Adaptive Universal Codes for Integer Representation

Márcio J. de C. Lima and Valdemar C. da Rocha Jr..

Abstract— **For a given arbitrary list of integer numbers, in general there is no known single universal code which is an overall optimum in the sense of representing the whole list with the shortest average codeword length. This is the motivation in this paper to introduce a class of adaptive universal codes based on pattern codes, for integer representation. A construction of adaptive universal codes is given, which is based on Fibonacci codes. This construction is shown to perform well over a wider range of integer values in comparison to known universal code constructions.**

Keywords: source coding, Fibonacci codes, algorithms, design of algorithms.

I. Introduction

The efficient encoding and decoding of an arbitrary list of integer numbers, often in a compact representation [1], [2], represents an interesting problem in areas such as source coding theory [3]-[6], data compression [7]-[10], digital data transmission theory [11], [12], synchronizable codes [11], [13]-[15], theory of computational complexity [22] and data base indexes for fast file access [16].

For a given arbitrary list of integer numbers, in general there is no known single universal code which is an overall optimum in the sense of representing the whole list with the shortest average codeword length [1], [8], [9], [17], where by the length of a codeword we mean the number of digits that it contains. We were then motivated to try an alternative approach and partially circumvent this difficulty by introducing adaptive universal (AU) [18] codes based on *pattern codes*, originally treated as prefix codes by Gilbert [3]. By an adaptive code we mean a code represented by a set of two or more codes together with a code selection rule. The code selection rule specifies which code in the set is to be used, as a function of the range of integers to be represented. Pattern codes have been studied in detail by Lakshmanan [23] and it is worth mentioning that they are related to a conjecture of Gilbert [3], studied later by Guibas and Odlyzko [19], concerning the maximum number of codewords allowed in a class of block codes known as *prefix-synchronized* codes.

In the following, we consider a generalization of the universal codes introduced by Capocelli [14] of the kind related to the Zeckendorf [20] representation of integers in terms of generalized Fibonacci numbers, that have the capability of locally confining errors. According to Capocelli [14], for a given positive integer r, universal codes can be constructed by considering the set of all binary strings of length greater than or equal to r in which the sequence formed by a 0 followed by $r-1$ 1's, denoted as 01_{r-1}, occurs only once as a suffix. These binary strings form a countably infinite set of prefix-free codewords which we will denote by $\mathcal{S} = S(r, 01_{r-1})$. It was also shown that the $\mathcal{S}$ codes are prefix-free and complete [21], and universal in the sense of Elias [22],[23]. Furthermore, $\mathcal{S}$ codes can be recognized by a finite-state automaton (i.e., they are regular codes) and thus cannot be asymptotically optimal [24]. Moreover, $\mathcal{S}$ codes are synchronizable and have synchronization delay equal to one, i.e., in case errors occur and synchronization is lost, after a single codeword the decoder recovers synchronization (more robust than most other variable-length codes) [1],[12].

In the following, we introduce a construction for a class of codes which enjoy properties similar to the $\mathcal{S}$ codes. However, the codes in our construction perform well over a wider range of integer numbers. These codes are denoted by $\mathcal{A}$ and result from a combination of at least two universal codes. In Section II, we present a basic introduction to classical universal codes and show that the generalized Fibonacci representation can be exploited to construct binary *uniquely decodable* codes. In Section III we define Adaptive Universal codes. In Section IV, we construct a class of codes, called Fibonacci Adaptive Universal (FAU) codes, and compare their performance with that of other universal schemes. In Section V, we close this paper by presenting some comments and conclusions.

II. Classical Universal Codes

After examining various universal codes, [5]-[7], [11]-[14], [22]-[25] the codes having the shortest average codeword length were obtained with the Fibonacci codes known as *pattern codes* [3]. This fact is illustrated in Table I, where L_Ω denotes the set of codeword lengths for code Ω and N denotes a positive integer. The three rightmost columns in Table I contain results from some of the proposed codes, to be described later. A *prefix-free* code is any code with the property that no codeword is a prefix of any other codeword. Let $\mathfrak{p} = p_1p_2\ldots p_m$ denote an arbitrary binary string, or pattern, of length m. We denote by $l(\mathfrak{p})$ the length of the pattern $\mathfrak{p}$, i.e., the length of the pattern $\mathfrak{p} = p_1p_2\ldots p_m$ is the positive integer m.

Definition 2.1: For a given binary pattern $\mathfrak{p}$ of length m, a *pattern code* ($\mathfrak{p}$-code) is a set T of binary strings of variable length $l+m$, $l \geq 0$, such that for any $x_1x_2\ldots x_lp_1p_2\ldots p_m \in T$, the pattern $\mathfrak{p} = p_1p_2\ldots p_m$ occurs only once as a suffix. For example, the pattern $\mathfrak{p} = 0111$ of length four is a suffix in the codeword 1100001000 0111. It is clear that every $\mathfrak{p}$-code is a prefix-free code, and is therefore uniquely decodable.

