Compression and Entropy*

Georges Hansel
Université de Rouen

Dominique Perrin
Université de Paris 7

Imre Simon†
Universidade de São Paulo

Abstract

The connection between text compression and the measure of entropy of a source seems to be well known but poorly documented. We try to partially remedy this situation by showing that the topological entropy is a lower bound for the compression ratio of any compressor. We show that for factorial sources the 1978 version of the Ziv-Lempel compression algorithm achieves this lower bound.

1 Introduction

The main objective of this paper is to elucidate the connection between text compression and the measure of entropy of a source. Such connection seems to be well known but poorly documented. We try to partially remedy this situation by showing that the topological entropy is a lower bound for the compression ratio of any compressor. We also show that for factorial sources the 1978 version of the Ziv-Lempel compression algorithm achieves this lower bound.

More precisely, we model compression as any injective function whose domain and co-domain are subsets of free monoids and define a measure of the compression obtained on any set of words. We model a source through the set of words it can emit, and measure its entropy by the so called topological entropy.

Among other results we show that

(1) the entropy is always a lower bound for the compression,

(2) ranking the words of a source is always a compression which achieves the entropy,

(3) the 1978 version of the Ziv-Lempel algorithm is a universal continuous compression which achieves the entropy on every factorial source,

*Authors addresses: G.H.: Université de Rouen, Mathématiques, BP 67, F-76130 Mont Saint-Aignan, France, <gh@litp.ibp.fr>; D.P.: Université de Paris VII, LITP, 2 place Jussieu, 75251 Paris Cedex 05, France, <dp@litp.ibp.fr>; I.S.: Universidade de São Paulo, IME, Caixa Postal 20570, 01498 São Paulo, SP, Brasil, <is@ime.usp.br>

†This work was done with partial support from FAPESP and from BID/USP (project 30.01).

(4) the factorial hypothesis is necessary in the previous item.

We also address briefly the problem of estimating the topological entropy of a source, both theoretically and practically. Finally, we remark that neither our results nor our arguments depend on any probabilistic concepts and in particular no explicit hypothesis of ergodicity of the source are assumed. On the other hand, an effort was made to make the paper self-contained (apart from some omitted proofs) and to state the concepts and results as simply and precisely as possible.

In the final section we consider the metric case, where both entropy and compression are computed according some invariant probability measure on the free monoid. In this case again entropy is a lower bound for compression.

2 Compression

In this section we define compression and propose measures of the compression rate achieved on a set of words. We note that our measure is geared to the model of a source we shall precise in the next section. Should we use a different model for the source we would need another measure of compression rate to maintain our results; another model will be seen in the last section.

Let A be any finite alphabet and let $\mathbf{2} = \{0, 1\}$ be a fixed binary alphabet. A *compression* or a *compressor* for us is any injective partial function $\gamma\colon A^* \to \mathbf{2}^*$. Note that injectivity is essential in order to keep the ability of unique reconstruction of the original text. Note also that allowing γ to be partial is a concession to achieve better compression rates, since the compressor does not have to worry about possibly irrelevant words. Finally, note that we did not require any algorithmic property of γ. Indeed, we do allow functions which are not even recursive. However, we observe that from a practical point of view the efficiency of a compressor is one of its main virtues; any compressor which does not work in linear time is more or less useless in practice.

A few words now about the alphabet size. Our definition requires a 2-letter output alphabet. This is not a severe restriction, it is done to allow all logarithms in this paper to be taken on base 2 and to allow to normalize the measure of both entropy and compression in "bits per character". Later on we shall see what to modify in order to recover the general case. For the time being we only alert the reader that this convention might pre-empt the semantics of the word compression since it might force compression rates much higher than 1.

We shall need a special class of compressions called ranking which we now define. Consider an alphabet A and fix a total order on it. Corresponding to this order on A we define the *lexicographic order* on A^*, where s precedes t if either s is a prefix of t or there exist words $u, s', t' \in A^*$ and letters $a < b$ in A, such that $s = uas'$ and $t = ubt'$. Now we define the *military order* on A^*: for $s, t \in A^*$ we write $s \leq t$ if either $|s| < |t|$ or if $|s| = |t|$ and s lexicographically precedes t. Clearly, both lexicographic and military orders are total over A^*. Let x be a word in a subset X of A^*. the *rank of x in X* is defined by $r(x, X) = |\{ t \in X \mid t \leq x \}|$. In other words, $r(x, X)$ is the index word x receives when the elements of X are enumerated in military order as $x_1 < x_2 < \cdots$. Given such an enumeration of X we shall say that x_p $(p \geq 1)$ is the p-th word of X in military order.

