View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Purdue E-Pubs

Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1985

Robust Transmission of Unbounded Strings Using Fibonacci Representations

Alberto Apostolico

Aviezri S. Fraenkel

Report Number:

85-545

Apostolico, Alberto and Fraenkel, Aviezri S., "Robust Transmission of Unbounded Strings Using Fibonacci Representations" (1985). *Department of Computer Science Technical Reports.* Paper 464.
https://docs.lib.purdue.edu/cstech/464

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

ROBUST TRANSMISSION OF UNBOUNDED STRINGS USING FIBONACCI REPRESENTATIONS

Alberto Apostolico

Aviezri S. Fraenkel
The Weizmann Institute of Science

CSD-TR-545
October 1985

Robust Transmission of Unbounded Strings Using Fibonacci Representations

Alberto Apostolico[1] and Aviezri S. Fraenkel[2]

Abstract. Families of Fibonacci codes and Fibonacci representations are defined. Their main attributes are: (i) robustness, manifesting itself by the local containment of errors; (ii) simple encoding and and decoding. The main application explored is the transmission of binary strings whose length is in an unknown range, using robust Fibonacci representations instead of the conventional error-sensitive logarithmic ramp representation. Though the former is asymptotically longer than the latter, the former is actually shorter for very large initial segments of integers.

Key words and phrases: Fibonacci codes, Fibonacci representations, Fibonacci systems of numeration, uniquely decipherable codes, prefix codes, universal representations, asymptotic length of Fibonacci codes.

1. Introduction

Efficient *logarithmic ramp* representations of binary strings of either unbounded length or a priori unknown length, have emerged some time ago in the somewhat related frameworks of data transmission [5,15], coding theory [3] and unbounded searching [1]. Logarithmic ramp representations rest on a simple idea: after writing the string S — encoded in binary, say — the length of S, with leading 1, is similarly encoded and prefixed to S. The process of recursively placing the length of a string in front of that string is repeated until a short string, of length 3, say, is obtained. Since all strings representing lengths begin with a leading 1-bit, the bit 0 can be used to mark the end of the logarithmic ramp and the beginning of S.

[1] Department of Computer Science, Purdue University, West Lafayette, Indiana 47907, USA.

[2] Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. Part of this work was done while visiting at the Department of Mathematics and Statistics, the University of Calgary, Calgary, Alberta, Canada. Supported in part by Canadian Research Grant NSERC 09-0695.

For example, the string $S = 001011100$ is represented as follows:

$$100-1001-0-001011100.$$

The major disadvantage of this representation, however, lies in its vulnerability to errors. If an error occurs in the logarithmic ramp, then the decoding capability is lost and cannot, normally, be recovered.

The main contribution of this paper is to show that generalized Fibonacci systems of numeration [12, 6] can be exploited to construct binary *uniquely decipherable* (UD) *codes* which are robust and easy to encode and decode. They can, in particular, be exploited to represent unbounded strings efficiently.

The key idea lies in the following property of a Fibonacci numeration system of order m ($m \geq 2$), denoted by $\mathcal{F}^{(m)}$ in the sequel: any positive integer N can be expressed uniquely as a sum of distinct m-th order Fibonacci numbers, provided that no m consecutive such numbers are used. In other words, the encoding of N in $\mathcal{F}^{(m)}$ is a binary encoding with the property that it contains no run of m or more consecutive 1-bits. A run of m consecutive 1-bits can thus be used as a *comma*, also called *separator*, separating consecutive codewords.

A *representation* is a bijection of a countable infinite set S_1 of strings onto a set S_2 of strings, such that any concatenation of the members of any subset of S_2 is UD [4, Ch. 4]. The set S_2 is called a *code*, and its members *codewords*. For example, the encoding of the positive integers using the standard binary numeration system $\{1, 10, 11, 100, 101, \ldots\}$ is not a representation: the parses 1,1 and 11 of the string 11 illustrate the problem. However, any prefix code is UD. (A *prefix code* is any code with the property that no codeword is a prefix of any other codeword.)

Let $P = (a_1, \ldots, a_p)$ be an arbitrary binary string (the *pattern*). A *pattern code* (*P-code*) is a set T of binary strings, each of length $\geq p$, such that for any $x = x_1 x_2 \cdots x_{n+p} \in T$ ($n \geq 0$), P occurs in x precisely once, as a suffix. That is, $x_{n+j} = a_j$ for $j = 1, \ldots, p$, and there is no $i \in [0, n-1]$ such that $x_{i+j} = a_j$ for $j = 1, \ldots, p$. Note that every P-code is a prefix code, and is thus UD.

A P-code is *comma free* or *synchronizable* (*SP-code*), if for any codeword $x = x_1 x_2 \cdots x_n a_1 \cdots a_p \in T$, the pattern P does not appear as a block anywhere in $a_2 \cdots a_p x_1 \cdots x_n a_1 \cdots a_{p-1}$. Thus for $P = 0101$, the string 11010101 is not in any P-code; 0110101 is in some P-code but not in any SP-code; and 10110101 is in some SP-code.

A receiver turned on in the midst of the transmission of an SP-code has only to identify P for unambiguous parsing of the code, which is not true for a general UD code. On the other hand, an SP-code is not in general complete. (A UD code is *complete* if addition of any codeword renders it non-UD.) However if P has *autocorrelation* $PP = 10 \cdots 0$ (see [10]), it is easy to see that the SP-code with pattern P can be completed. It is therefore not too surprising that the number of codewords of an SP-code of fixed length N is maximized — for P of suitable length — when $PP = 10 \cdots 0$. This has been proved by Guibas and Odlyzko [10] for large n and p about $\log_2 n$. It implies a conjecture of Gilbert [9].

A P-code with $P = (01)$ has been considered in [16]. In this code the length of an encoded integer increases as the square root of the integer. For Fibonacci representations it increases only logarithmically. Some initial properties of Fibonacci fixed-length codes have been considered by Kautz [11]. The universality of P-codes was investigated by Lakshmanan [13].

Fibonacci numbers came up in previous work on P-codes as *bounds* for code lengths, etc., but not, it seems, as *codewords* in UD codes. The main new features of this work is the construction of robust codes based on the Fibonacci numeration system which are easy to encode and decode, the exploration of their properties, application to the robust transmission of strings of unknown sizes, and the "asymptotic efficiency" computation of this transmission.

