On Universal Codes for Integers: Wallace Tree, Elias Omega and Beyond

Lloyd Allison[1,*], Arun S. Konagurthu[1], Daniel F. Schmidt[1]
[1] Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia.
*Corresponding author: Lloyd Allison (lloyd.allison@monash.edu)

Abstract

A universal code for the (positive) integers is a variable length code that can be used to store or compress a sequence of integers. It also implies a probability distribution on integers which can be a natural choice when the true distribution of a source of integers is unknown; such a code and distribution may be useful in statistical inference. This paper provides two improvements to the theory and practice of universal codes. First, it defines and examines a new universal code omega* (omega-star) that asymptotically beats the Elias omega code. Second, it analyses the properties of a code proposed by Wallace based on trees, and shows it to be a universal code, to have desirable properties for use in inference, and to beat the Elias omega code on almost all integers up to the 1697-bit code-word mark. Encoding and decoding routines for the codes described here are implemented and available for interactive use.[1]

1 Introduction

Universal codes for positive integers, $N \in \mathbb{Z}^+$, are of interest for at least three reasons. The first is in everyday data compression where such a code can be used to store or to transmit a sequence of integers when their true probability distribution is not known. The second is in inductive inference when a countable set of hypotheses $\{H_i\}$ in a statistics and machine learning task is mapped to the set of positive integers: If the true distribution of the set of hypotheses is unknown but they can be plausibly ordered in non-increasing probability, then the i^{th} hypothesis can be assigned the probability $2^{-|w(i)|}$, where $w(i)$ is the code-word of integer i and $|w(i)|$ is its length. Note if i and j are close, H_i and H_j should have similar probabilities and hence similar code-word lengths. Finally, it must also be admitted that there is simply fun to be had in trying to devise an efficient code for truly enormous integers.

Elias [1] defined a code having the *universal* property as one where the code-word length is monotonically increasing and "assigning messages in order of decreasing probability to codewords in order of increasing length gives an average code-word length, for any message set with positive entropy, less than a constant times the optimal average codeword length for that source." If the source has distribution $\Pr(\cdot)$ and entropy $H = \sum_{\forall N>0} \Pr(N) \cdot \log\left(\frac{1}{\Pr(N)}\right)$ then, for any universal code $w(\cdot)$ for positive integers, $E_w = \sum_{\forall N>0} \Pr(N) \cdot |w(N)| < K \cdot H$, where K is a constant *independent* of $\Pr(\cdot)$. The latter sum is at least finite, although the distribution

[1] The codes may be tried at www.allisons.org/ll/MML/Discrete/Universal/ ←click.

Data Compression Conference (DCC), pp.313-322, doi:10.1109/DCC50243.2021.00039, March 2021

implied by a universal code must itself have infinite entropy. Naturally the hope is that K is not large. Elias also defined an *asymptotically optimal* code as one where the ratio $\frac{E_w}{\max(1,H)} \leq R(H) \leq K$, where R is a function of H with $\lim_{H\to\infty} R(H) = 1$.

Wallace proposed a code for integers [2] inspired by binary trees. He suggested that its implied probability distribution is a good choice in inductive inference if the true distribution of a source of integers is unknown.

In the following sections, the Elias omega code, our improvement on it which we call the omega* (omega-star) code, and an effective implementation of the Wallace tree code (WTC) are analysed and compared. Encoding/Decoding routines and asymptotic analysis are given.

2 Elias omega (ω) code for integers

We introduce a version on the Elias omega code that gives identical code-word lengths to the original definition [1] and differs only in minor details. The code-word for an integer $N \geq 1$ consists of one or more sections: zero or more *length* sections followed by one *value* section. The value section is simply the minimal binary representation of N in $\lfloor \log_2(N) \rfloor + 1$ bits; note, the value section starts with a '1'.

The code-word for $N = 1$ is simply "1"; it is the only code-word that starts with a '1'. The code-word for $N \geq 2$ has at least one length section before the final value section. In general the value section by itself is not sufficient because a decoder does not know how long it is. The solution is to first encode the length of the value section minus one (the length must be ≥ 2 when $N \geq 2$), recursively, until the length-1 of a length-1 of ... of a length-1 gives one.