When $X = \mathbf{2}^*$, using the obvious order on $\mathbf{2}$, we denote $\mu(x) = r(x, \mathbf{2}^*)$, for $x \in \mathbf{2}^*$. We will also need the inverse of μ, denoted $\rho(n) = \mu^{-1}(n)$, for $n > 0$. Thus, if $\mathbb{N}_+$ denotes the set of strictly positive naturals then μ and ρ are bijections,

$$\mu\colon\ \mathbf{2}^* \to \mathbb{N}_+ \quad \text{and} \quad \rho\colon\ \mathbb{N}_+ \to \mathbf{2}^*.$$

Let $n \in \mathbb{N}$ be the canonical interpretation of word $x \in \mathbf{2}^*$. For instance, $n = 5$ for every $x \in 0^*101$. It is easy to see that $\mu(x) = n + 2^{|x|}$ and that $\rho(n)$ is the binary representation of $n - 2^{\lfloor \log n \rfloor}$ padded with zeros on the left to give a word of length $\lfloor \log n \rfloor$.

Let X be an infinite subset of A^*. Then $\gamma_X(x) = \rho(r(x, X))$ is a compression, called the *ranking of* X. Another way of looking at γ_X is by noting that it is the unique partial function

$$\gamma_X\colon\ A^* \to \mathbf{2}^*$$

with domain X which is monotonic with respect to military order and whose restriction to X is a bijection from X to $\mathbf{2}^*$. Note that $\gamma_{\mathbf{2}^*}$ is just the identity function on $\mathbf{2}^*$.

Now we measure the compression rate achieved by a given compressor $\gamma\colon\ A^* \to \mathbf{2}^*$. For $x \in A^*$ the *compression of* x is $\tau(x) = |\gamma(x)|/|x|$. For an infinite set $X \subseteq A^*$ some words might be compressed more than others. For $p > 0$, let $x_p \in X$ be the p-th word of X in military order. Assuming that $X \subseteq \operatorname{dom}\gamma$ we define the *compression of* γ *on* X by

$$\tau(X) = \limsup_{p\to\infty} \tau(x_p).$$

This is an asymptotic worst case measure of the compression. A more realistic measure is the (asymptotic worst case) *average compression rate* of γ on X:

$$\tau_{ave}(X) = \limsup_{n\to\infty} \frac{\sum_{x\in X\cap A^n} \tau(x)}{|X \cap A^n|}.$$

where we interpret $0/0$ as being 0. The measure τ always refers to a fixed compression γ; from the context it will always be clear to which γ does τ refer to. The two measures introduced are related by the next Proposition whose proof we omit.

Proposition 1 *For every infinite subset X of A^* and for every compression $\gamma\colon\ A^* \to \mathbf{2}^*$, such that $X \subseteq \operatorname{dom}\gamma$, we have that $\tau_{ave}(X) \le \tau(X)$. There are infinite sets X for which the above inequality is strict.*

We shall say that compressor γ is *uniform* on X if equality holds in the above Proposition.

2.1 The Ziv-Lempel Compression Algorithm

In this section we specify and study some properties of a truly remarkable algorithm, namely the 1978 version of the Ziv-Lempel compression algorithm [18], which we will call ZL. This is one of the most succesfull algorithms in the area, both in theory and in practice; its numerous variants constitute perhaps the most popular compression method in use. In particular the UNIX[1] utility `compress` is based on a variant of this algorithm. More details on this family of compressors can be found in [17, 2].

[1]UNIX is a trademark of AT&T Bell Laboratories.

In the sequel we describe the Ziv-Lempel algorithm, i.e. the total function $\mathrm{ZL}\colon A^* \to \mathbf{2}^*$. Let $x \in A^*$ be a word to be compressed. Word x is factorized as $x = x_0 x_1 x_2 \dots x_m x_{m+1}$ for some m subject to the following three conditions:

(1) x_0 is empty,

(2) for each i, $1 \le i \le m$, x_i is the longest prefix of $x_i x_{i+1} \dots x_{m+1}$ which can be written as $x_i = x_{j_i} a_i$, for some $j_i < i$ and $a_i \in A$,

(3) $x_{m+1} = x_{j_{m+1}}$, for some $j_{m+1} < m+1$.

In other words, the first factor is empty and each other factor but the last one is a new factor which consists of a previous factor followed by one letter. Clearly such a factorization is unique.

The words x_i, $1 \le i \le m$ are entered in a digital tree (or trie) which we will denote by $T(x)$ and this tree is usually called the dictionary. Each factor x_i corresponds to a vertex of $T(x)$ and if we label this vertex with label i then it is easy to see that the correspondence $x \mapsto (T(x), j_{m+1})$ is injective.

Now we describe the output of ZL. Factor x_i, $1 \le i \le m$, is transmitted as the pair (j_i, a_i). Factor x_{m+1} is transmitted just as j_{m+1}. To transmit j_i one usually uses $\lceil \log i \rceil$ bits which are necessary and sufficient to specify any of the numbers $0, 1, 2, \dots i-1$. The character a_i can be transmitted by itself and since A is fixed it takes some bounded number of bits to transmit it.