In Section 2 we construct two basic Fibonacci representations, $\varphi_1^{(m)}$ and $\varphi_2^{(m)}$, based on a single P-code $C_1^{(m)}$ derived from Fibonacci numbers of order m ($m \geq 2$). The representation $\varphi_1^{(m)}$ maps arbitrary binary integers onto $C_1^{(m)}$. Here and in the sequel, a *binary integer* is a binary sequence with leading 1. A *leading bit* or *leading string* is the most significant (leftmost) bit or string of a binary string. The representation $\varphi_2^{(m)}$ maps arbitrary binary strings, which may begin with leading 0, onto $C_1^{(m)}$. We also give in Section 2 encoding and decoding algorithms for transforming standard binary encodings to the $\varphi_1^{(m)}$ and $\varphi_2^{(m)}$ representations and vice versa. We finally prove in Section 2, using the Kraft equality [8, Ch. 3], that $C_1^{(m)}$ is *complete.*

In Section 3 we construct an alternate UD code $C_2^{(m)}$, based on Fibonacci numbers of order m, and a natural representation $\varphi_3^{(m)}$ which maps the positive integers onto $C_2^{(m)}$. The main difference between $C_1^{(m)}$ and $C_2^{(m)}$ is that $C_2^{(m)}$ contains binary integers only. It is possible to construct many other Fibonacci codes. Some variants of interest are investigated in [7], where also the robustness of Fibonacci codes is examined in greater detail. Using again the Kraft equality, we show that also $C_2^{(m)}$ is complete, and we compare the densities of $C_1^{(m)}$ and $C_2^{(m)}$.

In the main Section 4 we apply Fibonacci representations to the problem of the robust transmission of binary strings in an unknown range. We show that the logarithmic ramp representation is asymptotically shorter than any Fibonacci representation, but that, nevertheless, integers in a very large initial range have shorter Fibonacci representations, depending on the order m of the underlying Fibonacci numeration system. The "transition point" for $\varphi_1^{(m)}$ is $F_{27}^{(2)} - 1 =$

514, 228 for $m = 2$. For $m = 3$, it is

$$\frac{1}{2}\left(F_{63}^{(3)} + F_{61}^{(3)} - 1\right) = 34,696,689,675,849,696 \simeq 3.470 \times 10^{16},$$

and for $m = 4$,

$$\frac{1}{3}\left(F_{231}^{(4)} + 2F_{229}^{(4)} + F_{228}^{(4)} - 1\right) \simeq 4.194 \times 10^{65}.$$

These computations are based on a list of higher-order Fibonacci numbers which Gerald Bergum has kindly prepared for us.

We point out that every Fibonacci code is a *fixed* infinite set, independent of the probability distribution of any given source. In particular, the code does not have to be constructed anew for every probability distribution, as, for example, a Huffman code. On the other hand, the independence of probability implies that Fibonacci codes, unlike Huffman codes, are not generally optimal. In the final Section 5 we show, however, that a very broad family of Fibonacci representations, including $\varphi_1^{(m)}$, $\varphi_2^{(m)}$ and $\varphi_3^{(m)}$, is *universal* in the sense of Elias [3]. That is, the expected representation lengths lie within a constant multiple of the optimal entropy lower bound.

2. Two Basic Fibonacci Representations

Fibonacci numbers of order $m \geq 2$ are defined by the recurrence

$$F_n^{(m)} = F_{n-1}^{(m)} + F_{n-2}^{(m)} + \cdots + F_{n-m}^{(m)} \qquad (n \geq 1), \tag{1}$$

where $F_{-m+1}^{(m)} = F_{-m+2}^{(m)} = \cdots = F_{-2}^{(m)} = 0$, $F_{-1}^{(m)} = F_0^{(m)} = 1$.

Thus $F_1^{(m)} = 2$ for all $m \geq 2$, $F_2^{(3)} = 4$, $F_3^{(3)} = 7$, $F_4^{(3)} = 13$.

In the sequel we often write F_i for $F_i^{(m)}$ when an arbitrary but fixed m is the underlying order of F_i.

Every nonnegative integer N has precisely one binary encoding of the form

$$N = \sum_{i=0}^{k} d_i F_i \qquad (d_i \in \{0,1\}, 0 \leq i \leq k),$$

such that there is no run of m consecutive Fibonacci numbers of order m in the summation. This is the $\mathcal{F}^{(m)}$-*numeration system* [12, 6]. The encoding of the

Table 1: The $\mathcal{F}^{(3)}$-encoding, sequence of messages M, $C_1^{(3)}$-code and mappings $\varphi_1^{(3)}$ and $\varphi_2^{(3)}$.

Message	Code	$\mathcal{F}^{(3)}$-Encoding	Integer
M	7421 $\quad C_1^{(3)}$	137421	N
		0	0
0	111	1	1
00	0111	10	2
000	00111	11	3
1	10111	100	4
0000	000111	101	5
01	010111	110	6
2	100111	1000	7
3	110111	1001	8
00000	0000111	1010	9
001	0010111	1011	10
02	0100111	1100	11
03	0110111	1101	12
4	1000111	10000	13
5	1010111	10001	14
6	1100111	10010	15
000000	00000111	10011	16

first few nonnegative integers in $\mathcal{F}^{(3)}$ is shown in the two right-hand columns of Table 1.

For any $i \geq 1$, let 1_i denote the string of i consecutive 1-bits, 01_i the string 1_i prefixed by 0, and 1_i0 the string 1_i postfixed by 0. Similarly, 0_i denotes the string of of i consecutive 0-bits.

The code $C_1^{(m)} = C_1$ is a P-code with pattern $P = 1_m$ defined as follows. The first two codes are 1_m and 01_m of length m and $m+1$ respectively. The codes of length $m+n$ $(n \geq 2)$ each consist of the suffix 01_m and a prefix of length $n-1$. These prefixes are the first few $\mathcal{F}^{(m)}$-encodings of the nonnegative integers, in increasing size, which can be encoded by at most $n-1$ bits. Leading 0-bits are prefixed, where necessary, to complete the length to $m+n$. The first 16 codewords of $C_1^{(3)}$ appear in the middle column of Table 1.

