Journal of Universal Computer Science, vol. 21, no. 9 (2015), 1136-1142
submitted: 11/2/15, accepted: 14/7/15, appeared: 1/9/15 © J.UCS

A Note on Variable-Length Codes with Constant Hamming Weights

Peter Fenwick
(The University of Auckland, New Zealand (retired)
pmbjfw@gmail.com)

Abstract: A recent paper described a variable-length integer code based on the Goldbach conjecture where every codeword had exactly 2 1-bits but with an extremely irregular structure. A later, unpublished, work produced a much more regular code, again with a Hamming weight of 2. This paper extends that later work to weight-3 and weight-4 codes, which are shown to be competitive with more-usual codes over a useful range of values.
Key Words: Variable-length code, Hamming-weight 3
Category: E.4

1 Introduction

Variable length codes (more generally "Universal Codes", but see the concluding comment) represent integers by some permutation of bits, where —

- The codeword for frequent values (usually small values) is short, and
- The codeword is self-delimiting, including information to specify its length.

These codes are of great importance in many forms of data compression, where the coder output is a sequence of integers with a skew distribution and small values the most frequent. A survey of such codes is given by Fenwick [Sayood 2003], extended in [Fenwick 2014].

Some representative codes, relevant to the present discussion, are —

Elias α code The code for n is simply n zeroes followed by a one (or vice versa) [Elias 1975]. It is a basic component of many variable length codes not mentioned here.

Elias γ code The natural binary representation is transmitted least-significant bit first, with lesser-significant bits preceded by a 0 "flag" bit [Elias 1975]. The most significant 1 bit is replaced by a single 1 flag (which implies the data 1.) A variation, the γ' code, transmits the flag bits first and then data bits in descending order. (The leading part is an α code defining the length.)

Fraenkel-Klein codes These codes [Fraenkel and Klein 1996] are based on the Fibonacci numbers (1, 2, 3, 5, 8, 13, ... where each number is the sum of the two preceding numbers). As Zeckendorf has shown that any integer can be represented as the sum of Fibonacci numbers [Zeckendorf 1972], an integer is represented as a binary vector or mask, with a 1 indicating the inclusion of the corresponding Fibonacci number, ignoring one of the repeated initial 1s. Thus 12_{10} is represented by 10101000... $(1+3+8$, least-significant bit first). Now as each Fibonacci number is the sum of the two preceding numbers, this "Zeckendorf" representation can never have two adjacent 1s. The codeword is then terminated by following the most-significant 1 (or final 1) by another 1, The code for 12 is 101011.

Ternary code This code [Fenwick 1993] represents a value in base-3 using the digit pairs 00, 01 and 10, with the pair 11 acting as a final, terminating, "comma".

Ibsen Code Fenwick [Fenwick 2002] proposed a completely different code, based on the Goldbach conjecture that every integer is the sum of two primes. The bit vector now acts as a mask indicating which primes are to be added; the second 1 acts also as a terminator. (An extension is needed to encode odd values.) Although an effective code, the codeword length varies erratically with the encoded value.

The concept was refined by Ibsen [Ibsen 2006] [Fenwick 2014] who described a code with a more regular structure and, again, a Hamming weight of 2. With the bits numbered $\{0, 1, 2, 3, 4, \ldots\}$ (0-origin), the first 1 (leftmost, in position i) has a weight of i and the second (rightmost) bit in position k has a weight of $k(k-1)/2$ (with an offset when counting from other than 0). These weights will be justified later in Section 3.

Examples of some representations are given in Table 1[1]. All have the expected property that small values have shorter codewords.

The γ code has the big advantage that its code for 1 (the smallest value) is just one bit. This makes it especially useful for highly skewed distributions in which the smallest value often has a frequency of 30–50%, or more.

The Fraenkel-Klein codes are better for flatter distributions, being shorter than the gamma codes for values of 4 or greater. They are a good general-purpose code and will be used as a reference for comparisons.

The ternary code naturally starts at a value of 0 and here uses an offset of 1. Its performance is similar to that of the Frankel-Klein codes. Although its coding is inherently a little shorter, it expands by adding 2 bits rather an 1; the two largely cancel out.

---

[1] The Ibsen and ternary codes naturally start from 0 and may be offset by 1 in the examples to match the Elias γ and Fraenkel-Klein codes which start from 1.

The Ibsen codes (also offset by 1) are best for values from 2 to 10 and are in fact competitive with the Fraenkel-Klein for values to about 40.

