Non-Binary Robust Universal Variable Length Codes

Shmuel T. Klein[a] Tamar C. Serebro[b] Dana Shapira[b]

[a] Bar Ilan University, Israel [b] Ariel University, Israel

Data Compression Conference

Contribution

We extend the binary Fibonacci code to d-ary codes, with $d \geq 2$. This is motivated by future technological developments in which the basic unit of storage will not be just a 2-valued bit, but possibly an element that is able to distinguish between d different values. The proposed codes are prefix-free, complete and more robust than Huffman codes. Experimental results illustrate that the compression efficiency of non-binary Fibonacci codes are very close to the savings achieved by the corresponding non-binary Huffman coding of the same order.

Fibonacci Code

The Fibonacci code is a binary representation of the integers based on the numeration system composed of the Fibonacci sequence: any integer x can be written as $x = \sum_{i \geq 2} f_i F_i$, with $f_i \in \{0, 1\}$.

The Fibonacci Code is the set of codewords which consists of all the binary strings in which the substring 11 appears exactly once, at the right end of the string. This yields the prefix code:
$\mathcal{E}_{fib}$ = {11, 011, 0011, 1011, 00011, 10011, 01011, 000011, 100011, 010011, 001011, 101011, 0000011,...}.

We explore the extension of the binary Fibonacci code to d-ary codes, with $d \geq 2$. This might be motivated by future technological developments in which the basic unit of storage will not be just a 2-valued bit, but possibly an element that is able to distinguish between d different values.

Generalization of Fibonacci

Consider the following generalization of the standard Fibonacci sequence, depending on a parameter $m \geq 1$. Define the family of sequences

$$R_{-1}^{(m)} = 1, \quad R_0^{(m)} = 1,$$

$$R_i^{(m)} = mR_{i-1}^{(m)} + R_{i-2}^{(m)} \quad \text{for } i > 0.$$

For $m = 1$, this is the standard Fibonacci sequence F_i. The numeration system based of the sequence $\mathcal{R}^{(m)} = \{R_0^{(m)}, R_1^{(m)}, R_2^{(m)}, \ldots\}$ is a $d = (m+1)$-ary system, i.e., any integer L can be uniquely represented as $L = \sum_i a_i R_i^{(m)}$, where the coefficients a_i of the basis elements are not just binary, but belong to a larger set $0 \leq a_i \leq m$. The additional property, generalizing the non-adjacency of 1-bits of the Fibonacci encoding, is that if a_{i+1} reaches its maximal permitted value m, then the digit a_i just preceding it, if there is such a digit, has to be zero.

Ternary numeration system

As example, the elements of the ternary numeration system $\mathcal{R}^{(2)}$ are $\{1, 3, 7, 17, 41, \ldots\}$, and the sequence of the first codewords, representing the integers 1 to 20 according to $\mathcal{R}^{(2)}$, is

$$1, 2, 10, 11, 12, 20, 100, 101, 102, 110, 111, 112, 120, 200, 201, 202, 1000, 1001, 1002, 1010, \ldots.$$

m-ary extension of the Fibonacci code

Turning an m-ary representation into a useful code can be done by the following steps:

1. exploit the fact that integers are represented without leading zeros, so that the leftmost digit is one of $\{1, 2, \ldots, m\}$; we can thus prefix the representation of any integer by the digit m, which will act as a comma between codewords, because within a codeword, the digit m must be followed (in a left to right scan) by 0.

2. reverse all the codewords, thereby turning the resulting set into a prefix code, which is instantaneously decodable.

An alternative, equivalent, definition of this code for $m = 2$ is the sequence of all ternary strings, each terminating with a rightmost trit equal to 2, and with the constraint on every other trit, which is not in the leftmost position, that if it is equal to 2, then it is preceded by 0.

Properties

1. Prefix:

Every codeword w terminates on its right end by a pair of digits xm, with $x \neq 0$. It follows that w can not be the prefix of any longer codeword, because such xm does not appear anywhere else in any codeword.

2. Robustness

3. Completeness:

LEMMA 1: *For all $m \geq 1$, the infinite code obtained by the m-ary extension of the Fibonacci code is complete.*

Average codeword length as function of m