The representation $\varphi_1^{(m)} = \varphi_1$ maps the set of positive integers Z^+ bijectively onto C_1, such that if $N_1 < N_2$, then $\varphi_1(N_1)$ is lexicographically smaller than $\varphi_1(N_2)$ (see the three right-hand columns of Table 1 for $m = 3$).

Below we give some basic properties of C_1 and φ_1. We remark that since each codeword in C_1 ends in 1_m, C_1 is a prefix code. The *length* of a codeword is the number of binary bits it comprises. Regarding lengths of codewords in C_1 we have,

LEMMA 1. The code C_1 contains precisely F_{n-1} codewords of length $m+n$, which are partitioned as follows: F_{n-2} with leading 0, F_{n-3} with leading 10, F_{n-4} with leading $110, \ldots, F_{n-m-1}$ with leading $1_{m-1}0$ $(n \geq 0)$.

PROOF. Induction on n. Clear for $n = 0, 1$ and 2. Suppose the result holds for n. The definition of C_1 implies that the codewords of length $m+n+1$ can be produced from those of length $m+n$ by prefixing 0 to all the latter, and by prefixing 1 to all of them except to the F_{n-m-1} codewords with leading $1_{m-1}0$. This gives F_{n-1} codewords with leading 0, F_{n-2} with leading 10, F_{n-3} with leading $110, \ldots, F_{n-m}$ with leading $1_{m-1}0$ — a total of F_n codewords of length $m+n+1$. ▮

COROLLARY 1. Let $S_n^{(m)} = S_n = \sum_{i=-1}^{n} F_i^{(m)}$ $(n \geq -1)$ and $S_n^{(m)} = 0$ for $n < -1$. Then all and only all the F_{n-1} integers in the interval $I_{n-1} = [S_{n-2}+1, S_{n-1}]$ have φ_1-representation length $m+n$ $(n \geq 0)$.

PROOF. The integer 1 is in I_{-1} which has representation length m. Next $2 \in I_0$ of length $|\varphi_1(2)| = m+1$. By Lemma 1, the next $F_1 = 2$ integers, those in $[F_{-1}+F_0+1, F_{-1}+F_0+F_1]$ have length $m+2$, that is, $|\varphi_1(3)| = |\varphi_1(4)| = m+2$. The proof is completed by induction on n. ▮

We thus have,

COROLLARY 2. If $|\varphi_1(k)| \leq m+n$, then k is in the interval $[1, S_{n-1}]$ $(n \geq 0)$. ▮

For the encoding and decoding processes, it is useful to compute S_n efficiently.

LEMMA 2. Let $S_n = \sum_{i=-1}^{n} F_i$ $(n \geq -1)$. Then

$$S_n = \frac{1}{m-1}\left(F_{n+2} + \sum_{i=0}^{m-3}(m-2-i)F_{n-i} - 1\right) \qquad (n \geq -1, m \geq 2). \quad (2)$$

PROOF. Induction on n for arbitrary but fixed m. For $n = -1$, the right-hand-side of (2) becomes

$$\frac{1}{m-1}\left(F_1 + (m-2)F_{-1} - 1\right) = 1 = F_{-1} = S_{-1}.$$

If the assertion is true for n, then

$$\begin{aligned}S_{n+1} &= S_n + F_{n+1} \\ &= \tfrac{1}{m-1}\left(F_{n+2} + (m-1)F_{n+1} + \textstyle\sum_{i=0}^{m-3}(m-2-i)F_{n-i} - 1\right) \\ &= \tfrac{1}{m-1}\left(F_{n+3} + \textstyle\sum_{i=0}^{m-3}(m-2-i)F_{n+1-i} - 1\right),\end{aligned}$$

where we used the recurrence (1). ∎

Encoding Algorithm. Given a positive integer N, compute $\varphi_1(N)$.

(i) If $N = 1$, then $\varphi_1(N) = 1_m$. End. If $N = 2$, then $\varphi_1(N) = 01_m$. End.

(ii) Find n such that $S_{n-2} < N \leq S_{n-1}$. Let $Q = N - S_{n-2} - 1$ and encode Q in $\mathcal{F}^{(m)}$. {The approximate value of n can be computed from the asymptotic result $N \sim \lambda u^n$, see Theorem 4, Section 4 below. Then S_{n-2} can be computed using Lemma 2.}

(iii) Adjoin 01_m as suffix to the $\mathcal{F}^{(m)}$-encoding of Q. Adjoin leading 0-bits, if necessary, to make $\varphi_1(N)$ of length $m + n$. End.

Example. Let $m = 3$, $N = 11$. Since $S_2^{(3)} = 8 < N = 11 < S_3^{(3)}$, we have $n = 4$ and $Q = 2$. Hence $\varphi_1^{(3)}(11) = 0100111$.

Decoding Algorithm. Given $\varphi_1(N)$, compute N.

(i) Remove the suffix 1_m.

(ii) If the remaining prefix is empty, then $N = 1$. End. If it is $\{0\}$, then $N = 2$. End.

(iii) Remove the suffix 0.

(iv) The remaining prefix is an $\mathcal{F}^{(m)}$-encoding. Transform it into standard binary encoding, say b {for example, by using a stored table of m-th order Fibonacci numbers, or by computing them using (1)}. Then $N = b + S_{n-2} + 1$ {where $n = |\varphi_1(N)| - m$, and S_{n-2} is computed using Lemma 2}. End.

Example. Let $m = 3$, $\varphi_1(N) = 1010111$. Then $n = 4$ and $S_{n-2} = S_2 = 8$. The prefix 101 in $\mathcal{F}^{(3)}$ remaining at the end of step (iii) transforms into itself in standard binary encoding. Thus $N = 8 + 5 + 1 = 14$ in decimal, that is, N is the binary string 1110.

We shall now address ourselves to the problem of representing arbitrary binary strings which are not necessarily binary integers.

Towards this end, let $M = Z^0 \cup OZ^0$, where Z^0 is the set of nonnegative integers, and OZ^0 the set of all nonnegative integers with leading binary zeros. The bijection $\varphi_2^{(m)} = \varphi_2$ maps M onto C_1: The subset Z^0 is mapped onto the subset of integers of C_1, that is, onto the subset of codewords with leading 1-bit, such that if $N_1 < N_2$ with $N_1, N_2 \in Z^0$, then $\varphi_2(N_1)$ is lexicographically less than $\varphi_2(N_2)$. The subset OZ^0 is mapped onto the subset of C_1, with leading 0: Let $R = 0;N \in OZ^0$, where $N \in Z^0$. Then $\varphi_2(R) = 0;\varphi_2(N)$. This mapping for $m = 3$ can be observed in the two left-hand columns of Table 1.