The leading bit of each section would, on the face of it, be a '1' so that position can instead be used as a flag to indicate either a length section ("0...") or the final value section ("1..."). The decoder notes the flag. In the case of a '0' it then switches it to a '1' before computing the length of the next section. If present ($N \geq 2$) the first length section is just "0" which stands for one. If $N \geq 4$ the second length section is either "00" which stands for two or "01" which stands for three, and so on. (See table 1 for some examples.)

Note that the omega code is similar to, and can be thought of as an optimized and shifted version of, the Levenstein code [3] which is defined for $N \geq 0$. Also note that Rissanen [4] defined the $\log^*$ code as an approximation to the omega code and advocated its use in inductive inference; $\log^*(N) = c + \log_2(N) + \log_2(\log_2(N)) + \log_2(\log_2(\log_2(N)))\ldots$, all positive terms, where c is a normalising constant, and $\Pr(N) = 2^{-\log^*(N)}$.

3 Omega variations and improvements

Observe that the Elias omega code in effect uses a unary code ("0..." $\Rightarrow$ length section, "1..." $\Rightarrow$ final value section) to indicate the number (≥ 1) of sections in a code-word. Elias chose the name omega for the code because he considered it to be "penultimate", that is "not quite ultimate" (p.200) [1]. (That being so, the name *psi*, say, would have left some room to name codes that are closer to the ultimate.) He

```
function omega_r_enc(t_enc)
 { function enc(N)  // a bigInt N >= 1
    { var todo=N, nSect, nTet, CW="";
      for( nTet = 1; ; nTet ++ )
       { for( nSect = 1; ; nSect ++ )
          { var section = todo.binary();
            var len = section.length;
            if( len == 1 ) break;
            section = section.substring(1,len); // trim
            CW = section + CW;
            todo = bigInt.fromInt(len-1);
          }//for nSect
         if( nSect == 1 ) break
         todo = bigInt.fromInt(nSect-1);
       }//for nTet
      CW = t_enc(bigInt.fromInt(nTet)) + CW;  // !
      return CW;
    }//enc(N)
   return enc;
 }//omega_r_enc(t)

/* and */ function omega_star_enc(N) = omega_r_enc(omega_enc)(N);
```

Figure 1: Encoders for $\text{omega}_\text{r}(\text{t})(\text{N})$ and $\text{omega}^*(\text{N})$ in JavaScript-styled pseudocode.

noted that the unary code could be replaced with his gamma, delta or omega codes. In fact the leading bits of the sections can be moved to the front of the code-word and the unary code can be replaced by *any* other code for positive integers – say WTC.

Define $\text{omega}_\text{p}(\text{s})$ to be the Elias omega code modified and *parameterised* to use an integer code 's' for the number of sections. The code-word for 1 is "1". For $N > 1$, the code-word is the omega code-word for N with the leading bit of every section trimmed away, and the result preceded by the 's' code-word for the number of sections. $\text{omega}_\text{p}(\text{unary})$ is equivalent to the usual omega code. Let $\text{omega}^2 = \text{omega}_\text{p}(\text{omega})$ which uses the omega code for the number of sections. This code would make code-words of two, four and five sections longer and those of seven or more sections shorter than the usual Elias omega code.

However even omega^2 is not ultimate. It is possible to define a code, $\text{omega}_\text{r}(\text{t})$, that uses itself, recursively, to state the number of sections (minus one) in an omega code-word. Note that $\text{omega}_\text{r}(\cdot)$ needs some *other* integer code, 't', to encode the number of *tetrations*, i.e., the number of times that omega_r is applied. Define $\text{omega}^* = \text{omega}_\text{r}(\text{omega})$. For example, $N = 36$ is encoded using omega^* as follows

$$N = 36 \implies \text{trim}(\text{omega}(36)) = \emptyset\ \emptyset 0\ \emptyset 01\ \not{1}00100 = \quad 0\ 01\ 00100 \quad (1)$$

$$\#\text{sections} = 3 \implies \text{trim}(\text{omega}(3)) = \emptyset\ \not{1}1 = \quad 1 \quad (2)$$

$$\#\text{sections} = 1 \implies \text{trim}(\text{omega}(1)) = \not{1} \quad \text{Stop} \quad (3)$$

$$\#\text{tetrations} = 3,\ \text{encoded using omega}(3) = \qquad 0\ 11 \qquad (4)$$
$$\implies \text{omega}^*(36) = \qquad 011100100100 \qquad (5)$$

where spaces are added in code-words merely for readability, and trim(.) removes the leading bit of each length and value section of recursively applied omega code-words.