The codeword lengths for a value N are of the order $\log_{1.4} N$, $\log_{1.6} N$, $\log_{1.7} N$ and $\sqrt{N}$ respectively (actually $\log_{\sqrt{2}} N$, $\log_{\phi} N$, $\log_{\sqrt{3}} N$ and $\sqrt{N}$) [2].

Table 1: Examples of some variable-length codes

Value	Elias γ	Fraenkel-Klein	Ternary	Ibsen
1	1	11	11	11
2	001	011	0111	101
3	011	0011	1011	011
4	00001	1011	000111	1001
5	01001	00011	010111	0101
6	00011	10011	100111	0011
7	01011	01011	001011	10001
8	0000001	000011	011011	01001
9	0100001	100011	101011	00101
10	0001001	010011	00000111	00011
50	00010000011	001001011	0101100111	00001000001
75	0101000100001	0101010011	1000101011	0000000010001

2 The New Weight-3 Code

The Ibsen code is competitive over a modest range of values, even with the severe limitation of allowing only two bits to be 1. What then might be the performance of a similar code with a Hamming weight of 3?

The principle is an obvious extension of the Ibsen code. We assume that bits are written left to right, in increasing significance, starting with all three bits adjacent in the leftmost 3 bits. But consider first the more general case, of some arbitrary bit pattern. For increasing values, the first (or least-significant) bit moves to the right until it "hits" the second bit. The second bit then moves one place to the right and the first bit starts again from the left (equivalent to a carry). And if at any stage the second bit moves on to the third bit, the third bit will of course move right to a more significant position and the other two bits will revert back to the leftmost positions.

The first few values and their representations are shown in Table 2.

[2] ϕ is the golden ratio 1.618...

Table 2: Examples of Weight 3 code

Value	Codeword	Value	Codeword	Value	Codeword
0	111	4	11001	8	01011
1	1101	5	10101	9	00111
2	1011	6	01101	10	110001
3	0111	7	10011	11	101001

3 The Bit Weightings.

The fundamental observation is that a *more-significant* bit allows only a certain number of *combinations* of less-significant bits, rather than all possible combinations as in conventional representations. When all lower combinations have been exhausted the bit moves up by one place, equivalent to an overflow or carry.

Table 3: Development of Ibsen code

2 out of 4 bits	Ibsen code	value	components
1 1 0 0	1 1	0	0 + 0
1 0 1 0	1 0 1	1	0 + 1
0 1 1 0	0 1 1	2	1 + 1
1 0 0 1	1 0 0 1	3	0 + 3
0 1 0 1	0 1 0 1	4	1 + 3
0 0 1 1	0 0 1 1	5	2 + 3
1 0 0 0 1	1 0 0 0 1	6	0 + 6

To illustrate the underlying principles, consider an Ibsen code in Table 3. The left column shows all ${}^4C_2 = 6$ combinations of 2 out of 4 bits. The next column shows the trailing zeros omitted, to give an Ibsen code, whose value follows. (Note that, for example, the third bit position (index 2) has a weight of 2 if it is the *first* bit, but 1 if it is the *second* bit.) The next codeword, shown in the last line, must be $10001 = 6$, with the ones having weights of 0 and 6. Generally, it will seen that the cluster of left-most 1s all have weights of 0; the newly-shifted bit has a weight of the next sequential value. The principle clearly extends to longer codewords.

For the 3-bit code, assume that the three bit indices are i, j, k, where $i < j < k$, 0-origin, and that the bit weights in the different positions are W_i, W_j and W_k. The value is then $W_i + W_j + W_k$, possibly adjusted by a constant to set the origin of the values.

Then we have —

- The least-significant bit, index i, simply steps by 1 from an initial value of 0 as values increase (an α code), and

$$W_i = i$$

- For the "middle" bit, index j, all lower-valued codes have exactly 2 1-bits out of j codeword positions (an Ibsen code). There are thus ${}^jC_2$ preceding values ($0 \ldots ({}^jC_2 - 1)$), and the next value must be the weight for this bit.

$$W_j = {}^jC_2 = \frac{j(j-1)}{2!}$$

- Similarly for the most significant bit, index k, the preceding values (again including 0) use all combinations of 3 of the k bits.

$$W_k = {}^kC_3 = \frac{k(k-1)(k-2)}{3!}$$

If the value range starts at 1 it is necessary to encode 1 less than the desired value; this is done in Table 4 below to facilitate comparison with other codes.

Generation of a codeword exactly parallels the steps for arbitrary base conversion — find the largest possible most-significant value (here corresponding to the last bit position) subtract that value from the running value and repeat for successively lower positions.