Definition 2.2: For given integers N_1 and N_2, $N_1 \leq N_2$, let A and B denote codes for representing an integer i, $N_1 \leq i \leq N_2$. Code A is defined as uniformly better than code B in the

Márcio J. de C. Lima and Valdemar C. da Rocha Jr., Communications Research Group, Department of Electronics & Systems - UFPE, CP7800, CEP 50711-970 Recife, PE, BRAZIL. Email: marcio.clima@ufpe.br, vcr@ufpe.br.

TABLE I

COMPARISON OF CODEWORD LENGTHS FOR ELIAS CODES [22] γ, δ, ω, WANG CODES [5] $W(2)$, $W(3)$ AND $W(4)$, YAMAMOTO CODES [6] $Y(2)$, $Y(3)$ AND $Y(4)$, FIBONACCI CODES BY APOSTOLICO AND FRAENKEL [11] $AF_1^{(2)}$ AND $AF_1^{(3)}$, FIBONACCI CODES BY CAPOCELLI [14] $C(3)$ AND $C(4)$, AND PROPOSED ADAPTIVE FIBONACCI CODES $\mathcal{A}(3,4)$, $\mathcal{A}(3,5)$ AND $\mathcal{A}(4,5)$.

$\lfloor \log_2(N) \rfloor$	L_γ	L_δ	L_ω	$L_{W(2)}$	$L_{W(3)}$	$L_{W(4)}$	$L_{Y(2)}$	$L_{Y(3)}$	$L_{Y(4)}$	$L_{AF_1^{(2)}}$	$L_{AF_1^{(3)}}$	$L_{C(3)}$	$L_{C(4)}$	$L_{\mathcal{A}(3,4)}$	$L_{\mathcal{A}(3,5)}$	$L_{\mathcal{A}(4,5)}$
0	1	1	1	3,0	4,0	5,0	2,0	3,0	4,0	2	3	3	4	3	3	4
1	3	4	3	4,5	5,0	6,0	3,5	4,0	5,0	3	4	4	5	4	4	5
2	5	5	6	6,0	6,2	7,0	5,0	5,2	6,0	4	5	5	6	5	5	6
3	7	8	7	7,5	7,3	8,1	6,5	6,5	7,1	6	6	6	7	6	6	7
4	9	9	11	9,0	8,5	9,1	8,0	7,7	8,2	7	8	7	8	7	8	8
5	11	10	12	10,5	9,7	10,2	9,5	9,0	9,3	8	9	8	9	8	9	9
6	13	11	13	12,0	10,8	11,3	11,0	10,2	10,5	10	10	9	10	10	10	10
7	15	14	14	13,5	12,0	12,3	12,5	11,5	11,6	11	11	10	11	11	11	11
8	17	15	16	15,0	13,2	13,4	14,0	12,7	12,7	13	12	12	12	12	12	12
9	19	16	17	16,5	14,3	14,5	15,5	14,0	13,8	14	13	13	13	13	14	13
10	21	17	18	18,0	15,5	15,6	17,5	15,2	15,0	16	14	15	14	14	15	14
12	25	19	20	21,0	17,8	17,7	20,0	17,7	17,2	18	17	17	16	17	17	17
14	29	21	22	24,0	20,2	19,8	23,0	20,2	19,5	21	19	20	19	19	19	19
16	33	25	28	27,0	22,5	22,0	26,0	22,7	21,7	24	21	23	21	22	22	21
18	37	27	30	30,0	24,8	24,1	29,0	25,2	24,0	27	23	26	23	24	24	23
20	41	29	32	33,0	27,2	26,3	32,0	27,7	26,2	30	25	29	25	26	26	25
22	45	31	34	36,0	29,5	28,4	35,0	30,2	28,5	33	28	32	28	29	28	28
24	49	33	36	39,0	31,8	30,6	38,0	32,7	30,7	36	30	35	30	31	30	30
26	53	35	38	42,0	34,2	32,7	41,0	35,2	33,0	39	33	38	32	33	33	32
28	57	37	40	45,0	36,5	34,8	44,0	37,7	35,2			40	35			34
30	61	39	42	48,0	38,8	37,0	47,0	40,2	37,5			43	37			

interval (N_1, N_2) when the codeword $A(i), A(i) \in A, N_1 \leq i \leq N_2$, has either the same length or a shorter length than $B(i), B(i) \in B$, for any value of i, $N_1 \leq i \leq N_2$.

It is well known [1], [8], [14] that pattern codes known as Fibonacci codes are so far the best for representing a list of integers. However, when considering a sufficiently large range of integers, the integer representation provided by a single Fibonacci code is not uniformly better than that given by another Fibonacci code (see Table I). This is also the case for other families of universal codes.

Among other properties, we notice that Fibonacci codes have an easy-to-understand structure, a simple indication of codeword length, and as indicated in Table I they have a compact representation for integer numbers over large intervals. By $C(m)$ we denote a Fibonacci code employing a pattern of length m. For example, in Table II, $C(3)$, $C(4)$ and $C(5)$ denote Fibonacci codes with patterns $\mathfrak{p} = 011$, $\mathfrak{p} = 0111$ and $\mathfrak{p} = 01111$, of length 3, 4 and 5, respectively.