Omega* makes code-words of integers with nine or more sections shorter than under omega; the smallest such integer is $2 \uparrow\uparrow 8$ in Knuth's up-arrow notation. Encoders for $\text{omega}_\text{r}(\text{t})$ and omega* are given in Fig. 1; the decoder adopts similar logic in reverse.

Although omega* might be called post-penultimate it is not ultimate because one can reconsider the encoding of the number of tetrations but the integers for which there would be any further improvement would be "very large" indeed.

4 Wallace tree code (WTC) for integers

The code for integers proposed by Wallace [2, 5] is based on full binary trees, and hence depends on the Catalan numbers $C_f, \forall f \geq 0$, with the first few numbers being $C_0 = 1, C_1 = 1, C_2 = 2, C_3 = 5, C_4 = 14$, and so on.

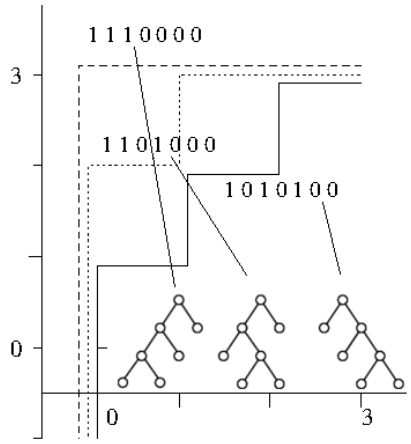
Any full binary tree consists of $f \geq 0$ *fork* (i.e. internal) nodes and $f + 1$ *leaf* nodes. Each fork node has *exactly* two sub-trees and each leaf node has *zero* sub-trees. The number of full binary trees containing f fork nodes is the f^{th} Catalan number, defined as: $C_f = \frac{1}{f+1}\binom{2f}{f} = \frac{(2f)!}{(f+1)!f!}$. This yields the recurrence $C_{f+1} = \frac{2(2f+1)C_f}{f+2}$. Furthermore, $\lim_{f\to\infty} \frac{C_{f+1}}{C_f} = 4$, and $C_{f+1} = \sum_{j=0}^{f}(C_j \cdot C_{f-j})$ [6]. Finally, we will need the *cumulative* Catalan numbers $cC_f = \sum_{j=0}^{f} C_f, \forall f \geq 0$, which are $cC_0 = 1, cC_1 = 2, cC_2 = 4, cC_3 = 9, cC_4 = 23$, and so on.

As a preliminary matter, note that a full binary tree can be encoded [7] during a pre-order traversal of the tree, outputting a '1' for each fork and a '0' for each leaf (e.g. see Fig. 2a). The end of the tree's code-word is indicated upon reaching one more '0' than '1's, and this event does not happen earlier within the code-word so this is a prefix code. It is easy to recover the tree from the code-word.

Reading a '1' as left and a '0' as down, a tree's code-word can also be interpreted as encoding a Dyck path [8] in a square lattice from some, initially unknown, point on the diagonal (f, f) to $(0, -1)$, that is, a zig-zag path that does not go below the diagonal until it terminates with the final '0'. Fig. 2a shows three of the five paths from $(3, 3)$ to $(0, -1)$ with their code-words including the lexicographically first '1010100' and last '1110000'. The number of paths (Fig. 2b) from row r, column c, to $(0, 0)$ is given by

$$\begin{aligned}
&\text{paths}_{r,c} = 0,\ \text{if } c > r,\\
&\text{paths}_{0,0} = 1,\\
&\text{paths}_{r,0} = 1, r \geq 0,\\
&\text{paths}_{r,c} = \text{paths}_{r-1,c} + \text{paths}_{r,c-1},\ \text{otherwise.}
\end{aligned} \qquad (6)$$

It can be shown that $\text{paths}_{f,f} = C_f$.