Next we note that the length of the output depends exclusively on the number m of edges of $T(x)$. Indeed, we have

$$|\,\mathrm{ZL}(x)| = \sum_{1}^{m+1} \lceil \log i \rceil + K'' m,$$

for an apropriate constant K''. Since $\lceil \log i \rceil \le 1 + \log m$ we can put

$$|\,\mathrm{ZL}(x)| \le m \log m + Km,$$

for some constant K depending on A. The reason behind the last inequality is that whatever is the shape and labeling of $T(x)$ the output always consists of $m+1$ numbers of predetermined length (at most $\lceil \log m \rceil$) and m characters from A. Returning to the constant K we note that it might even be negative. We also observe that it will usually depend on the details of the protocol used to form the output and that clever protocols might lead to smaller constants. While this will change the length of the output it does not change $\tau(X)$ because the term $m \log m$ is dominant in the sum.

Let us now analyse the length of the input x in function of the tree $T(x)$. For vertex i the contribution of factor x_i to $|x|$ is exactly the height of vertex i, denoted height(i). Thus,

$$|x| = \mathrm{height}(j_{m+1}) + \sum_{i=1}^{m} \mathrm{height}(i).$$

The dominant term $\sum_{i=1}^{m} \mathrm{height}(i)$ depends only on the shape of the tree $T(x)$ and it is usually called *internal path length* of $T(x)$; see for instance [8].

```
   T(y)                          T(x)

    00                            00
    /                           /      \
   01                       01          03
  / \                      /  \        /  \
 03 02                   04    02    08    06
   /  \                  /\     \    /     /\
 04    06               09 07    05 11    12 10
 /\     \
07 05    08
 \        \
  09        10                 A = { a, b }
   \
    11
    /                           a /    \ b
   12
```

Figure 1: Trees $T(x)$ and $T(y)$.

From the above discussion one can conclude that if we fix m and consider any set of words x factored with m factors by ZL then the compression rate $\tau(x)$ is essentially inversely proportional to the internal path length of $T(x)$. In other words, if $T(x)$ is short and fat then τ is large and ZL is inefficient but if $T(x)$ is tall and slim then τ is small and ZL is efficient.

We illustrate this section with two examples. We consider $A = \{ a, b \}$, and

$$x = a.ab.b.aa.abb.bb.aab.ba.aaa.bbb.baa.bba,$$

$$y = a.ab.aa.aba.abab.abb.abaa.abbb.abaab.abbbb.abaabb.abaabba, abaabba$$

$$\mathrm{ZL}(x) = a.1b.00b.01a.010b.011b.100b.011a.0100a.0110b.1000a.0110a.0000,$$

$$\mathrm{ZL}(y) = a.1b.01a.10a.100b.010b.100a.110b.0111b.1000b.1001b.1011a.1100,$$

$$m = 12, |x| = 28, |y| = 53, |\,\mathrm{ZL}(x)| = |\,\mathrm{ZL}(y)| = 49,$$

$$\tau(x) = 1.75, \tau(y) = 0.92.$$

On Figure 1 we represent the trees $T(x)$ and $T(y)$.

Observe that ZL is a continuous compression. By this we mean that function ZL is continuous with respect to the usual "longest common prefix" distance in A^*.

Observe also that to uncompress $\mathrm{ZL}(x)$ one has to reconstruct the tree $T(x)$ exactly as it was constructed during the compression and the factors x_i are obtained from this tree. The output protocol must be defined in order to allow this synchronized reconstruction of the tree $T(x)$.

For future reference we resume the properties of the Ziv-Lempel compression algorithm.

Proposition 2 *The Ziv-Lempel algorithm* ZL: $A^* \to \mathbf{2}^*$ *is a continuous compression. Let $x \in A^*$ be such that the tree $T(x)$ has m edges. Let I be the internal path length of $T(x)$ and let τ be the compression of* ZL. *Then, for some constant K, $\tau(x) \leq (m \log m + Km)/I$.*

3 Topological Entropy

Entropy is one of these words used for a number of related concepts in Mathematics and Physics. Usually it is expressed by a formula which through some magic mechanism leads one to unexpected properties. It seems that the word entropy was first used by R. Clausius in 1864 in a thermodynamical context. The classical version of the formula and the use of the letter H to denote entropy goes back to Boltzmann who used it in 1890 in the context of Statistical Mechanics:

$$h(S) = -\sum_{i=1}^{n} p_i \log p_i, \text{ with } \sum_{i=1}^{n} p_i = 1.$$

The modern use of entropy was pioneered by Claude Shannon in 1948 when he launched Information Theory [16]. Intuitively speaking Shannon showed that the above formula can be used to measure the information content, in bits per symbol, of a source which independently emits symbols i (over a fixed alphabet $\{1, 2, \ldots, n\}$) with probability p_i. Such a source is just a Bernoulli distribution. It is said that von Neumann suggested the term *entropy* to Shannon because of its former use by Boltzmann and because "no one knows what entropy really is, so in a debate you will always have the advantage" (Tribus and McIrvine, Energy and information, Scientific American, 1971). For more details on the misteries of entropy we refer the reader to the book of Rényi [15].