A. Fibonacci codes

Fibonacci codes are binary codes constructed based on the Fibonacci enumeration system [13], [20], [24]. The Fibonacci *enumeration system of order* $m, m \geq 2$, [11], denoted as $\mathcal{F}^{(m)}$, can be used to represent any non-negative integer number, employing the following:

a) a binary alphabet, e.g., $\{0, 1\}$, and
b) the fact that there is a unique manner of writing a given integer which does not contain m consecutive digits equal to 1 [13],[26].

The representation $\mathcal{F}^{(2)}$ is known as the Zeckendorf's representation [20]. The generalized sequence of Fibonacci numbers of order r, $r \geq 2$, is defined by the linear recursion

$$F_s^{(r)} = F_{s-1}^{(r)} + F_{s-2}^{(r)} + \cdots + F_{s-r}^{(r)}, \;\; s \geq 1,$$

where

$$F_{-r+1}^{(r)} = F_{-r+2}^{(r)} = \cdots = F_{-2}^{(r)} = F_{-1}^{(r)} = 0$$

and $F_0^{(r)} = 1$.

Example 2.1: The enumeration system $\mathcal{F}^{(2)}$ uses the Fibonacci sequence of order two, consisting of the numbers $1, 2, 3, 5, 8, 13, 21, 34, \ldots$, as a basis. The $\mathcal{F}^{(2)}$ representation for the numbers 11 and 17 is 00101 and 101001, respectively, as follows from

$$\begin{aligned} \mathcal{F}^{(2)}(11) &= 0 \cdot 1 + 0 \cdot 2 + 1 \cdot 3 + 0 \cdot 5 + 1 \cdot 8; \\ \mathcal{F}^{(2)}(17) &= 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 3 + 0 \cdot 5 + 0 \cdot 8 + 1 \cdot 13. \end{aligned}$$

Examples of Fibonacci codes are presented in Table II, specifically, codes $C(3), C(4)$ and $C(5)$, which belong to the family of Fibonacci codes presented by Capocelli et al [14],[24],[25], following the common practice in the literature of showing only some initial part of the integers. In general, for a given integer m, $m > 2$, $C(m)$ denotes a pattern code, as introduced by Capocelli, consisting of the set of all binary strings of length greater than or equal to m in terms of generalized Fibonacci numbers.

III. ADAPTIVE UNIVERSAL CODES

Our proposal for constructing Adaptive Universal (AU) codes consists of combining two or more pattern codes so that the resulting adaptive code benefits from good properties of the component codes and pays a small penalty in efficiency. We call the reader's attention to the fact that, as explained in the following, each codeword representing a positive integer in an adaptive universal code begins with an identification pattern of 0's and 1's of fixed length, which uniquely specifies the suffix of the Fibonacci code being used to construct that codeword.

Given an arbitrary list of positive integers, for coding purposes this list can be subdivided into nonoverlapping intervals. The purpose of an adaptive code is to represent all intervals

TABLE II
FIRST 16 CODEWORDS OF CODES $C(3)$, $C(4)$ AND $C(5)$.

N	$C(3)$	$C(4)$	$C(5)$
1	011	0111	01111
2	0 011	0 0111	0 01111
3	1 011	1 0111	1 01111
4	00 011	00 0111	00 01111
5	01 011	01 0111	01 01111
6	10 011	10 0111	10 01111
7	11 011	11 0111	11 01111
8	000 011	000 0111	000 01111
9	001 011	001 0111	001 01111
10	010 011	010 0111	010 01111
11	100 011	011 0111	011 01111
12	101 011	100 0111	100 01111
13	110 011	101 0111	101 01111
14	111 011	110 0111	110 01111
15	0000 011	111 0111	111 01111
16	0001 011	0000 0111	0000 01111

with the same accuracy of the best pattern code for each interval. Let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$ denote s pattern codes with disjoint codebooks, i.e., such that $\mathcal{C}_i \cap \mathcal{C}_j = \{\phi\}$, $i \neq j, 1 \leq i \leq s, 1 \leq j \leq s$.

Definition 3.1: An adaptive code $\mathcal{A}(\mathcal{C}_1, \ldots, \mathcal{C}_s)$, is a multi-pattern code with a codebook consisting of subsets of codewords of s disjoint pattern codes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$, i.e., $\mathcal{A}(\mathcal{C}_1, \ldots, \mathcal{C}_s) \subset \mathcal{C}_1 \cup \mathcal{C}_2 \ldots \cup \mathcal{C}_s$.

A specific codebook construction for $\mathcal{A}(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s)$ is described in subsection III-A.