(a)

$r\uparrow$									
	$\vdots$								
6	1	6	20	48	90	132	132	0	$\cdots$
5	1	5	14	28	42	42	0	$\cdots$	
4	1	4	9	14	14	0	$\cdots$		
3	1	3	5	5	0	$\cdots$			
2	1	2	2	0	$\cdots$				
1	1	1	0	$\cdots$					
0	1	0	$\cdots$						
	0	1	2	3	4	5	6	$\cdots$	$\rightarrow c$

(b)

Figure 2: (a) Three Dyck paths with corresponding code-words and implied trees. (b) $\text{paths}_{r,c}$.

The code for integers is most easily explained for $N \geq 0$; call this version of the code WTC0. The full binary trees are sorted on their code-word lengths and, for a given length, lexicographically. For a given length, the first code-word is of the form $(10)^f 0$ and the last $1^f 0^{f+1}$ (Fig. 2a). Integer $N \geq 0$ is given the code-word of the N^{th} full binary tree (in the lexicographic order of full binary trees, counting from zero).

Encoding and decoding routines are given in Fig. 3. The code-word for $N = 0$ is "0". For $N > 0$, the C_f integers in the range $cC_{f-1} < N \leq cC_f$, all have code-words of length $2f + 1$. f can be found by searching for the largest cumulative Catalan number, cC_{f-1}, that is less than N. The code-word for N is the K^{th} of the lexicographically ordered code-words of length $2f + 1$ where $K = (N - cC_{f-1})$. The code-word can be found using $\text{paths}_{r,c}$. Starting with an empty code-word at position (r, c) where $r = f, c = f$. If $K > \text{paths}_{r-1,c}$ then append a '1' (move left, $c = c-1$) to the code-word and we need a code-word at least that much further up the rankings ($K = K - \text{paths}_{r-1,c}$). Otherwise, append a '0' (move down, $r = r-1$). Repeat until $r = c = 0$.

The decoding routine (also in Fig. 3) follows similar logic to the encoding routine. A valid code-word, *str*, contains $f = \lfloor |str|/2 \rfloor$ '1's and $f+1$ '0's and no proper prefix contains more '0's than '1's. As *str* is processed from left to right, every '1' (move left) means that *str* is known to be at least $\text{paths}_{r,c}$ further up in rank amongst the code-words of this length. Repeat until the end of the code-word.

It is easy to "shift" the WTC0 code (for integers $N \geq 0$) to instead encode integers $N \geq 1$; call the shifted code WTC1 if we need to distinguish between it and WTC0.

5 Comparing Elias Omega and Wallace Tree code

Table 1 gives examples for integers coded under WTC1 and omega. The lengths of WTC1 code-words increase from time to time always in steps of two. However, lengths in the Elias omega code increase in steps of various sizes, for example increasing by

```
function WTC0enc(N)
 { if( N.isZero() ) return "0";
   var f=cCsearch(N);
       //min f st cC(f)>N
   var K=N-cCatalan(f-1);
   var r=f, c=f, ans="";
   while( r > 0 )
    { var Decr=paths(r-1, c);
      if( K >= Decr )
           { ans=ans+"1"; c-- ;
             K=K-Decr; }
      else { ans=ans+"0"; r-- ; }
    }//while
   return ans + "0";
 }//WTC0enc
```

```
function WTC0dec(str)
 { //assumes str a valid code-word
   if( str == "0" ) return Zero;
   var i, f=Math.floor(str.length/2);
   var r=f, c=f, Ans=cCatalan(f-1);
   for( i=0; i < str.length; i ++ )
      if( str.charAt(i)=="0" ) r--;
      else /* "1" */
       { Ans=Ans+paths(r-1,c);
         c-- ;
       }
   return Ans;
 }//WTC0dec
```

Figure 3: Encoder and decoder for WTC0 for all integers ≥ 0.