Shannon himself extended the formula to measure the entropy of sources modeled by finite Markov chains; we refer the reader to the books on Information Theory and especially to the little book of Khinchin [7] where we found the first explicit mention of the relationship between compression and entropy.

Perhaps the deepest use of entropy is encountered in modern Ergodic Theory, where entropy is one of the most successful invariants introduced by Kolmogorov and Sinai in 1958 and 1959. It is noteworthy to observe that even though entropy occupies a significant portion of any modern book on Ergodic Theory neither the concept nor the word occur in the classical book "Lectures on Ergodic Theory" of Paul Halmos which appeared in 1953. For a modern survey on entropy we refer the reader to the book of Martin and England [11].

Our interest here is to explore the relationship between entropy and compression. For the sources considered by Shannon such a relationship seems to be implicit in his work. For stationary ergodic sources the relationship has been known for some time but until recently it was not well documented. See the work of Hansel [6] and also the recent papers by Ornstein, Weiss, and Shields [13, 14]. From our point of view there are some inconveniences with the approaches above: they are either too simplistic (the Shannon source) or too sophisticated (the stationary ergodic source) to be of easy use in Computer Science. This fact led us to the investigations reported in the sequel. Note that on the one hand our approach can be used to model a great variety of sources and on the other it does not depend on any probabilistic considerations (except some of the material in the last section).

The key concept we use is the topological entropy of a source. A *source* is any infinite subset X of A^*, the words in X constitute the messages which can be emitted by the source. The *topological entropy* of source X is defined by

$$H(X) = \limsup_{n\to\infty} \frac{1}{n} \log s_n,$$

where $s_n = |X \cap A^n|$. Here, by convention, $\log 0 = 0$. As far as we know the first explicit use of this formula goes back to Shannon who used it to compute the channel capacity. This motivated further investigations including its computation for recognizable and context free languages by Chomsky and Miller, de Luca, Kuich and others [3, 4, 9]. Later, Adler, Konheim and McAndrew introduced its modern use in 1965 [1] in Ergodic Theory which was recently surveyed by Newhouse [12]. We shall explore the relationship of this concept to text compression.

Intuitively speaking the topological entropy is a measure, in bits per character, of the asymptotic exponential growth rate of the numbers s_n; it is a collective property of X and not of any individual word $s \in A^*$. More precisely, assume we plot $\log s_n$ against n. Then, the entropy of X is the greatest angular coefficient of the lines through the origin which can be sustained infinitely often by the points of our graph.

We indicate an alternate view now. A subset X of A^* is *factorial* if every factor of every word in X also belongs to X.

Proposition 3 *Let $X \subseteq A^*$ be infinite and assume that $\lim_{n\to\infty} \log(s_{n+1}/s_n)$ exists. Then its value is exactly $H(X)$. The above limit might not exist even if we assume that X is factorial.*

If X is factorial then we can interpret s_{n+1}/s_n as being the average branching factor of X at words of length n (imagine the digital tree containing the words in X). Thus, for such a well behaved X, H is the asymptotic value of the number of bits to write the average branching factor of X. Another related view is obtained if we note that $(1/n)\log s_n = \log \sqrt[n]{s_n}$. Thus, the term $(1/n)\log s_n$ can be thought of as being the hypothetical branching factor at level n one would get if branching was uniform between levels 1 and n.

Next we compute $H(X)$ for some sources. If $X = A^*$ then $H(X) = \log |A|$. Now, clearly, if $Y \subseteq X$ then $H(Y) \leq H(X)$, hence $\log |A|$ is an upper bound for the entropy of any subset of A^*.

Now we give two examples of null entropy. Let $X = \{\, a^n b^n \mid n \geq 0 \,\}$, then $H(X) = 0$. Let now $A = \{\, a, b \,\}$ and let X be the set of factors of $\{\, \varphi^p(a) \mid p \geq 1 \,\}$, where φ is a morphism defined by $\varphi(a) = ab$ and $\varphi(b) = ba$. Thus, X is just the set of factors of the infinite Thue-Morse word (see Lothaire [10]). Then the reader can show that $H(X) = 0$. An interesting experience supporting this fact (and also the results in the next section) is to apply the UNIX `compress` to the words $\varphi^n(a)$ and observe the astonishing compression obtained which should be contrasted with the much poorer results obtained by `pack` or `compact` which use different algorithms (Huffman code and adaptive Huffman code respectively).

Finally, we leave it to the reader to show that for $X = \{\, a, b \,\}^* c \{\, a, b, d \,\}^*$, $H(X) = \log 3$; see also Section 5.

Regarding the breadth of our model we note that Shannon's First Theorem for the noiseless discrete channel states that the topological entropy H (there called channel capacity) of the set of most probable words is just $-\sum p_i \log p_i$. Hence, the Shannon source can be viewed as a particular case of the model we focus on.