Table 4: Comparison of code-word lengths (Weight-3 values offset by 1)

Value	Gamma	F-Klein	Weight 3		Value	Gamma	F-Klein	Weight 3
1	1	2	3		10	7	6	5
2	3	2	4		30	9	8	7
3	3	3	4		100	13	11	10
4	5	4	4		200	15	12	12
5	5	4	5		287	17	13	14
6	5	5	5		300	17	13	14
7	5	5	5		500	17	14	16
8	7	5	5		1000	19	16	20
9	7	6	5		2000	21	17	24

4 Comparisons

Table 4 compares the codeword lengths for mostly small values and a selection of large values. (287 is the value where the Weight-3 code becomes more expensive than the Fraenkel-Klein code. The table also shows that the Weight-3 code is shortest-equal from 6, and shortest from 9.)

The γ code is best for small values, but is relatively poor for values greater than 3. The Fraenkel-Klein code is generally better than the γ code, except that 1 is encoded as 2 bits, compared with 1 for the γ. The ternary code (not shown) resembles the Fraenkel-Klein code, but see the earlier comments. The new Weight-3 code fares poorly for very small values, needing at least 3, and then 4 bits. But from 6 it is the best, or best equal of the three codes for values to about 300. (Actually 287, see above.)

5 Extension to larger weights

Jørgen Ibsen has pointed out that we now have a sequence of codes of increasing Hamming weights – the Elias α code (weight 1), his code (weight 2) and the present one (weight 3). Corresponding codes clearly exist for larger weights and may be generated by obvious extension of the methods of Section 3. Table 5

Table 5: Some comparative codeword lengths

Value	Weight-4 Length	Weight-5 Length	F-Klein Length	Gamma Length
1	4	5	2	1
3	5	6	4	5
5	5	6	5	5
7	6	7	5	5
15	6	7	7	7
35	7	8	9	9
1,365	15	14	16	21
1,820	16	14	17	21
2,380	17	15	17	23
3,060	18	16	18	23
3,876	19	16	18	23

compares the lengths of the weights 4 and 5 codes and the Frankel-Klein code for the largest values for each weight-4 length. (It also shows the γ code, which is clearly longer for all but very small values. The weight-5 code is included, for comparison, using these chosen values.) Inverting the formulas for the bit positions (and adding an observed adjustment) gives the length of the k-bit code for a value N as approximately

$$\text{Codeword length} \approx \sqrt[k]{k!N} + 2$$

The weight-4 code is the shortest or shortest-equal for values from 8 to 3,060. The weight-5 code is longer than the weight-4 to about 35 and often 1 bit shorter from about 200 to about 15,000. (The weight-6 code is then shorter than Fraenkel-Klein to about 400,000, and the weight-7 to about 4 million.)

Constant-weight codes, and especially the lower-weight ones, are therefore worth considering for broadly-peaked distributions with few extreme values.

6 Final Comment

Finally, "Universal Codes" are often synonymous with "Variable Length" codes as being both self-delimiting and able to represent any value. But the smaller-weight codes, although they can in fact represent any integer, are so inefficient for large values that they hardly deserve to be described as "Universal". (The weight-4 code for 1 million requires 72 bits, and the weight-3 code 183!)

Acknowledgments

Jørgen Ibsen developed the original systematic weight-2 code and then pointed out the sequence of Hamming weights as noted in Section 5. His contributions were essential to the present work.

References

[Elias 1975] P.Elias, "Universal Codeword Sets and Representations of the Integers", *IEEE Trans.Info.Theory*, Vol IT 21, No 2, pp 194–203, Mar 1975

[Fenwick 1993] P.M.Fenwick, "Ziv-Lempel encoding with multi-bit flags", *Proc.Data Compression Conference, DCC-93*, Snowbird, Utah, pp 138–147, Mar 1993

[Fenwick 2002] P.M.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, Aug 2002.

[Fenwick 2014] P. M. Fenwick *Introduction to Computer Data Representation*, Bentham Books, 2014. eISBN 978-1-60805-882-2 [Chapter 9]

[Fraenkel and Klein 1996] A.S.Fraenkel and S.T.Klein, "Robust universal complete codes for transmission and compression", *Discrete Applied Mathematics* Vol 64 (1996) pp 31–55.

[Ibsen 2006] J.Ibsen, Private Communication, 2006

[Sayood 2003] K.Sayood (ed), *Lossless Compression Handbook*, Academic Press, 2003 [Chapter 3]

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