Encoding and decoding are even simpler than for φ_1. The essence is that if $N \in Z^+$, then the $\mathcal{F}^{(m)}$-encoding of N, with 01_m postfixed, gives $\varphi_2(N)$. The process is reversed for decoding.

For later reference we record the following

LEMMA 3. If $k \in Z^+$ and $|\varphi_2(k)| \le m+n+1$, then $k \in [1, F_n - 1]$, that is, precisely the first $F_n - 1$ positive integers have φ_2-representations of lengths up to $m+n+1$ $(n \ge 1)$.

PROOF. The definition of φ_2 implies that for $k \in Z^+$, $\varphi_2(k)$ is the $\mathcal{F}^{(m)}$-encoding $E(k)$ of k, postfixed by 01_m. For $k = F_n - 1$ we clearly have $|E(k)| = n$, and for $k = F_n$, $|E(k)| = n+1$. ∎

Our final result in this section is

THEOREM 1. The code C_1 is complete.

PROOF. Any countable UD code C satisfies the Kraft inequality $\sum_{c \in C} 2^{-|c|} \le 1$ (see e.g. [8, Ch. 3; Ch. 9, Ex. 3.7]). It follows that if $\sum_{c \in C} 2^{-|c|} = 1$ (the Kraft *equality*), then C is complete.

Let $\sigma_1^{(m)} = \sigma_1 = \sum_{c \in C_1} 2^{-|c|}$. By Corollary 1,

$$\sigma_1 = \sum_{n=0}^{\infty} 2^{-m-n} F_{n-1}.$$

Thus by (1),

$$\begin{aligned}
\sigma_1 - (2^{-m} + 2^{-m-1}) &= \textstyle\sum_{n=2}^{\infty} 2^{-m-n}(F_{n-2} + F_{n-3} + \cdots + F_{n-m-1}) \\
&= \textstyle\frac{1}{2}\sum_{n=1}^{\infty} 2^{-m-n} F_{n-1} + \frac{1}{2^2}\sum_{n=0}^{\infty} 2^{-m-n} F_{n-1} \\
&\quad \textstyle+ \frac{1}{2^3}\sum_{n=-1}^{\infty} 2^{-m-n} F_{n-1} + \cdots \\
&\quad \textstyle+ \frac{1}{2^m}\sum_{n=-m+2}^{\infty} 2^{-m-n} F_{n-1} \\
&= \textstyle\frac{1}{2}(\sigma_1 - 2^{-m}) + \frac{1}{2^2}\sigma_1 + \frac{1}{2^3}\sigma_1 + \cdots + \frac{1}{2^m}\sigma_1.
\end{aligned}$$

Thus, $\sigma_1 - 2^{-m} = \frac{\sigma_1}{2}\left(1 + \frac{1}{2} + \cdots + \frac{1}{2^{m-1}}\right) = \sigma_1\left(1 - \frac{1}{2^m}\right)$, so $\sigma_1 = 1$. ∎

Notes. (i) The P-code C_1 is not an SP-code, since $1_m \in C_1$. Deleting 1_m makes it into an SP-code with $P = 01_m$, but then it is not complete. It can be completed, but then it will contain runs of length $\ge m$ of 1-bits, and the decoding may be harder than for C_1. The Lakshmanan codes [13] are all incomplete, since they do not contain P.

(ii) If b_n denotes the number of codes of length $m+n$ in a P-code with $|P| = m$, then for C_1 Lemma 2 says that $b_n = F_{n-1}$ $(n \geq 0)$. Lakshmanan [13] showed that $b_n \leq F_n$ $(n \geq 1)$. The bound $b_n \leq F_n$ cannot, however, be assumed for all $n \geq 1$, since, as in the proof of Theorem 1, it can be seen easily that $\sum_{n=1}^{\infty} 2^{-m-n} F_n = 2 - 3 \cdot 2^{-m} > 1$.

(iii) It is not hard to construct P-codes for which the bound F_n is assumed for small n. But for larger n the inequality $b_n \leq F_n$ is then strict. The decoding of such codes may be harder than for C_1.

3. An Alternate Fibonacci Code and Representation

The code we define now is conceptually simpler than C_1. For $m \geq 2$, $C_2^{(m)} = C_2$ consists of the codeword 1_{m-1} and the $\mathcal{F}^{(m)}$-encodings of the positive integers in increasing order, the latter postfixed by 01_{m-1}.

The representation $\varphi_3^{(m)} = \varphi_3$ maps the positive integers bijectively onto C_2 such that if $N_1, N_2 \in Z^+$ with $N_1 < N_2$, then $\varphi_3(N_1)$ is lexicographically smaller than $\varphi_3(N_2)$. The first few integers represented by $\varphi_3^{(2)}$ and $\varphi_3^{(3)}$ are shown in Table 2.

We note that C_2 is not a prefix code: Table 2 shows that $\varphi_3^{(3)}(1)$ is a prefix of, say, $\varphi_3^{(3)}(4)$, and $\varphi_3^{(3)}(2)$ is a prefix of, say, $\varphi_3^{(3)}(11)$. Of course $\varphi_3^{(2)}(1)$ is a prefix of $\varphi_3^{(2)}(N)$ for every $N > 1$. However, C_2 is a UD code, because the concatenation of any two codewords generates the separator 01_m between them. When parsing a concatenation of C_2-codewords, the comma is placed just in front of the last 1-bit of any 1_m encountered.

The general length-distribution of codewords in C_2 is given by

LEMMA 4. For C_2 there are $F_{n-1} - F_{n-2}$ codewords of length $m+n-1$ which can be partitioned as follows: F_{n-3} with leading 10, F_{n-4} with leading 110, $\ldots, F_{n-m-1}$ with leading $1_{m-1}0$ $(n \geq 0)$.

PROOF. Note that C_2 is the same as C_1 except for two changes: (i) all codewords with leading 0 are omitted; (ii) the common suffix is 01_{m-1} instead of 01_m. Now apply Lemma 1. ∎

COROLLARY 3. The code C_2 contains precisely F_n codewords of length not exceeding $m+n$ $(n \geq 0)$.