A. Codebook construction rule for $\mathcal{A}(\mathcal{C}_1, \ldots, \mathcal{C}_s)$

Without loss of generality, let $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s\}$ denote a set of s disjoint pattern codes, for which $l(\mathfrak{p}_1)$ is the shortest pattern length. The idea behind the rule for constructing the codebook for $\mathcal{A}(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s)$, $s \geq 2$, is to select codewords as short as possible. This rule is described as follows:

- The first codeword is represented by the binary sequence $\mathfrak{p}_1 = p_1 p_2 \ldots p_{m_1}$ (the shortest pattern), $\mathfrak{p}_1 \in \mathcal{C}_1$.
- Except for the first codeword, other codewords are produced as follows:
  1) Let i = 1.
  2) Use the binary sequence $q_1 q_2 \ldots q_a$, of length $a = \lceil \log s \rceil$, that specifies in $\mathcal{C}_i$ those codewords beginning with $q_1 q_2 \ldots q_a$.
  3) Append to $q_1 q_2 \ldots q_a$ an unused shortest binary sequence $x_1 x_2 \ldots x_l$, $l \geq 0$, where $l = 0$ means a void sequence.
  4) Append to $q_1 q_2 \ldots q_a x_1 x_2 \ldots x_l$ the binary sequence representing the pattern $\mathfrak{p}_i$, thus forming a codeword of length $T = a + l + m_i$ from $\mathcal{C}_i$.
  5) Go to step 2 and continue in this manner, selecting codewords from code $\mathcal{C}_i$, as long as there are codewords of length T in $\mathcal{C}_i$. Otherwise, go to step 6.
  6) Make $i + 1 \rightarrow i$. If $i \leq s$, go to step 2. Otherwise, go to step 7.
  7) Make $l + 1 \rightarrow l$.
  8) Go to step 1.

Summarizing, the codebook construction rule for $\mathcal{A}(\mathcal{C}_1, \ldots, \mathcal{C}_s)$ produces codewords X that are binary sequences, such that one codeword has the form $p_1 p_2 \ldots p_{m_i}$ and the remaining codewords have the general form $q_1 q_2 \ldots q_a x_1 x_2 \ldots x_l p_1 p_2 \ldots p_{m_i}$.

Example 3.1: Let $s = 2$ and let $\mathcal{A}(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{A}(C(3), C(4))$, where $C(3)$ and $C(4)$ denote Fibonacci codes, as defined by Capocelli [14], with patterns $\mathfrak{p}_1 = 011$ and $\mathfrak{p}_2 = 0111$, respectively. Since $s = 2$, it follows that $a = \lceil \log_2 2 \rceil = 1$, and the first digit in each codeword of $\mathcal{A}(C(3), C(4))$ identifies whether it belongs to $C(3)$ or $C(4)$. Specifically, codewords in $\mathcal{A}(C(3), C(4))$ that start with $q_1 = 0$ come from $C(3)$ and codewords that start with $q_1 = 1$ come from $C(4)$. The first 16 codewords of codes $C(3), C(4)$ and $\mathcal{A}(C(3), C(4))$ are shown in Table III, where the codewords in $\mathcal{A}(C(3), C(4))$ that came from code $C(4)$ are written in boldface type. By the codebook construction rule, the first codeword in $\mathcal{A}(C(3), C(4))$ is the pattern 011, and the second, third and fourth codewords are $0011, 00011, 01011$, all from $C(3)$, since up to length 4 there is no codeword beginning with 1 in $C(4)$ with a length equal to or shorter than 4. The next codeword beginning with 0 in $C(3)$ is 000011, of length 6, and we notice that $C(4)$ offers 10111, which begins with 1 and has length 5, and is thus selected for $\mathcal{A}(C(3), C(4))$, and so on.

Example 3.1 serves the purpose of illustrating our construction but is by no means restricted to the Fibonacci codes introduced by Capocelli [14]. In fact, it is important to point out that our construction of universal adaptive codes allows the freedom to choose any binary number to represent a pattern.

In the next section, we present a specific algorithm to construct an adaptive universal code by selecting codewords from two Fibonacci codes, without having to construct the codebook for the Fibonacci component codes.

IV. CONSTRUCTING FIBONACCI ADAPTIVE UNIVERSAL CODES

Although Definition 3.1 indicates how to combine two or more pattern codes to produce an adaptive code, in general this is not practical because it would imply constructing first the pattern codes, containing possibly a long list of codewords each, and then combing them. For that reason we present now a procedure that produces an adaptive code without the need for the intermediate step of constructing the component pattern codes. As illustrated in Table I, the integer representation provided by a single Fibonacci code is not uniformly better than that given by another Fibonacci code over a sufficiently large range of integers. This is also the case for other families of universal codes, when used for integer representation. Along the same reasoning adopted in Section III we propose the construction of Fibonacci Adaptive Universal (FAU) codes by combining two Fibonacci codes so that the resulting adaptive code benefits from good properties of the component codes. The possibility of combining more than two codes still needs further investigation before a practical algorithm is devised.

A. Construction

For given positive integers u and v, $u < v$, let $C(u)$ and $C(v)$ denote Fibonacci codes [14]. We now show how to

TABLE III
The first 16 codewords of pattern codes $C(3), C(4)$ and adaptive code $\mathcal{A}(C(3), C(4))$.