4 Compression and Entropy

In this section we state our main results and prove the most important ones. Note that the proofs are based both on counting arguments and on approximations of the quantities in question.

We begin with two simple facts.

Lemma 4 *The length of the n-th word in $\mathbf{2}^*$ is $\lfloor \log n \rfloor$; i.e. $|\rho(n)| = \lfloor \log n \rfloor$.*

Lemma 5 *Let $g(n)$, for $n \geq 1$, denote the sum of lengths of the first n words of $\mathbf{2}^*$ in military order. Then,*

$$g(n) = \sum_{i=1}^{n} |\rho(i)| = (n+1)m - 2^{m+1} + 2,$$

where $m = \lfloor \log n \rfloor$. In particular,

$$n \log n - 3n \leq g(n) \leq \sum_{i=1}^{n} \log i \leq n \log n.$$

The next result shows in which sense is entropy a lower bound for compression.

Proposition 6 *For every compression $\gamma \colon A^* \to \mathbf{2}^*$ and for every infinite $X \subseteq A^*$, if $X \subseteq \operatorname{dom} \gamma$ then $H(X) \leq \tau_{ave}(X)$.*

Proof. Let $R \subseteq R' \subseteq B^*$ and, for some $n \geq 1$ let r_n and r'_n be the n-th word of R and R' in military order. Clearly, $|r_n| \geq |r'_n|$. Let then $R = \{\, \gamma(x) \mid x \in X \cap A^n \,\}$ and $R' = \mathbf{2}^*$. Since $X \subseteq \operatorname{dom} \gamma$ and γ is injective it follows that $|R| = s_n$, where $s_n = |X \cap A^n|$. Thus, in view of Lemma 5 it follows that

$$\sum_{x \in X \cap A^n} |\gamma(x)| \geq g(s_n) \geq s_n \log s_n - 3 s_n.$$

Dividing by $n s_n$ we get

$$\frac{\sum_{x \in X \cap A^n} |\gamma(x)|}{n s_n} \geq \frac{\log s_n}{n} - \frac{3}{n}.$$

Hence, taking the lim sup of the sequence and observing that the term $-3/n$ tends to zero we have

$$\tau_{ave}(X) = \limsup_{n \to \infty} \frac{\sum_{x \in X \cap A^n} \tau(x)}{|X \cap A^n|} = \limsup_{n \to \infty} \frac{\sum_{x \in X \cap A^n} |\gamma(x)|}{n s_n} \geq$$

$$\limsup_{n \to \infty} \frac{\log s_n}{n} - \frac{3}{n} = H(X),$$

as required. ∎

Now we show that ranking is a uniform compression which is optimal for every source X. Initially we state a technical Lemma without proof. For a sequence $(s_n)_{n>0}$ let H_s denote its topological entropy, i.e., $H_s = \limsup_{n\to\infty}(1/n) log s_n$.

Lemma 7 *Let $(s_n)_{n>0}$ be a sequence of real numbers such that, for some constant $k \geq 1$, either $s_n = 0$ or $1 \leq s_n \leq k^n$. Let $t_n = \sum_{i=1}^n s_i$. Then, $H_s = H_t$.*

The reader can verify that Lemma 7 is equivalent to stating that for every infinite $X \subseteq A^*$ and every $a \in A$, $H(X) = H(Xa^*)$.

Proposition 8 *For every infinite $X \subseteq A^*$, the compression rate $\tau(X)$ of γ_X satisfies $\tau(X) \leq H(X)$. Hence, γ_X is always a uniform optimal compression.*

Proof. Let $s_n = |X \cap A^n|$ and $t_n = \sum_{i=1}^n s_i$. For $p \geq 1$ let x_p be the p-th word of X in military order and assume that $|x_p| = n$. Then, $p \leq t_n$. Since γ_X is the ranking of X, in view of Lemma 4, we have that $\tau(x_p) = |\gamma_X(x_p)|/n \leq \lfloor \log t_n \rfloor/n$. It follows that the sequence $(\tau(x_p))_{p\geq1}$ is majorized, term by term, by the sequence which consists of s_n terms equal to $(\log t_n)/n$, for $n = 1, 2, \ldots$ successively. Using Lemma 7 it follows that

$$\tau(X) = \limsup_{p\to\infty} \tau(x_p) \leq \limsup_{n\to\infty} \frac{\log t_n}{n} = \limsup_{n\to\infty} \frac{\log s_n}{n} = H(X).$$

This concludes the proof of Proposition 8. ∎

This results allows an alternative interpretation of entropy, a variant of which will play an important role in the sequel.

Corollary 9 *Let $X \subseteq A^*$ be an infinite set and let x_p denote the p-th word of X in military order. Then,*

$$H(X) = \limsup_{p\to\infty} \frac{\log p}{|x_p|}.$$

As noted earlier, usually γ_X is not continuous, though one can construct examples of sets where γ_X is a continuous compression. Does there exist continuous optimal compressions for every X? In the sequel we show that the Ziv-Lempel algorithm satisfies this requirement for certain sets X. First we need the promised variant of Corollary 9.