Table 1: Example code-words (left) and code-word lengths in bits (right)

N	Elias ω	WTC1
1	1	0
2	010	100
3	011	10100
4	000100	11000
5	000101	1010100
6	000110	1011000
7	000111	1100100
8	0011000	1101000
9	0011001	1110000
10	0011010	101010100

N	|Elias ω|	|WTC1|
10^2	13	13
10^3	17	17
10^4	21	21
10^5	28	25
10^6	31	27
10^7	35	31
10^8	38	35
10^9	41	39
$\vdots$	$\vdots$	$\vdots$
10^{100} (googol)	349	345

four on going from $N = 15$ to $N = 16$ when the value section grows by one and a whole new length section is added; there is no upper limit to the step size and this flows through to the $\log^*$ distribution. Small uniform steps are desirable in inductive inference so that hypotheses of similar probability rank do have similar probabilities.

6 Implied probability distributions

An efficient code for the positive integers $N \geq 1$ implies a probability distribution on them in which $\Pr(N) = \frac{1}{2^{|w(N)|}}$. The Elias omega and Wallace tree (WTC1) codes imply proper probability distributions on the positive integers: Consider an infinite string of bits generated at random, independent and identically distributed (i.i.d.), with $\Pr(\text{`0'}) = \Pr(\text{`1'}) = 0.5$. For each code, the infinite string has some prefix, of length L and probability $\frac{1}{2^L}$, which is a valid code-word in that code. In principle, by

removing the prefix and repeating forever, all possible code-words will be sampled in proportion to their probabilities under the corresponding distribution.

6.1 Elias omega:

An Elias code-word is made up of zero or more length sections followed by one value section. The infinite string of bits starts either '1' or '0'. If it starts '1', that itself is a prefix which is an Elias code-word for the value one, of probability 0.5. If it starts '0' that is decoded as 1 (the lead bit having been changed) which indicates that a section of length $2 = 1 + 1$ follows. If the next section starts '1' it is a value. If it starts '0' it is a length, "00" giving $3 = 2 + 1$ or "01" giving $4 = 3 + 1$. And so on. Eventually a section starting '1', of length s, will appear marking the end of a code-word of length L. There are 2^{s-1} code-words of length L.

6.2 WTC:

Because a one-dimensional random walk ('0' left, '1' right, say) returns to the origin with probability one, the infinite string has some prefix of length $L = (2f + 1), i \geq 0$, that is a valid WTC1 code-word. The probability of the prefix is $\frac{1}{2^{2f+1}}$. There are C_f code-words of length $(2f + 1)$. The total probability of those integers having code-words of length $(2f + 1)$ is $C_f/2^{2f+1}$, $f \geq 0$. The total probability of all positive integers, $\sum_{f \geq 0} C_f/2^{2f+1}$, must be one.

6.3 Comparative code-word lengths

Beyond $N = 2^{15}$ and up to the 506 decimal digit integer corresponding to $cC_{847} + 1$, WTC has the shorter code-words, rarely equalled by the Elias omega code (e.g., for values between $cC_{134} + 1$ and $2^{255} - 1$, both codes using 269 bits). To put these numbers into context, this is past the size of the human genome (3.2×10^9 base-pairs), the estimated number of baryons in the universe (10^{80}) and one googol (10^{100}). Beyond some further point the Elias omega code must take a lead over WTC either permanently or at least most of the time – see section 7.

For each code, integers come in "blocks" that contain integers having code-words of the same length under that code. The sizes of the blocks differ between the codes. For WTC, the blocks are $[1, 1], [2, 2], [3, 4], [5, 9], \ldots, [cC_{f-1} + 1, cC_f], \ldots$ having code-lengths $1, 3, 5, 7, \ldots, 2f + 1, \ldots$ bits respectively. For $f \geq 848$ and up to at least $f = 1000$, $|\text{WTC}(cC_{f-1} + 1)| = |\text{omega}(cC_{f-1} + 1)|$ and $|\text{WTC}(cC_f)| = |\text{omega}(cC_f)| - 2$.

1. The smallest $f > 134$ such that $|\text{WTC}(cC_{f-1} + 1)| = |\text{omega}(cC_{f-1} + 1)|$ is 848. The code-length is 1697 bits.