N	$C(3)$	$C(4)$	$\mathcal{A}(3,4)$
1	011	**0111**	011
2	0 011	**0 0111**	0 011
3	1 011	**1 0111**	00 011
4	00 011	**00 0111**	01 011
5	01 011	**01 0111**	**1 0111**
6	10 011	**10 0111**	000 011
7	11 011	**11 0111**	001 011
8	000 011	**000 0111**	010 011
9	001 011	**001 0111**	**10 0111**
10	010 011	**010 0111**	**11 0111**
11	100 011	**011 0111**	0000 011
12	101 011	**100 0111**	0001 011
13	110 011	**101 0111**	0010 011
14	111 011	**110 0111**	0100 011
15	0000 011	**111 0111**	0101 011
16	0001 011	**0000 0111**	**100 0111**
17	0010 011	**0001 0111**	**101 0111**
18	0100 011	**0010 0111**	**110 0111**
19	0101 011	**0011 0111**	**111 0111**

TABLE IV
Numbers used for constructing the adaptive code $\mathcal{A}(3,4)$ for $k \leq 8$.

k	0	1	2	3	4	5	6	7	8
$F_k^{(2)}$	1	1	2	3	5	8	13	21	34
$F_k^{(3)}$	1	1	2	4	7	13	24	44	81
$G_k^{(3)}$	1	2	4	8	15	28	52	96	177
$S_k = G_{k-2}^{(3)} + F_k^{(2)}$	1	1	3	5	9	16	28	49	86
$M_k = M_{k-1} + S_k$	1	2	5	10	19	35	63	112	198
I_k	1–1	2–2	3–5	6–10	11–19	20–35	36–63	64–112	113–198

combine $C(u)$ and $C(v)$ to produce a FAU code, denoted by $\mathcal{A}(C(u), C(v))$, or $\mathcal{A}(u, v)$ for simplifying the notation.

We denote as $\mathcal{B}_{\mathcal{A}(u,v)}(N)$, $\mathcal{B}_{\mathcal{A}(u,v)}(N) \in \mathcal{A}(u, v)$, the codeword representing the positive integer N in $\mathcal{A}(u, v)$. Let $G_k^{(u)} = \sum_{i=0}^{k} F_i^{(u)}$ and let $S_k = G_{k-(v-u)-1}^{(u)} + F_k^{(u-1)}$, where $G_k^{(u)} = 0,\ k < 0$. Let $M_k = M_{k-1} + S_k$, for $k \geq 0$, where $M_k = 0$ for $k < 0$. Finally, let I_k denote the interval $I_k = [1 + M_{k-1}, M_k]$. For a given positive integer N, the associated codeword $\mathcal{B}_{\mathcal{A}(u,v)}(N)$ of the adaptive code $\mathcal{A}(u, v)$ is constructed as follows.

1. If $N = 1$, then emit the codeword $\mathcal{B}_{\mathcal{A}(u,v)}(1) = 01_{u-1}$.
2. If $N > 1$, then find k, such that $N \in I_k$.
3. Compute $Q = N - M_{k-1} - 1$.
4. If $Q \leq F_k^{(u-1)} - 1$, then $\mathcal{B}_{\mathcal{A}(u,v)}(N)$ consists of the symbol 0 followed by the binary sequence represented by $\mathcal{F}^{(u-1)}(Q)$, having length $k - (u - 2)$. If necessary, fill $\mathcal{F}^{(u-1)}(Q)$ with zeroes on the left in order to occupy all $k - (u - 2)$ places. Append the suffix for codewords from $C(u)$.
5. If $Q \geq F_k^{(u-1)}$, then $\mathcal{B}_{\mathcal{A}(u,v)}(N)$ consists of the symbol 1 followed by the binary sequence represented by $\mathcal{F}^{(v-1)}(Q - F_k^{(u-1)})$, having length $k - (v - 2)$. If necessary, fill $\mathcal{F}^{(v-1)}(Q - F_k^{(u-1)})$ with zeroes on the left in order to occupy all $k - (v - 2)$ places. Append the suffix for codewords from $C(v)$.

Example 4.1: We describe next how to combine the Fibonacci codes $C(3)$ and $C(4)$ to produce the FAU code $\mathcal{A}(3, 4)$. For a given positive integer N, the associated codeword of the adaptive code $\mathcal{A}(3, 4)$ is constructed as follows.

1. If $N = 1$, then emit the codeword $\mathcal{B}_{\mathcal{A}(3,4)}(1) = 011$.
2. If $N > 1$, then find k, such that $N \in I_k$.
3. Compute $Q = N - M_{k-1} - 1$.
4. If $Q \leq F_k^{(2)} - 1$, then $\mathcal{B}_{\mathcal{A}(3,4)}(N)$ consists of the symbol 0 followed by the binary sequence represented by $\mathcal{F}^{(2)}(Q)$, having length $k - 1$. If necessary, fill $\mathcal{F}^{(2)}(Q)$ with zeroes on the left in order to occupy all $k - 1$ places. Append the suffix 011, of code $C(3)$.
5. If $Q \geq F_k^{(2)}$, then $\mathcal{B}_{\mathcal{A}(3,4)}(N)$ consists of the symbol 1 followed by the binary sequence represented by $\mathcal{F}^{(3)}(Q - F_k^{(2)})$, having length $k - 2$. If necessary, fill $\mathcal{F}^{(3)}(Q - F_k^{(2)})$ with zeroes on the left in order to occupy all $k - 2$ places. Append the suffix 0111, of code $C(4)$.

Example 4.2: Continuing with Example 4.1, we consider now generating the codeword $\mathcal{B}_{\mathcal{A}(3,4)}(15)$ representing the integer $N = 15$ using code $\mathcal{A}(3, 4)$. Please refer to Table IV and Table V. Since $N = 15 > 1$, we can start from step 2.