Proposition 10 *Let $X \subseteq A^*$ be an infinite set and let x_p denote the p-th word of X in military order, for $p \geq 1$. Then,*

$$H(X) = \limsup_{m\to\infty} \frac{\sum_{p=1}^m \log p}{\sum_{p=1}^m |x_p|}.$$

Proof. Let $z_m = \sum_{p=1}^m \log p / \sum_{p=1}^m |x_p|$, and let $H = H(X)$. Using Lemma 5 we have that $\sum_{p=1}^m \log p \geq m \log m - 3m$. Since $|x_p|$ is monotonic increasing with p we have that $\sum_{p=1}^m |x_p| \leq m|x_m|$. Then,

$$z_m \geq \frac{m \log m - 3m}{m|x_m|} = \frac{\log m}{|x_m|} - \frac{3}{|x_m|}.$$

Since, by Corollary 9, $H = \limsup(\log m)/|x_m|$ we conclude that,

$$\limsup_{m\to\infty} z_m \geq \limsup_{m\to\infty}(\frac{\log m}{|x_m|} - \frac{3}{|x_m|}) = \limsup_{m\to\infty} \frac{\log m}{|x_m|} = H.$$

In the next paragraph we will show that

$$\forall \epsilon > 0 \exists m_0 \forall m \geq m_0 : \quad z_m \leq H + \epsilon.$$

It will then follow that $\limsup z_m \leq H$, as required.

Since, by Corollary 9, $H = \limsup(\log p)/|x_p|$, we can conclude that, for $\delta = \epsilon/2$,

$$\exists p_0 \forall p \geq p_0 : \quad \frac{\log p}{|x_p|} \leq H + \delta.$$

Hence, $|x_p| \geq (\log p)/(H + \delta)$. Choose m_0 such that

$$\sum_{p=1}^{m_0} \log p \geq (2 + H/\delta)[\sum_{p=1}^{p_0} \log p - (H + \delta)\sum_{p=1}^{p_0} |x_p|].$$

Such an m_0 clearly exists since the left hand side tends to infinity while the right hand side is a constant. Let now $m \geq m_0$. We have,

$$z_m = \frac{\sum_{p=1}^{m} \log p}{\sum_{p=1}^{m} |x_p|} = \frac{\sum_{p=1}^{m} \log p}{\sum_{p=1}^{p_0} |x_p| + \sum_{p=p_0+1}^{m} |x_p|} \leq$$

$$\frac{\sum_{p=1}^{m} \log p}{\sum_{p=1}^{p_0} |x_p| + 1/(H+\delta)\sum_{p=p_0+1}^{m} \log p} = \frac{(H+\delta)\sum_{p=1}^{m} \log p}{(H+\delta)\sum_{p=1}^{p_0} |x_p| + \sum_{p=p_0+1}^{m} \log p} =$$

$$\frac{(H+\delta)\sum_{p=1}^{m} \log p}{(H+\delta)\sum_{p=1}^{p_0} |x_p| - \sum_{p=1}^{p_0} \log p + \sum_{p=1}^{m} \log p} = \frac{H+\delta}{1-Q},$$

where

$$Q = \frac{\sum_{p=1}^{p_0} \log p - (H+\delta)\sum_{p=1}^{p_0} |x_p|}{\sum_{p=1}^{m} \log p}.$$

Using now the choice of m_0 and the fact that $m \geq m_0$ we have that

$$\sum_{p=1}^{m} \log p \geq \sum_{p=1}^{m_0} \log p \geq (2 + H/\delta)[\sum_{p=1}^{p_0} \log p - (H + \delta)\sum_{p=1}^{p_0} |x_p|];$$

hence,

$$Q \leq \frac{\delta}{H + 2\delta},$$

and, consequently,

$$\frac{H+\delta}{1-Q} \leq \frac{H+\delta}{1 - \delta/(H+2\delta)} = H + \epsilon.$$

Altogether, $z_m \leq H + \epsilon$, as claimed. This concludes the proof of Proposition 10. ∎

Finally we have the main result of this paper.

Theorem 11 *Let X be an infinite factorial subset of A^* and let τ be the compression of the Ziv-Lempel algorithm,* ZL: $A^* \to \mathbf{2}^*$. *Then, $\tau(X) \leq H(X)$, i.e.* ZL *is a universal continuous compression which is optimal on every factorial source X.*