2. The smallest $f > 134$ such that $|\text{WTC}(cC_{f-1} + 1)| > |\text{omega}(cC_{f-1} + 1)|$ is 3389. The code-lengths are 6779 and 6778 bits, respectively.

3. The smallest $f > 134$ such that $|\text{WTC}(cC_f)| > |\text{omega}(cC_f)|$ is $13,877,006$. The code-lengths are $27,754,013$ and $27,754,012$ bits, respectively. (There are larger f where $|\text{WTC}(cC_f)| \leq |\text{omega}(cC_f)|$.)

7 Asymptotic Analysis of Wallace tree code

It is of interest to determine the asymptotic behaviour of WTC for increasing integer N. Let $L(N)$ denote the length of the binary code-word assigned to integer N by WTC1. Recall from Section 4 that C_f denotes the f^{th} Catalan number, and cC_f denotes the sum of the first f Catalan numbers. Then, the length, in bits, of the code-word assigned by WTC1 to integer N is

$$L(N) = 2\,f(N) + 1 \tag{7}$$

where $f(N) = \inf_f \{\mathbb{Z} : N > cC_f\}$ denotes the smallest integer f such that N exceeds cC_f.

7.1 Bounds on $L(N)$

The following lemma provides appropriate upper and lower bounds for $L(N)$.

Lemma 1. *Let $L(N)$ denote the length of the code-word assigned to integer N by WTC defined by (7). Let*

$$\overline{f}(N) \equiv \overline{f} = \frac{\left(\log N + \frac{3}{2}\log\left(\frac{\log N}{\log 4}\right) + \frac{1}{2}\log\left(\frac{9\pi}{16}\right)\right)}{(1 - 3/2/\log N)\log 4}$$

and

$$\underline{f}(N) \equiv \underline{f} = \frac{\log N}{\log 4 - (3/2/\overline{f})\log \overline{f}}. \tag{8}$$

Then, for all $N \geq 5$, we have

$$2\underline{f}(N) + 1 < L(N) < 2\overline{f}(N) + 1.$$

These bounds allow us to determine the asymptotic behaviour of the length of WTC1 code-words.

Theorem 1.
$$L(N) = \log_2 N + \frac{3}{2}\log_2\log_2 N + \varepsilon_N, \tag{9}$$

$$\text{where} \quad \limsup_{N\to\infty}\{\varepsilon_N\} \leq \frac{3 + \log\left(\frac{9\pi}{32}\right)}{\log 4} < 2.0748, \tag{10}$$

$$\liminf_{N\to\infty}\{\varepsilon_N\} \geq -\frac{1}{2}. \tag{11}$$

The proofs of Lemma 1 and Theorem 1 are available separately[2] Complementing the comparisons in section 6.3, an interesting consequence of Theorem 1 is that there

[2] www.allisons.org/ll/MML/Discrete/Universal/appendicies.pdf ←click

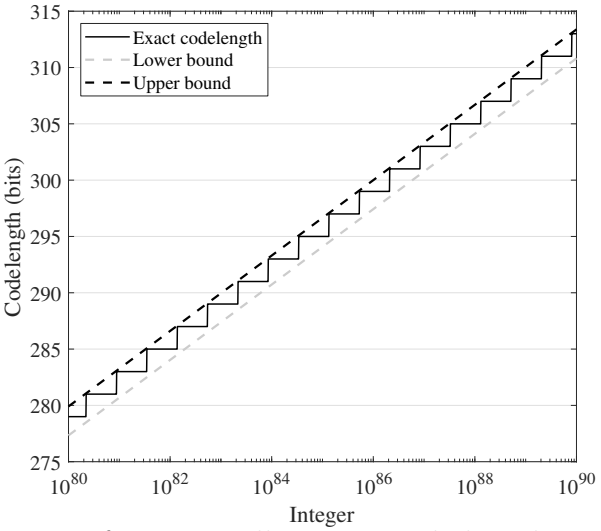


Figure 4: Comparison of exact Wallace-tree code-lengths against upper and lower bounds derived in Lemma 1.

exists some integer M such that $\forall N > M$, $|\mathrm{WTC}(N)| > |\mathrm{omega}(N)|$, although the precise M is unknown and likely inconceivably large.

