log_2 x \rfloor - 1$. Consider the case $n_1 = \lfloor \log_2 x \rfloor$. This means that (9) corresponds to a maximal block.

It implies inequalities

$$2^{n_1} + 3^{k_1} x_2 < 2^{n_1+1};\ 3^{k_1} x_2 < 2^{n_1};$$

$$k_1 \log_2 3 + \log_2 x_2 < n_1,\ k_1 < \frac{n_1 - \log_2 x_2}{\log_2 3} < \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil.$$

Consider the second case, $n_1 = \lfloor \log_2 x \rfloor - 1$.

This gives inequalities:

$$2^{n_1+1} < 2^{n_1} + 3^{k_1} x_2 < 2^{n_1+2};\ 2^{n_1} < 3^{k_1} x_2 < 3 \cdot 2^{n_1},$$

$$\frac{n_1 - \log_2 x_2}{\log_2 3} < k_1 < \frac{n_1 - \log_2 x_2}{\log_2 3} + 1.$$

Considering that k_1 is a whole number, we get

$$k_1 = \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil.$$

Sufficiency. Assume that for k_1 the inequalities

$$1 \le k_1 \le \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil$$

hold.

Consider the case

$$k_1 \le \left\lfloor \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rfloor.$$

It implies inequalities

$$k_1 \log_2 3 + \log_2 x_2 < n_1, \quad x = 2^{n_1} + 3^{k_1} x_2 < 2^{n_1+1}.$$

Therefore, $n_1 = \lfloor \log_2 x \rfloor$.

Consider the second case

$$k_1 = \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil.$$

This implies inequalities

$$\frac{n_1 - \log_2 x_2}{\log_2 3} < k_1 < \frac{n_1 - \log_2 x_2}{\log_2 3} + 1,$$

$$2^{n_1} < 3^{k_1} x_2 < 3 \cdot 2^{n_1},$$

$$2^{n_1+1} < 2^{n_1} + 3^{k_1} x_2 < 2^{n_1+2}.$$

It follows that $n_1 = \lfloor \log_2 x \rfloor - 1$.

This ends the proof.

Evidently, $n_1 = \Delta_1 + k_1 + \lfloor \log_2 x_2 \rfloor$.

LENGTHS OF (2,3)-CODEWORDS

For further considerations we outline some properties of canonical (2,3)-representations and corresponding lengths of codewords. Note that we consider codes on the restricted set $\mathbb{N}_{2,3}$.

Corollary 1. Let $x = 2^{n_1} + 3^{k_1} x_2$ be the canonical (2,3)-representation.

Then $2^{n_1+1} + 3^{k_1} x_2$ is also the canonical (2,3)-representation.

Proof. Theorem 1 states that the inequalities

$$1 \le k_1 \le \left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil$$

hold. It is obvious that this inequality implies the corresponding inequality for $n_1 + 1$:

$$1 \le k_1 \le \left\lceil \frac{n_1 + 1 - \log_2 x_2}{\log_2 3} \right\rceil.$$

Thus, $2^{n_1+1} + 3^{k_1} x_2$ is also canonical.

Corollary 2. For any integer l, $l \ge 6$, in the code $C^+_{2,3}$ there exists a codeword of the length l.

Proof. If a number x from Corollary 1 has the corresponding codeword of the length l then the number $2^{n_1+1} + 3^{k_1} x_2$ produces the codeword of the length $l+1$. For 5 the codeword is 01 0111. Its length is 6.

Corollary 3. For any number x from $\mathbb{N}_{2,3}$ the codeword $01C^+_{2,3}(x)$ is also the codeword from $C^+_{2,3}$.

Proof. Let $x = 2^{n_1} + 3^{k_1} x_2$ be the canonical (2,3)-representation, $n = \lfloor \log_2 x \rfloor$. Then the word $01C^+_{2,3}(x)$ is the codeword for the number $2^{n+1} + 3x$ with $\Delta = 0$ and $k_1 = 1$.

GENERATING (2,3)-CODES

The obtained correspondence between parameters n_1, k_1 and x_2 in a canonical representation allows us to construct a monotonic generating procedure producing $C^+_{2,3}$-codewords for numbers from $\mathbb{N}_{2,3}$ in increasing order of their lengths.

Let canonical representation (1) for x has the length of its (2, 3)-codeword equal to L, The codeword for x is the concatenation of the code of the first pair (Δ_1^1, k_1) and the codeword of x_2, where

$$\Delta_1^1 = \Delta_1 + 1 \equiv n_1 - \bar{n}_2 - k_1 + 1,$$

$$C_{2,3}^+(x) = 0^{\Delta_1^1} 1^{k_1} C_{2,3}^+(x_2).$$

The length of the code of (Δ_1^1, k_1) is equal to $n_1 - \bar{n}_2 - k_1 + 1 + k_1 = n_1 - \bar{n}_2 + 1 = L - |C_{2,3}^+(x_2)|$, where $|s|$ denotes the length of a string s. It follows that

$$n_1 = L - |C_{2,3}^+(x_2)| + \lfloor \log_2 x_2 \rfloor - 1.$$

The codeword $C_{2,3}^+(x_2)$ has a smaller length than x. Varying all x_2 having smaller lengths of codewords we can get all possible values for n_1. Using Theorem 1 for a fixed n_1, we can choose all possible valid values for k_1. The number 1 is encoded by the delimiter 0111. The next codeword c is 010111 corresponding to the number 5. Its length is 6. There is no codeword of the length 5.

Let A be an array that listed (2,3)-codewords, *Decode* (c) is a decoding procedure which for a given codeword c outputs the corresponding number $x, C_{2,3}^+(x) = c$, MAX $[l] = i$ is a maximum index i in the array A such that $|A[i]| = l$. The smallest value of l is 4, which corresponds to the delimiter 0111.