PROOF. There is one codeword of length $m-1$, $F_1 - F_0$ of length $m+1$, $F_2 - F_1$ of length $m+2, \ldots, F_n - F_{n-1}$ of length $m+n$ $(n \geq 0)$. Adding gives the

Table 2: The codes $C_2^{(2)}$, $C_2^{(3)}$ and the mappings $\varphi_3^{(2)}$, $\varphi_3^{(3)}$.

13 7 4 2 1 $C_2^{(3)}$	13 8 5 3 2 1 $C_2^{(2)}$	N
11	1	1
1011	101	2
10011	1001	3
11011	10001	4
100011	10101	5
101011	100001	6
110011	100101	7
1000011	101001	8
1001011	1000001	9
1010011	1000101	10
1011011	1001001	11
1100011	1010001	12
1101011	1010101	13
10000011	10000001	14
10001011	10000101	15
10010011	10001001	16

result. ∎

Encoding and decoding for φ_3 is very simple: the encoding of 1 is 1_{m-1}. For $N \in Z^+$, $N > 1$, the $\mathcal{F}^{(m)}$-encoding of $N-1$ with 01_{m-1} postfixed is $\varphi_2(N)$. The procedure is reversed for decoding.

We now prove,

THEOREM 2. The code C_2 is complete.

PROOF. It suffices to show that the Kraft equality $\sum_{c \in C_2} 2^{-|c|} = 1$ holds.

Let $\sigma_2^{(m)} = \sigma_2 = \sum_{c \in C_2} 2^{-|c|}$. By Lemma 4,

$$\sigma_2 = \sum_{n=0}^{\infty} (F_{n-1} - F_{n-2}) 2^{-m-n+1}.$$

Let $\sigma_3 = \sum_{n=0}^{\infty} F_{n-1}2^{-m-n+1}$, $\sigma_4 = \sum_{n=0}^{\infty} F_{n-2}2^{-m-n+1}$. Then

$$\sigma_4 = \frac{1}{2}\sum_{n=0}^{\infty} F_{n-1}2^{-m-n+1} = \frac{1}{2}\sigma_3.$$

Thus

$$\sigma_2 = \sigma_3 - \frac{1}{2}\sigma_3 = \frac{1}{2}\sigma_3 = \sum_{n=0}^{\infty} F_{n-1}2^{-m-n} = 1,$$

as we established in the proof of Theorem 1. ∎

Theorems 1 and 2 assert that no codeword can be adjoined to either C_1 or C_2 without losing their UD property. This does not imply that they necessarily have the same density, however. In fact, C_2 contains one code of length $m-1$, whereas the minimum length of the C_1-codewords is m. Since both C_1 and C_2 satisfy the Kraft equality, the density of C_1 must be larger than that of C_2 for some codelengths. We shall see that this is in fact the case everywhere except for small codelengths. This may at first seem counterintnitive, since the separator 01_{m-1} of C_2 is shorter than the separator 01_m of C_1. Note, however, that if we rotate the leading 1-bit of every codeword of C_2 to its right-hand end, the resulting code — which is a prefix code! — contains words with leading 0-bits and every word ends in 01_m. It is not, however, identical to C_1: In the latter there are codewords with leading 1_{m-1}, which do not exist in the former.

Corollary 2 implies that precisely the first S_{n-1} codewords of C_1 have lengths $\leq m+n$ $(n \geq 0)$. By Corollary 3, the first F_n codewords of C_2 have lengths $\leq m+n$ $(n \geq 0)$. Thus the quantity

$$D_n^{(m)} = D_n = S_{n-1} - F_n \qquad (n \geq 0)$$

measures the density difference between the two codes C_1 and C_2.

THEOREM 3. The density difference between C_1 and C_2 is $D_{-1} = -1$, $D_n = S_{n-m-1}$ for $n \geq 0$; thus $D_n = 0$ for $0 \leq n < m$, and $D_n > 0$ for $n \geq m$.

PROOF. We have $D_0 = S_{-1} - F_0 = 0$. For $n > 0$, $D_n = S_{n-1} - F_n = \sum_{i=-1}^{n-1} F_i - (F_{n-1} + \cdots + F_{n-m}) = S_{n-m-1}$. ∎

4. Transmission of Binary Strings in an Unknown Range

Transmitting a binary string whose length is unbounded or lies in an unknown range by means of a P-code or a Fibonacci code such as C_2 clearly has the advantage that any error such as a transmission error, will be locally contained, because of the solid separator P which terminates each codeword of the string (except the last, for C_2). This is in contrast to the logarithmic ramp representation, where any error in the logarithmic ramp section can play total havoc with the decoding efforts.

If the transmission is restricted to integers, one of the representations φ_1 or φ_3 can be used. For arbitrary strings which are not necessarily integers, φ_2 is employed.

It is natural to inquire about the asymptotic length of Fibonacci representations. How does it compare with the length of the logarithmic representation? What size should m be? These are the kind of questions we address ourselves to in this section.

We show that $F_n \sim \lambda u^n$ ($\sim$ denotes "asymptotic to"), where $u^{(m)} = u$ is a Pisot-Vijayaraghavan (PV) number, that is, an algebraic integer > 1 all of whose conjugate other than u itself lie in the open unit circle $|z| < 1$ (see e.g. Cassels [2, Ch. 8]) and $\lambda^{(m)} = \lambda$ a positive number. We also give a sharp estimate of u. These facts give us a good handle on estimating the asymptotic length of Fibonacci representations.

LEMMA 5. For all $m \geq 2$, the polynomial

$$f(z) = z^m - (z^{m-1} + z^{m-2} + \cdots + z + 1) \tag{3}$$

has m distinct roots, one of which is a PV-number satisfying

$$2 - 2^{-m+1} < u < 2 - 2^{-m}.$$

PROOF. The first part has been proved by Miles [14]. See also Knuth [12, Sect. 5.4.2, Ex. 5]. For proving the second part, note that

$$f(z) = \frac{z^{m+1} - 2z^m + 1}{z - 1} = \frac{z^m(z-2) + 1}{z - 1}.$$

Let $p(z) = 1 - (2 - z)z^m$. Then

$$p(2 - 1/2^m) = 1 - 2^{-m}(2 - 2^{-m})^m = 1 - (1 - 2^{-(m+1)})^m > 0.$$

On the other hand, using the binomial expansion, we get

$$\begin{aligned} p(2 - 1/2^{m-1}) &= 1 - 2^{-(m-1)}(2 - 2^{-(m-1)})^m = 1 - 2(1 - 2^{-m})^m \\ &< 1 - 2(1 - m2^{-m}) = -1 + m2^{-(m-1)} \leq 0 \end{aligned}$$

for $m \geq 0$. ∎

We remark that with a little more effort, the interval for u can be narrowed further.