Proof. For $n \geq 1$, let x_n be the n-th word of X in military order. Let us consider the tree $T(x_n)$ and let m_n and I_n denote its number of edges and its internal length. Since X is factorial, every one of the m_n words contributing to I_n lies in X; hence

$$I_n \geq \sum_{p=1}^{m_n} |x_p|.$$

Using then Proposition 2 and Lemma 5

$$\tau(x_n) \leq \frac{m_n \log m_n + K m_n}{I_n} \leq \frac{\sum_{p=1}^{m_n} \log p + K' m_n}{\sum_{p=1}^{m_n} |x_p|},$$

for appropriate constants K and K'. Since $\sum \log p$ is the dominant term in the numerator we have

$$\tau(X) = \limsup_{n\to\infty} \tau(x_n) \leq \limsup_{n\to\infty} \frac{\sum_{p=1}^{m_n} \log p}{\sum_{p=1}^{m_n} |x_p|}.$$

Since there are only finitely many trees $T(x)$ with any given number m of edges, we have the fiirst equality below. The second one comes from Proposition 10.

$$\limsup_{n\to\infty} \frac{\sum_{p=1}^{m_n} \log p}{\sum_{p=1}^{m_n} |x_p|} = \limsup_{m\to\infty} \frac{\sum_{p=1}^{m} \log p}{\sum_{p=1}^{m} |x_p|} = H(X).$$

Altogether, $\tau(X) \leq H(X)$; concluding the proof of Theorem 11. ∎

Let A be a fixed set now. Consider, for every $n \geq 1$, the word x_n of length n such that $\tau(x_n) = \max\{\, \tau(x) \mid x \in A^n \,\}$. The reader can verify that X is an infinite set of entropy $H(X) = 0$, for which $\tau(X) = \log |A|$. Of course if Y is the set of factors of X then $H(Y) = \tau(Y) = \log |A|$. It follows that Theorem 11 does not hold for arbitrary infinite subsets X of A^*.

5 Some Remarks

Throughout the paper we considered a binary output alphabet but now we wish to argue that the results do not depend on this particularity. Let B be an output alphabet, with $|B| = k$, and let $\gamma\colon A^* \to B^*$ be a compression. In order to maintain our results, we would have to define the compression coefficient as

$$\tau(x) = \frac{|\gamma(x)|}{|x|} \log k.$$

This is necessary in order to normalize the measure of the compression rate in "bits per character", and keep it compatible with the entropy.

An equivalent solution, perhaps even preferable, would be to change all the logarithms to base k and continue to use $\tau(x) = |\gamma(x)|/|x|$. This corresponds to measure entropy and compression in k-ary digits per character. This is preferable because the compression coefficients will be less than one exactly when there is a compression in the ordinary sense of the word.

We leave it as an interesting exercise to verify that all the results are indeed maintained after the obvious changes are made.

Another question relates to expansions. Let $\gamma\colon A^* \to A^*$ be a compression with identical input and output alphabets. If γ is a total function then clearly $|\gamma(x)| < |x|$ for some x if and only if $|\gamma(y)| > |y|$ for some y. In other words, every total proper compression must expand some words. Can one limit the amount of expansion? The answer is "yes" because we can use the first output character to show whether or not there is compression and in case the compression would expand we just transmit the input. With this protocol one can see that expansion can be limited to one character, without compromising the compression rate. We make two observations. First, something similar happens in real life. Indeed, the UNIX utility `compress` refuses to compress a file if it would expand. The corresponding bit indicating whether or not compression was performed is put in the name of the file which will or not have the extension `".Z"`. In second place we note that limiting expansions in the above way destroys continuity. Is it possible to limit expansion while maintaining continuity? This might lead to interesting questions. In particular one can verify that ZL uses unlimited expansion and thus it does not settle the question.

Now we make some remarks about the estimation of the topological entropy. A consequence of the proof of Theorem 11 is that if one can enumerate in military order the words of a set X then the compression rate $\tau(x_1x_2\dots x_m)$ of the concatenation of the first m words of X will give a numerical estimation of the entropy of X. Here, τ refers to the Ziv-Lempel algorithm and we have to assume that X is factorial. Of course $H(X)$ is the $\limsup$ of the above numbers and some preliminary experiments indicate that convergence might be (very) slow. Nevertheless, one has a method of estimating the entropy of X.

Another way of estimating the entropy would be to consider the values of $(\log p)/|x_p|$. According Corollary 9 the $\limsup$ of such a sequence is always $H(X)$. This again assumes that we can compute $|x_p|$ for a given p.

The entropy of a rational language can be easily esimated using classical methods of numerical computation instead of performing tests on certain sequences as in the general case. Let indeed X be a rational language and let A be a deterministic (or, more generally, unambiguous) automaton recognizing X. We assume A to be trim. Let M be the adjacency matrix of the graph of the automaton. Then it is a classical result that

$$H(X) = \log \rho,$$

where ρ is the maximal eigenenvalue of M, also called its Perron eigenvalue. The computation of $H(X)$ therefore reduces to the computation of the maximal eigenvalue of a positive matrix. This is a well-known computational problem since ρ is also the maximal root of the characteristic polynomial $p(\lambda)$ of the matrix M.