2. $N = 15 > 1$, and the value of k for which $N = 15 \in I_k$, i.e., for which $10 < N \leq 19$, is $k = 4$;
3. Compute $Q = N - M_{k-1} - 1 = 15 - 10 - 1 = 4$;
4. Since $Q \leq F_4^{(2)} - 1$, emit the symbol 0 and represent Q as $\mathcal{F}^{(2)}(Q) = \mathcal{F}^{(2)}(4)$ having length $k - 1 = 3$, i.e., as $\mathcal{F}^{(2)}(4) = 101$. Finally, the codeword $B_{\mathcal{A}(3,4)}(15) = 0\underline{101}\ 011$ is produced.

Table IV shows the numbers involved in the construction

TABLE V

DEVELOPMENT OF ADAPTIVE CODE $\mathcal{A}(3,4)$.

N	$\mathcal{A}(3,4)$	Q
1	011	–
2	0 011	0
3	0$\underline{0}$ 011	0
4	0$\underline{1}$ 011	1
5	1 0111	0
6	0$\underline{00}$ 011	0
7	0$\underline{01}$ 011	1
8	0$\underline{10}$ 011	2
9	1$\underline{0}$ 0111	0
10	1$\underline{1}$ 0111	1
11	0$\underline{000}$ 011	0
12	0$\underline{001}$ 011	1
13	0$\underline{010}$ 011	2
14	0$\underline{100}$ 011	3
15	0$\underline{101}$ 011	4
16	1$\underline{00}$ 0111	0
17	1$\underline{01}$ 0111	1
18	1$\underline{10}$ 0111	2
19	1$\underline{11}$ 0111	3

of $\mathcal{A}(3,4)$ for $k \leq 8$. Figure 1 illustrates a typical behavior of our construction of FAU, in comparison with the respective component codes. Notice that code $\mathcal{A}(3,4)$ performs better than code $C(4)$ for codeword lengths $t,\ t < 11$, and performs better than code $C(3)$ for $t \geq 11$, where $N_\Omega(t)$ denotes the number of distinct codewords with length t by scheme Ω.

Example 4.3: The numbers required for the construction of codes $\mathcal{A}(3,5)$ and $\mathcal{A}(4,5)$ are presented in Tables VI and VII, respectively, for values $k \leq 8$. Codes $\mathcal{A}(3,5)$ and $\mathcal{A}(4,5)$ are constructed from the combination of Fibonacci codes $C(3)$ and $C(5)$, and codes $C(4)$ and $C(5)$, respectively, using the general construction described in Section IV-A. Figure 2 illustrates the fact that code $\mathcal{A}(3,5)$ performs better than code $C(5)$ for codeword lengths $t,\ t \leq 12$, and performs better than code $C(3)$ for $t \geq 15$.

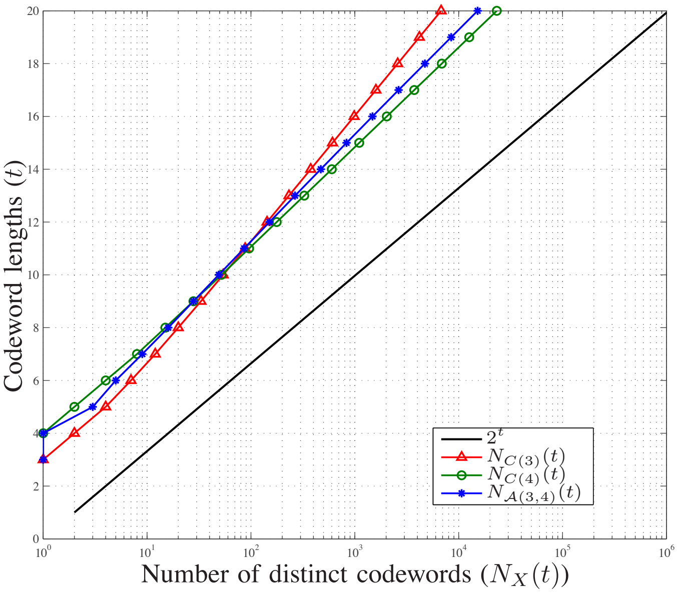


Fig. 1. Comparison of the number of length-t codewords for the binary representation with Fibonacci codes $C(3), C(4)$ and Adaptive code $\mathcal{A}(3,4)$.

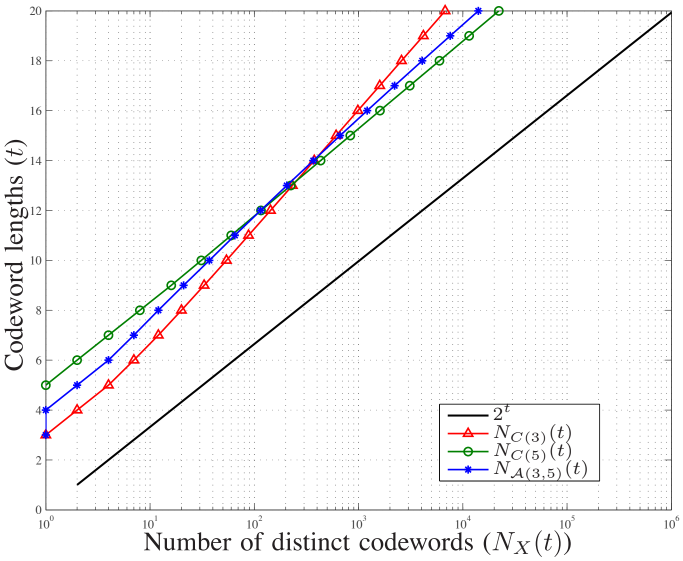


Fig. 2. Comparison of the number of length-t codewords for the binary representation with Fibonacci codes $C(3), C(5)$ and Adaptive code $\mathcal{A}(3,5)$.