THEOREM 4. If F_n is the n-th term of the m-order Fibonacci sequence defined by the recurrence (1), then $F_n \sim \lambda u^n$ for large n (fixed m), where

$$\lambda = \lambda_1 = \frac{(2-u_2)(2-u_3)\cdots(2-u_m)}{(u-u_2)(u-u_3)\cdots(u-u_m)} > 0,$$

and $u_i^{(m)} = u_i$ $(i > 1)$ are the conjugates of the PV-number $u = u_1$ in the polynomial (3). Moreover, $F_n \sim u^n$ for large n and large m.

PROOF. The solution of the recurrence

$$F_n - F_{n-1} - \cdots - F_{n-m} = 0$$

is clearly

$$F_n = \lambda_1 u_1^n + \lambda_2 u_2^n + \cdots + \lambda_m u_m^n,$$

where $u = u_1, u_2, \ldots, u_m$ are the roots of the polynomial (3), and $\lambda_1, \lambda_2, \ldots, \lambda_m$ are suitable constants. Since u is a PV-number, we have $|u_i| < 1$ for $i > 1$. Hence

$$\lambda_1 u_1^n + \lambda_2 u_2^n + \cdots + \lambda_m u_m^n \sim \lambda_1 u_1^n$$

for large n.

The recurrence (1) implies $F_i = 2^i$ for $0 \leq i < m$. Hence,

$$V\lambda_1 = \begin{bmatrix} 1 & 1 & & 1 \\ 2 & u_2 & \cdots & u_m \\ \vdots & \vdots & & \vdots \\ 2^{m-1} & u_2^{m-1} & & u_m^{m-1} \end{bmatrix},$$

where

$$V = \begin{bmatrix} 1 & 1 & & 1 \\ u_1 & u_2 & \cdots & u_m \\ \vdots & \vdots & & \vdots \\ u_1^{m-1} & u_2^{m-1} & & u_m^{m-1} \end{bmatrix} = \Pi_{k>\ell}(u_k - u_\ell)$$

is the Vandermonde determinant. This implies

$$\lambda_1 = \lambda = \frac{(2-u_2)(2-u_3)\cdots(2-u_m)}{(u_1-u_2)(u_1-u_3)\cdots(u_1-u_m)}.$$

By the Symmetric Polynomial Theorem it follows that λ is real. Since $F_n \sim \lambda u^n$ and F_n and u are positive, we have in fact $\lambda > 0$.

In view of Lemma 5 we can write $u_1 = u = 2 - \delta/2^{m-1}$, where δ is a suitable number in the range $1/2 < \delta < 1$. Thus

$$\lambda = \frac{(u_1-u_2+\delta 2^{-(m-1)})(u_1-u_3+\delta 2^{-(m-1)})\cdots(u_1-u_m+\delta 2^{-(m-1)})}{(u_1-u_2)(u_1-u_3)\cdots(u_1-u_m)}$$
$$= \left[1+\tfrac{\delta}{2^{m-1}(u_1-u_2)}\right]\left[1+\tfrac{\delta}{2^{m-1}(u_1-u_3)}\right]\cdots\left[1+\tfrac{\delta}{2^{m-1}(u_1-u_m)}\right].$$

Let $\varepsilon > 0$. For m sufficiently large

$$\frac{\delta}{2^{m-1}|u_1-u_i|} < \frac{\varepsilon}{m} \qquad (i = 2, \ldots, m).$$

Hence $|\lambda| < \left(1+\frac{\varepsilon}{m}\right)^m \to e^{\varepsilon}$, which can be made arbitrarily close to 1 for sufficiently large m. Thus $\lambda \to 1$ as $m \to \infty$. ∎

Theorem 4 enables us to give an asymptotic estimate of the length of any Fibonacci representation. We carry this out below for φ_3, but it is not much different for φ_1 and φ_2 (for which we get a slightly smaller asymptotic length).

Corollary 3 implies that the largest integer representable by φ_3 with $m+n$ bits is F_n $(n \geq 0)$. If k is the number of bits in the standard binary numeration system necessary to encode F_n, then $2^{k-1} \leq F_n < 2^k$. For large n we have by Theorem 4, $2^{k-1} \leq \lambda u^n < 2^k$, where $u = 2 - \delta 2^{-m+1}$, δ a suitable real number satisfying $1/2 < \delta < 1$. Then $k - 1 \leq n\, lg\, u + lg\, \lambda < k$, where lg denotes log to the base 2.

Expanding $lg\, u$ into a Taylor series,

$$\begin{aligned} lg\, u &= lg(2 - \delta 2^{-m+1}) = 1 + lg(1 - \delta 2^{-m}) \\ &= 1 - (\delta 2^{-m} + \delta^2 2^{-(2m+1)} + \cdots) lg\, e. \end{aligned}$$

Thus

$$k \approx n\big(1 - (\delta 2^{-m} + \delta^2 2^{-(2m+1)}) lg\, e\big) + lg\, \lambda. \tag{4}$$

In the logarithmic ramp representation $R^{(m)}(F_n^{(m)}) = R(F_n)$ of F_n, an extra 0-bit prefixes the string itself. Therefore,

$$|R(F_n)| = k + 1 + \lceil lg(k+1) \rceil + \lceil lg\lceil lg(k+1) \rceil + 1 \rceil + \cdots + 3,$$

where the last $lg\, lg \ldots$ term is 3. Using the approximation $k = n(1 - \delta 2^{-m} lg\, e) + lg\, \lambda$, the lengths difference is

$$\Delta = m - 1 + \frac{n\delta}{2^m} lg\, e - lg\, \lambda - \lceil lg(k+1) \rceil - \lceil lg\lceil lg(k+1) \rceil + 1 \rceil - \cdots - 3,$$

where k is given by (4).