Thus, A [1] = 0111, *Decode* (0111) = 1, A [2] = 01 0111, *Decode* (010111) = 5, A [3] = 001 0111, *Decode* (0010111) = 7, A [4] = 011 0111, *Decode* (0110111) = 11, MAX [4] = 1, MAX [6] = 2, MAX [7] = 4.

The minimum length of a block is 2. It follows that there is no codeword of the length 5. For convenience, we set MAX [5] = 1. The generation of all codewords of the length L, $L \geq 6$, if we have already filled tables A and MAX with smaller codewords, in a sketch form looks as follows.

Procedure *Generate* (*L*)

1. Input: L;
2. j = MAX $[L - 1] + 1$;
3. for $i = 1$ to MAX $[L - 2]$ by 1 do
   {
   3.1 $x_2 = Decode\, A[i]$;
   3.2 $\bar{n}_2 = \lfloor \log_2 x_2 \rfloor$
   3.3 $n_1 = \bar{n}_2 + L - |A[i]| - 1$
   3.4 for $k = 1$ to $\left\lceil \frac{n_1 - \log_2 x_2}{\log_2 3} \right\rceil$ by 1 do
   {
   3.4.1 $\Delta = n_1 - \bar{n}_2 - k + 1$;
   3.4.2 A[j] = $0^{\Delta} 1^k$ A[i];
   3.4.3 $j = j + 1$
   }
   }
4. MAX[L] = $j - 1$.

Comments. Line 2: j is the first index of a codeword having the length L. Corollary 2 states that it will be the next codeword after the last codeword of the length $L-1$. Line 3: Corollary 3 shows that the minimal first block of x should only be 01.

Generalization of the aforementioned algorithm for the case of all natural numbers consists in the additional prepending to the resulting codeword from $\mathbb{N}_{2,3}$ two starting bits that characterize the value a in the correspondence given by Table 2. To do this it is necessary to decode the codeword A[j] (line 3.4.2) and depending on its form $6k+1$ or $6k+5$ to create two or four consequent codewords of the same length. We omit this easy correction of the procedure $Generate(L)$ extending it to $\mathbb{N}$.

REFERENCES

1. Elias P. Universal codewords sets and representations of integers. *IEEE Transactions on Information Theory.* 1975. Vol. 21, N 2. P. 194–203. https://doi.org/10.1109/TIT.1975.1055349.
2. Levenshtein V.I. On the redundancy and deceleration of separable coding of natural numbers, *Probl. Kibern.* 1968. N 20. P. 173–179.
3. Anisimov A.V. Two-base numeration systems. *Cybernetics and Systems Analysis.* 2013. Vol. 49, N 4. P. 501–510. https://doi.org/10.1007/s10559-013-9535-y.
4. Fraenkel A.S. The use and usefulness of numeration systems. *Information and Computation.* 1989. Vol. 81, N 1. P. 46–61. https://doi.org/10.1016/0890-5401(89)90028-X.
5. Apostolico A., Fraenkel A.S. Robust transmission of unbounded strings using Fibonacci representations. *IEEE Transactions on Information Theory.* 1987. Vol. 33, N 2. P. 238–245. https://doi.org/10.1109/TIT.1987.1057284.
6. Anisimov A.V. Prefix encoding by means of (2,3)-representations of numbers. *IEEE Transactions on Information Theory.* 2013. Vol. 59, N. 4. P. 2359–2374. https://doi.org/10.1109/TIT.2012.2233544.
7. Anisimov A.V., Zavadskyi I.A. Robust prefix encoding using lower (2,3) number representation. *Cybernetics and Systems Analysis.* 2014. Vol. 50, N 2. P. 163–175. https://doi.org/10.1007/s10559-014-9604-x.
8. Butenko S., Pardalos P., Sergienko I., Shylo V., Stetsyuk P. Estimating the size of correcting codes using extremal graph problems. In: *Optimization. Springer Optimization and Its Applications.* Pearce C., Hunt E. (Eds). Vol. 32. New York: Springer, 2009. P. 227–243. https://doi.org/10. 1007/978-0-387-98096-6_12.

Надійшла до редакції 16.06.2020

А.В. Анісімов
ГЕНЕРУВАННЯ (2, 3)-КОДІВ

Анотація. У (2,3)-поданні цілих чисел використано змішану систему числення за базисом 2 та допоміжним базисом 3. Це представлення породжує універсальне безпрефіксне двійкове кодування усіх натуральних чисел з багатьма корисними властивостями: робастність (самосинхронізація), локальні виправлення помилок, статистичні закономірності параметрів коду тощо. Описано процедуру монотонного генерування (2,3)-кодових слів у порядку зростання їхніх довжин.

Ключові слова: система числення, базис, цілочисельне кодування, префіксне кодування.

А.В. Анисимов
ГЕНЕРАЦИЯ (2, 3)-КОДОВ

Аннотация. В (2,3)-представлении целых чисел использована смешанная система счисления по основанию 2 и вспомогательному основанию 3. Это представление порождает универсальное префиксно-свободное двоичное кодирование всех натуральных чисел, которое имеет много полезных свойств: робастность (самосинхронизация), локальные исправления ошибок, статистические закономерности параметров кода и т. п. Описана процедура монотонной генерации (2,3)-кодовых слов в порядке возрастания их длин.

Ключевые слова: система счисления, основание, целочисленное кодирование, префиксное кодирование.

Anisimov Anatolii Vasylyovich,
Dr. of Sciences, professor, the dean of the faculty of the Taras Shevchenko National University of Kyiv,
e-mail: anatoly.v.anisimov@gmail.com.