V. COMMENTS AND CONCLUSIONS

In this paper we introduced Adaptive Universal codes to represent integers by combining two or more pattern-like codes, in particular, Fibonacci codes. This method represents a new approach to the classic problem of the compact representation of an arbitrary list of integers. As an application of our construction technique, we chose Fibonacci codes as the component codes since they showed the best individual performance among various other codes (see Table I). As it happens so far with all known coding constructions for the integers, our construction is also sub-optimal in the sense that adaptive codes are asymptotically longer than the logarithmic representation [11], [24]. However, the construction of adaptive codes represents an interesting alternative to the problem of coding for arbitrary symbols, represented by integers, frequently in a compact representation, in many practical applications that use universal codes.

ACKNOWLEDGEMENT

This work received partial support from the Brazilian National Council for Scientific and Technological Development - CNPq, Projects 304696/2010-2 and 555046/2008-1. The first author is grateful to his colleague Dr. Bruno Ávila for helpful discussions on the early stages of this research.

REFERENCES

[1] D. Salomon and G. Motta, *Handbook of Data Compression*, Springer, London, 5th Edition, 2010.

[2] N. Abramson, *Information Theory and Coding*, McGraw Hill, New York, 1963.

[3] E. N. Gilbert, "Synchronization of binary messages", *IEEE Trans. Inform. Theory*, vol. 6, no. 4, pp. 470–477, 1960.

[4] J. Rissanen and G. G. Langdon, "Universal modeling and coding", *IEEE Trans. Inform. Theory*, vol. 27, no. 1, pp. 12–23, 1981.

[5] M. Wang, "Almost asymptotically optimal flag encoding of the integers", *IEEE Trans. Inform. Theory*, vol. 34, no. 2, pp. 324–326, 1988.

[6] H. Yamamoto and H. Ochi, "A new asymptotically optimal code for the positive integers", *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1420–1429, 1991.

[7] P. Fenwick, "Variable length integer codes based on the Goldbach conjecture and other additive codes", *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2412–2417, 2002.

TABLE VI

NUMBERS USED FOR CONSTRUCTING THE ADAPTIVE CODE $\mathcal{A}(3,5)$ FOR $k \leq 8$.

k	0	1	2	3	4	5	6	7	8
$F_k^{(2)}$	1	1	2	3	5	8	13	21	34
$F_k^{(4)}$	1	1	2	4	8	15	29	56	108
$G_k^{(4)}$	1	2	4	8	16	31	60	116	224
$S_k = G_{k-3}^{(4)} + F_k^{(2)}$	1	1	2	4	7	12	21	37	65
$M_k = M_{k-1} + S_k$	1	2	4	8	15	27	48	85	150
I_k	1–1	2–2	3–4	5–8	9–15	16–27	28–48	49–85	86–150

TABLE VII

NUMBERS USED FOR CONSTRUCTING THE ADAPTIVE CODE $\mathcal{A}(4,5)$ FOR $k \leq 8$.

k	0	1	2	3	4	5	6	7	8
$F_k^{(3)}$	1	1	2	4	7	13	24	44	81
$F_k^{(4)}$	1	1	2	4	8	15	29	56	108
$G_k^{(4)}$	1	2	4	8	16	31	06	116	224
$S_k = G_{k-2}^{(4)} + F_k^{(3)}$	1	1	3	6	11	21	40	75	141
$M_k = M_{k-1} + S_k$	1	2	5	11	22	43	83	158	299
I_k	1–1	2–2	3–5	6–11	12–22	23–43	44–83	84–158	159–299

[8] P. Fenwick, "Universal Codes", in *Lossless Compression Handbook*, pp. 55–78. Academic Press, 2003.

[9] K. Sayood, *Introduction to Data Compression*, Third Edition, Morgan Kaufmann, December 2005.

[10] S. T. Klein and M. K. Ben-Nissan, "On the usefulness of Fibonacci compression codes", *The Computer Journal*, vol. 53, no. 6, pp. 701–716, 2010.

[11] A. Apostolico and A.S. Fraenkel, "Robust transmission of unbounded strings using Fibonacci representations", *IEEE Trans. Inform. Theory*, vol. 33, no. 2, pp. 238–245, 1987.

[12] A.S. Fraenkel and S.T. Klein, "Robust universal complete codes for transmission and compression", *Discrete Applied Mathematics* vol. 64, no. 1, pp. 31–55, 1996.

[13] W. H. Kautz, "Fibonacci codes for synchronization control", *IEEE Trans. Inform. Theory*, vol. 11, pp. 284 - 292, 1965.