Theorem 1 can be used to demonstrate both the universality and asymptotic optimality of WTC, building on Elias [1]. To achieve this we first note that the Elias delta code [1], which has asymptotic code-length $\log_2 N + 2\log_2\log_2 N + O(1)$, is both universal and asymptotically optimal. From Theorem 1 it is clear that code-words of WTC are asymptotically shorter than those of Elias delta code. This establishes the universality and asymptotic optimality of WTC.

7.2 Asymptotic code-length formulas for WTC

It is useful to have a simple expression for the code-word lengths of integers under WTC. The requirement for such lengths arises in inductive inference by minimum length encoding. It is common to use Rissanen's $\log^*$ code-word length formula to provide an approximate length for the statement of integer parameters. WTC provides an alternative coding scheme in such settings. Theorem 1 suggests the approximate code-word length formula for WTC as:

$$L(N;c) = \begin{cases} 1 & \text{for } N = 0 \\ 3 & \text{for } N = 1 \\ \log_2 N + \frac{3}{2}\log_2\log_2 N + c & \text{for } N \geq 2 \end{cases} \tag{12}$$

where c is a constant. Possible choices for c are:

- $c = 2$, based on the upper-bound on ε_N in Theorem 1, which ensures $L(N;c)$ is non-decreasing;
- $c = -0.5$, based on the lower-bound on ε_N; or
- $c = 0.75$, which is the average of the two error bounds.

The accuracy of both the bounds given by Lemma 1 and the asymptotic expression (12) is demonstrated in Fig. 4. The figure shows a close correspondence between the asymptotic expression (9) and the exact code-length (7), particularly as N increases.

8 Conclusions

The Wallace tree code (WTC1) for positive integers $N \geq 1$ has shorter code-words than the Elias omega code for most integers up to at least 2.6855×10^{505} (sec.6.3). Code-word length increases in steps of two from time to time as N increases. A formula for the approximate code-word length was derived in section 7.2.

We note that there is a second recursive version of the code, $\mathrm{WTC_r}$. It has the same code-word lengths as WTC but is based on a non-lexicographical ordering of code-words: For code-words of length $2f+1, f > 1$, consider all partitions of $2f$ into j and k such that $f = j + k$. Order code-words of the form '1'$++v++w$, where sub-code-words $|v| = j, |w| = k$ and $j+k = 2f$, on v and within that on w, recursively.

The standard Elias omega code in effect uses a unary code ("0..." $\Rightarrow$ length section, "1..." $\Rightarrow$ final value section) to indicate the number (≥ 1) of sections in a code-word. As defined in section 3, this unary code can be replaced by another code for positive integers and, beyond that even recursively which leads to the omega* code which is more efficient for huge values far beyond those met in everyday data compression.

Importantly, ordering various codes on increasing asymptotic *efficiency* gives: Elias delta, Wallace WTC, Elias omega, omega^2, and omega* (sec.3); all are asymptotically optimal in the sense of Elias [1].

Acknowledgment

The authors would like to thank the late Chris Wallace (1933–2004).

[1] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. IT-21, no. 2, pp. 194–203, 1975.

[2] C. S. Wallace, *Statistical and Inductive Inference by Minimum Message Length*, Springer, 2005, see sections 2.1.4, 2.15 and 2.16, pp.93-100.

[3] A. V. Levenstein, "in Russian," 1968, see D. Salomon, Variable-length Codes for Data Compression, Springer, p.80, 2007.

[4] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[5] L. Allison, *Coding Ockham's Razor*, Springer, 2018.

[6] A. de Segner, "Enumeratio modorum, quibus figurae planae rectilineae per diagonales dividuntur in triangula," *Novi commentarii academiae scientiarum Petropolitanae*, pp. 203–209, 1761, (1758-1759, published 1761).

[7] C. S. Wallace and J. D. Patrick, "Coding decision trees," *Machine Learning*, vol. 11, no. 1, pp. 7–22, 1993.

[8] F. Ruskey and Williams A., "Generating balanced parenthesis strings by prefix shifts," in *CATS, Wollongong*, 2008, pp. 107–115.