This formula shows that for every fixed m, $\Delta > 0$ if n is sufficiently large, so ultimately the logarithmic ramp representation is shorter than any Fibonacci representation. But the crossover point depends exponentially on m. In fact, for very large initial values, the latter representations are in fact shorter than the former. The following computational results for $m = 2$, 3 and 4 refer to φ_1 and C_1.

We have $|R^{(2)}(n)| = 4$ bits and $|\varphi_1^{(2)}(n)| = 5$ bits for $n = 5, 6, 7$. But $|\varphi_1^{(2)}(n)| \leq |R^{(2)}(n)|$ for all integers n in the range

$$8 \leq n \leq F_{27}^{(2)} - 1 = 514,228.$$

Beyond this point, the representation $\varphi_1^{(2)}(n)$ becomes slowly larger than $R^{(2)}(n)$.

For $m = 3$, $|R^{(3)}(n)| < |\varphi_1^{(3)}(n)|$ for $3 \leq n \leq 7$. But $|\varphi_1^{(3)}(n)| \leq |R^{(3)}(n)|$ for all

$$8 \leq n \leq \frac{1}{2}\big(F_{63}^{(3)} + F_{61}^{(3)} - 1\big) = 34,696,689,675,849,696 \simeq 3.470 \times 10^{16}.$$

For larger n, $|\varphi_1^{(3)}(n)|$ becomes slowly larger than $|R^{(3)}(n)|$. Thus for $n = \frac{1}{2}\big(F_{80}^{(3)} + F_{78}^{(3)} - 1\big) \simeq 1.095 \times 10^{21}$ we have $|\varphi_1^{(3)}(n)| - |R^{(3)}(n)| = 1$, and this difference is 5, for example at $n = \frac{1}{2}\big(F_{146}^{(3)} + F_{144}^{(3)}\big) \simeq 3.208 \times 10^{38}$. Incidentally, the difference Δ does not increase monotonically: it usually decreases at points where $R(n)$ picks up a new $lg\,lg\ldots$ term on its logarithmic ramp.

We have $|R^{(4)}(n)| < |\varphi_1^{(4)}(n)|$ for $2 \leq n \leq 7$, $|\varphi_1^{(4)}(n)| \leq |R^{(4)}(n)|$ for $8 \leq n \leq 116$, and $|\varphi_1^{(4)}(n)| - |R^{(4)}(n)| = 1$ for $117 \leq n \leq 127$. But $|\varphi_1^{(4)}(n)| \leq |R^{(4)}(n)|$ for all

$$128 \leq n \leq \frac{1}{3}\big(F_{231}^{(4)} + 2F_{229}^{(4)} + F_{228}^{(4)} - 1\big) \simeq 4.194 \times 10^{65}.$$

Beyond this point, the representation $\varphi_1(n)$ becomes very slowly larger than $R^{(4)}(n)$.

These computational results and the asymptotic formula for Δ (valid for φ_3), both indicate that $|\varphi_1^{(m)}(n)| \leq |R^{(m)}(n)|$ for exponentially larger n as m increases. Hence if we expect many of the transmitted strings to be very large, it may be advantageous to select a larger value of m than for the transmission of shorter strings.

5. Universality of Fibonacci Codes and Representations

Let C be a countably infinite UD binary code, and $M = \{m(1), m(2), \ldots\} \supset Z^+$ a countable set of messages. Let $S = \{(1, p_1), (2, p_2), \ldots, (n, p_n)\}$ be a source of the first n positive integers, with associated positive probabilities $p_1 \geq p_2 \geq \cdots \geq p_n$ ($\sum_{i=1}^n p_i \leq 1$). The source may also include some noninteger messages with their associated probabilities. The entropy of the integers in the source is $H(P) = -\sum_{i=1}^n p_i \, lg \, p_i$. Let $\varphi : Z^+ \rightarrow C$ be a binary representation of the positive integers such that $|\varphi(i)| \leq |\varphi(i+1)|$ ($i \geq 1$). Then φ is called *universal* if

$$\frac{E_P(L)}{\max\{1, H(P)\}} \leq K,$$

where $E_P(L) = \sum_{i=1}^n p_i |\varphi(i)|$ is the expected codeword length of the representations of the integers in the source, and K is a positive constant independent of the probability distribution $P = \{p_1, p_2, \ldots, p_n\}$. This definition reduces to Elias' universality definition when there are no noninteger messages.

Let $f = \{f_i(x)\}_{i=1}^k$ be a finite sequence of polynomials with $deg(f_1) \geq deg(f_j) > 0$ for all j satisfying $1 \leq j \leq k$. All polynomial coefficients are constants which may depend on m. For simplicity we assume that all coefficients of f_1 are nonnegative.

A more general notion of Fibonacci representation than used above will now be introduced.

A *Fibonacci representation* of a set $M \supset Z^+$ is any binary representation ψ such that all and only all the first $\mathcal{L}(F_{f(\ell)}) + d$ positive integers have representations of length up to ℓ, where $\mathcal{L}$ denotes a finite linear combination: $\mathcal{L}(F_{f(\ell)}) = \sum_i c_i F_{f_i(\ell)}$, where the c_i and d are constants which may depend on m, and $c_1 > 0$.

By applying the definition successively to $\ell_{\min} = |\psi(1)|, \ell_{\min}+1, \ell_{\min}+2, \ldots$, it follows that ψ is a representation of M if and only if all the positive integers n in the interval

$$\left[\mathcal{L}(F_{f(\ell-1)}) + d + 1, \mathcal{L}(F_{f(\ell)}) + d\right]$$

have representation length $|\psi(n)| = \ell$ for all $\ell \geq \ell_{\min}$.