[14] R. M. Capocelli, "Comments and additions to 'Robust transmission of unbounded strings using Fibonacci representations' ", *IEEE Trans. Inform. Theory*, vol. 35, no. 1, pp. 191–193, 1989.

[15] A. J. Wijngaarden and H. Morita, "Partial-prefix synchronizable codes", *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1839–1848, 2001.

[16] H. E. Williams and J. Zobel, "Compressing integers for fast file access", *The Computer Journal* vol. 42, no. 3, pp. 193–201, 1999.

[17] M. J. C. Lima, "Source Encoding and Integer Representation," Master Dissertation, Department of Electronics & Systems - UFPE, Federal University of Pernambuco, Pernambuco, Brazil, 2007. (in Portuguese)

[18] M. J. C. Lima, V. C. da Rocha Jr., B. T. Ávila, "Adaptive Universal Codes for Integer Representation," Proceedings of the *XXVI Brazilian Telecommunications Symposium*, Rio de Janeiro, Brazil, 2008. (in Portuguese)

[19] L. J. Guibas and A. M. Odlyzko, "Maximal Prefix-Synchronized Codes", *SIAM Journal on Applied Mathematics*, vol. 35, no.2, pp. 401–408, 1978.

[20] E. Zeckendorf, "Représentation des nombres naturels par une somme des nombres de Fibonacci ou de nombres de Lucas", Bull. Soc. Roy. Sci. Liège 41, pp. 179–182, 1972.

[21] J. M. Berstel and D. Perrin, *Theory of Codes*, Pure and Applied Mathematics, vol. 117. Academic Press, Inc., New York, 1985.

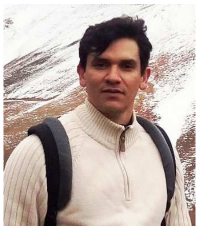
[22] P. Elias, "Universal codeword sets and representations of the integers", *IEEE Trans. Inform. Theory*, vol. 21, no.2, pp. 194–203, 1975.

[23] K. B. Lakshmanan, "On universal codeword sets", *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 659–662, 1981.

[24] R. M. Capocelli and A. De Santis, "Regular universal codeword sets", *IEEE Trans. Inform. Theory*, vol. 32, no. 1, pp. 129–133, 1986.

[25] R. M Capocelli, "Flag encoding related to the Zeckendorf representation of integers", in *Sequences, Combinatorics, Compression, Security, and Transmission*, R. M. Capocelli, Ed. New York: Springer-Verlag, pp. 449-466, 1990.

[26] D. E. Knuth, *The Art of Computer Programming*, Volumes 1-3 Boxed Set. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.



Márcio José de Carvalho Lima was born in Paulo Afonso, BA, Brazil on September 9, 1978. He received the Electrical/Electronic Engineering degree (2004) and the M.Sc. degree (2007) from the Federal University of Pernambuco. Since 2008 he has been doing research for his doctoral thesis on error-correcting codes at the Federal University of Pernambuco. Research interests include turbo codes, multiple access channel and cryptography.

Valdemar C. da Rocha Jr. (M'77, SM'04, Life SM'13) was born in Jaboatão, Pernambuco, Brazil, on August 27, 1947. He received in 1970 the B.Sc. degree in Electrical/Electronics Engineering from the Escola Politécnica, Recife, Brazil, and in 1976 he received the Ph.D. degree in Electronics from the University of Kent at Canterbury, U.K. He joined the faculty of the Federal University of Pernambuco, Recife, Brazil, in 1976 as an Associate Professor and founded its Electrical Engineering Postgraduate Programme. He served as Department Chair (1992-1996), and in 1993 he became Professor of Telecommunications.

He was editor for Coding Theory and Techniques, Journal of Communication and Information Systems, co-sponsored by the Brazilian Telecommunications Society and the IEEE Communications Society, and has been a reviewer for a number of scientific journals including IET Electronics Letters, IET Communications and IEEE Transactions on Information Theory. He has also been involved in the organization of conferences in Brazil and abroad.

He is a founder (2002) and past President (2002-2004) of the IEEE Information Theory Society Chapter, Brazil Council. He is founder (2003) and Vice-President for three consecutive terms (2003-2015) of the Institute for Advanced Studies in Communications. He is a founding member (1983) of the Brazilian Telecommunications Society, served as Vice-President for two terms (2000-2004) and as President also for two terms (2004-2008). He joined the IEEE Communications Society in 1977 and the IEEE Information Theory Society in 1981. He is a Member (1982) of the Brazilian Society of Applied and Computational Mathematics, and a Fellow (1992) of the Institute of Mathematics and its Applications, UK.

He was a Visiting Professor at the Swiss Federal Institute of Technology-Zurich, Institute for Signal and Information Processing (1990-1992), at the University of Leeds, Institute of Integrated Information Systems (2005-2006), and at Lancaster University, Department of Communication Systems (2007).

Prof. da Rocha research interests are in applied digital information theory, including error-correcting codes and cryptography. He has published over 100 engineering and scientific papers, including journal and conference papers, and the book Communication Systems, by Springer, 2005, in co-authorship.