In practice, one may also use the following more direct and elementary method. Let M be an $n \times n$ positive matrix. We assume M to be irreducible and aperiodic. Let x be a vector with all its components strictly positive and let $R_{max}(x) = \max\{\, (xM)_i/x_i \mid 1 \le i \le n\,\}$ and $R_{min}(x) = \min\{\, (xM)_i/x_i \mid 1 \le i \le n\,\}$. It is well known that for any such vector x one has the inequalities $R_{min}(x) \le \rho \le R_{max}(x)$, hence giving an estimate of the Perron eigenvalue of M. This estimate is especially visible in the case where x is the vector with all its components equal to one. In this case, R_{min} (resp. R_{max}) is the minimum (resp. the maximum) of the indegrees of the vertices. Also, the estimate can be made as precise as necessary since for any such vector x, the sequence starting at $x_0 = x$ and defined by $x_{n+1} = x_n M$ is such that R_{max} and R_{min} converge to ρ.

Ranking functions have received some attention recently, both as a way of compression and also from the computational point of view. See, for instance [5] and the references therein, where another theory of compression is delineated.

We close this paper by considering very briefly a probabilistic model of a source and indicate that in this case also entropy is a lower bound for compression. The results in this case however are not as complete as for the topological model of a source.

We consider an invariant probability measure ν on A^*. By this we mean a function $\nu\colon A^* \to [0,1]$ such that for every $u \in A^*$ one has

$$\nu(u) = \sum_{a \in A} \nu(au) = \sum_{a \in A} \nu(ua)$$

and such that $\nu(1) = 1$. The (metric) entropy of ν is given by

$$h(\nu) = \lim_{n\to\infty} \frac{1}{n} \sum_{u \in A^n} -\nu(u) \log \nu(u).$$

Let X be the support of ν, i.e. $X = \{\, u \in A^* \mid \nu(u) > 0 \,\}$. Then $h(\nu) \le H(X)$, since one can verify that

$$-\sum_{u \in A^n} \nu(u) \log \nu(u) \le \log |X \cap A^n|.$$

Let γ be a total compression. We define the metric compression rate $\tau(\nu)$ of γ on ν by

$$\tau(\nu) = \limsup_{n\to\infty} \sum_{u \in A^n} \nu(u) \frac{|\gamma(u)|}{|u|}.$$

By exploiting the convexity of the logarithmic function one can prove the following result.

Proposition 12 *For every ν and γ as above, $h(\nu) \le \tau(\nu)$.*

References

[1] R. Adler, A. Konheim, and M. McAndrew. Topological entropy. *Trans. Amer. Math. Soc.*, 114:309–319, 1965.

[2] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression*. Prentice Hall, Englewood Cliffs, N.J., 1990.

[3] N. Chomsky and G. A. Miller. Finite state languages. *Information and Control*, 1:91–112, 1958.

[4] A. de Luca. On the entropy of a formal language. In H. Brakhage, editor, *Automata Theory and Formal Languages*, pages 103–109, Springer-Verlag, Berlin, 1975. Lecture Notes in Computer Science, 33.

[5] A. V. Goldberg and M. Sipser. Compression and ranking. *SIAM J. Comput.*, 20:524–536, 1991.

[6] G. Hansel. Estimation of the entropy by the Lempel-Ziv method. In M. Gross and D. Perrin, editors, *Electronic Dictionaries and Automata in Computational Linguistics*, pages 51–65, Springer-Verlag, Berlin, 1989. Lecture Notes in Computer Science, 377.

[7] A. I. Khinchin. *Mathematical Foundations of Information Theory.* Dover, New York, 1957.

[8] D. E. Knuth. *The Art of Computer Programming, Vol. 1, Fundamental Algorithms.* Addison-Wesley Pu. Co., Reading, MA, 1968.

[9] W. Kuich. On the entropy of context-free languages. *Information and Control*, 16:173–200, 1970.

[10] M. Lothaire. *Combinatorics on Words.* Volume 17 of *Encyclopedia of Mathematics and its Applications*, Addison-Wesley Pu. Co., Reading, MA, 1983.

[11] N. F. G. Martin and J. W. England. *Mathematical Theory of Entropy.* Volume 12 of *Encyclopedia of Mathematics and its Applications*, Addison-Wesley Pu. Co., Reading, MA, 1981.

[12] S. E. Newhouse. Entropy in smooth dynamical systems. 1990. To appear in *Proceedings of the 1990 World Congress of Mathematicians.*

[13] D. Ornstein and B. Weiss. Entropy and data compression schemes. 1990. manuscript.

[14] D. S. Ornstein and P. C. Shields. Universal almost sure data compression. *The Annals of Probability*, 18:441–452, 1990.

[15] A. Rényi. *A Diary on Information Theory.* John Wiley & Sons, New York, NY, 1984.

[16] C. E. Shannon. A mathematical theory of communication. *Bell System Technical J.*, 27:398–403, 1948.

[17] J. A. Storer. *Data Compression - Methods and Theory.* Computer Science Press, Rockville,MD, 1988.

[18] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Information Theory*, IT-24:530–536, 1978.