Note that φ_1, φ_2 and φ_3 are representations also according to the new definition: By Corollary 2, the φ_1-representations of precisely the first S_{n-1} positive integers have length up to $m+n$. Furthermore, we have

$$S_{n-1} = \mathcal{L}(F_{f(m+n)}) + d = \frac{1}{m-1}\left(F_{n+1} + \sum_{i=0}^{m-3}(m-2-i)F_{n-1-i} - 1\right),$$

where

$$f_1(n+m) = n+m-(m-1), \quad c_1 = \frac{1}{m-1}, \quad d = -\frac{1}{m-1},$$

$$f_i(n+m) = n+m-(m-1+i), \quad c_i = \frac{m-i}{m-1} \quad (i = 2, \ldots, m-1).$$

Let $f = \{f_1\}$. For $f_1(m+n+1) = m+n+1-(m+1)$, $c_1 = 1$, $d = -1$, Lemma 3 implies that precisely the first $\mathcal{L}\big(F_{f(m+n+1)}\big) + d = F_n - 1$ positive integers have φ_2-representations of length not exceeding $m+n+1$. For $f_1(m+n) = m+n-(m)$, $c_1 = 1$, $d = 0$, Corollary 3 implies that precisely the first $\mathcal{L}\big(F_{f(m+n)}\big) + d = F_n$ positive integers have φ_3-representations of length not exceeding $m + n$.

LEMMA 6. Let ψ be a binary representation such that $|\psi(k)| \le c_1 + c_2 \, lg \, k$ $(k \in Z^+)$, where c_1 and c_2 are constants and $c_2 > 0$. Let $p_k = p(k)$ be the probability of k. If $p_1 \ge p_2 \ge \cdots \ge p_n$, $\sum_{i=1}^{n} p_i \le 1$, then ψ is universal.

PROOF. For $1 \le j \le n$ we have $1 \ge \sum_{i=1}^{j} p_i \ge jp_j$, so $lg \, j \le -lg \, p_j$. Hence

$$\sum_{j=1}^{n} p_j \, lg \, j \le -\sum_{j=1}^{n} p_j \, lg \, p_j = H(P),$$

and

$$\begin{aligned} E_P(L) = \textstyle\sum_{i=1}^{n} p_i |\psi(i)| &\le \textstyle\sum_{i=1}^{n} p_i (c_1 + c_2 \, lg \, i) \\ &\le c_1 + c_2 \textstyle\sum_{i=1}^{n} p_i \, lg \, i \le c_1 + c_2 H(P). \end{aligned}$$

Thus

$$\frac{E_P(L)}{\max\{1, H(P)\}} \le \begin{cases} c_1 + c_2 \text{ for } H(P) \le 1 \\ \frac{c_1}{H} + c_2 < c_1 + c_2 \text{ for } H(P) > 1. \end{cases} \quad \blacksquare$$

THEOREM 5. Any Fibonacci representation is universal.

PROOF. If ψ is a Fibonacci representation, then any integer $n \in \big[\mathcal{L}\big(F_{f(\ell-1)}\big) + d + 1, \mathcal{L}\big(F_{f(\ell)} + d\big]$ has representation length $|\psi(n)| = \ell$ for all $\ell \ge \ell_{\min}$. Thus

$$n \ge \mathcal{L}\big(F_{f(\ell-1)}\big) + d + 1 \ge c_1 F_{f_1(\ell-1)} + d \ge c_1 F_{a_1(\ell-1)} + d,$$

where a_1 is the leading coefficient of f_1. By Theorem 4, $F_k \sim \lambda u^k$ where $\lambda > 0$, so $F_k > K u^k$ for all $k \ge 0$, where $K > 0$ is a suitable constant. Thus

$$n \ge c_1 K u^{a_1(\ell-1)} + d,$$

so

$$\ell \le \frac{1}{a_1}\Big(\log_u \frac{n-d}{c_1 K}\Big) + 1 = \frac{1}{a_1}\big(\log_u (n + K_1) - \log_u (c_1 K)\big) + 1,$$

where $K_1 = -d$ and $n + K_1 > 0$.

If $K_1 \le 0$, then $\log_u (n + K_1) \le \log_u (n)$. So assume $K_1 > 0$.

If $n \ge K_1$, then $\log_u (n + K_1) \le \log_u (2n) = \log_u 2 + \log_u n$.

If $n < K_1$, then $\log_u(n+K_1) < \log_u(2K_1)$. Thus in all cases $\log_u(n+K_1) \leq K_2 + \log_u n$ for a suitable constant $K_2 > 0$. Thus

$$|\psi(n)| = \ell \leq K_3 + K_4\, lg\, n$$

for suitable constants K_3 and $K_4 > 0$. Now apply Lemma 6. ▮

References

1. J.L. Bentley and A.C-C. Yao, An almost optimal algorithm for unbounded searching, *Inform. Process. Letters* 5 (1976) 82-87.
2. J.W.S. Cassels, *An Introduction to Diophantine Approximation*, Cambridge University Press, Cambridge, 1957.
3. P. Elias, Universal codeword sets and representations of the integers, *IEEE Trans. Inform. Theory* IT-21 (1975) 194-203.
4. S. Even, *Graph Algorithms*, Computer Science Press, Potomac, MD, 1979.
5. S. Even and M. Rodeh, Economical encoding of commas between strings, *Comm. ACM* 21 (1978) 315-317.
6. A.S. Fraenkel, Systems of numeration, *Amer. Math. Monthly* 92 (1985) 105-114.
7. A.S. Fraenkel and S.T. Klein, Robust constant codes as alternatives to Huffman codes, Department of Applied Mathematics, The Weizmann Institute of Science, 1985.
8. R.G. Gallager, *Information Theory and Reliable Communication*, John Wiley, New York, 1968.
9. E.N. Gilbert, Synchronization of binary messages, *IRE Trans. Inform. Theory* IT-6 (1960) 470-477.
10. L.J. Guibas and A.M. Odlyzko, Maximal prefix-synchronized codes, *SIAM J. Appl. Math.* 35 (1978) 401-418.
11. W.H. Kautz, Fibonacci codes for synchronization control, *IEEE Trans. Inform. Theory* IT-11 (1965) 284-292.
12. D.E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, MA, Second Printing, 1975.
13. K.B. Lakshmanan, On universal codeword sets, *IEEE Trans. Inform. Theory* IT-27 (1981) 659-662.
14. E.P. Miles, Jr., Generalized Fibonacci numbers and associated matrices, *Amer. Math. Monthly* 67 (1960) 745-752.

15. M. Rodeh, V.R. Pratt and S. Even, Linear algorithm for data compression via string matching, *J. ACM* **28** (1981) 16-24.

16. R.G. Stone, On encoding of commas between strings, *CACM* **22** (1979) 310-311.