Department of Computer Science Technical Reports

Department of Computer Science

1995

On the Average Redundancy Rate of the Lempel-Ziv Code

Guy Louchard

Wojciech Szpankowski
Purdue University, spa@cs.purdue.edu

Report Number:
95-049

On the Average Redundancy Rate of the Lempel-Ziv Code

Guy Louchard
Laboratoire d'Informatique Théorique
Universitè Libre de Bruxelles
B-1050 Brussels, Belgium
Wojciech Szpankowski
Department of Computer Science
Purdue University
West Lafayette, IN 47907

CSD-TR-95-049
July, 1995

ON THE AVERAGE REDUNDANCY RATE OF THE LEMPEL-ZIV CODE

July 14, 1995

Guy Louchard
Laboratoire d'Informatique Théorique
Université Libre de Bruxelles
B-1050 Brussels
Belgium

Wojciech Szpankowski*
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.

Abstract

It was conjectured that the average redundancy rate, r_n, for the Lempel-Ziv code (LZ78) is $\Theta(\log\log n/\log n)$ where n is the length of the database sequence. However, it was also known that for infinitely many n the redundancy r_n is bounded from the below by $2/\log n$. In this paper we settle the above conjecture in the negative by proving that for memoryless and Markov sources the average redundancy rate attains asymptotically $Er_n = (A+\delta(n))/\log n + O(\log\log n/\log^2 n)$ where A is an explicitly given constant that depends on the source characteristics, and $\delta(x)$ is a fluctuating function. This result is a consequence of recently established second-order properties for the number of phrases in the Lempel-Ziv algorithm. We also derive the leading term for the kth moment of the number of phrases. Finally, in concluding remarks we discuss generalized Lempel-Ziv codes for which the average redundancy rates are computed and compared with the original Lempel-Ziv codes.

Index Terms: Data compression, Lempel-Ziv parsing scheme, generalized Lempel-Ziv scheme, average redundancy rate, digital search trees, suffix trees.

*This research was partially supported by NSF Grants NCR-9206315 and CCR-9201078, and NATO Collaborative Grant CRG.950060.

1. INTRODUCTION

The *redundancy* of a noiseless code measures how far the code is from being optimal for a given source of information. While asymptotically optimal codes require that the redundancy tends to zero (with the length of the code), sometimes a stronger requirement is necessary: Namely, that the (average) redundancy per symbol goes to zero at some universal rate. For example, while there are several asymptotically optimal data compression codes (e.g., several versions of Lempel-Ziv scheme [23, 25]), one can further optimize the rate of convergence to the optimal compression ratio. This is of prime importance for some practical on-line and off-line data compression schemes.

It is known that some prefix codes exist for which the expected redundancy per symbol is $O(\log n/n)$ for a class of sources (e.g., Markov, finite-state sources, etc.). But, recently Shields [17] proved that such a redundancy rate cannot be achieved for general sources. It must be further observed that often for practical universal data compression codes the above redundancy rate is not achievable.

In this paper, we investigate the redundancy rate of the Lempel-Ziv parsing scheme [25] – also known as LZ78 algorithm – that was proved to be universal and asymptotically optimal. This scheme is used in the UNIX `compress` command and in a CCITT standard for data compression for modems. To recall, the algorithm first partitions a training sequence (dictionary or database) of length n into variable phrases such that the next phrase is the shortest phrase not seen in the past. The code consists of pairs of numbers: each pair being a pointer to the previous occurrence of the prefix of the phrase and the last bit of the phrase. Thus, if M_n is the number of phrases constructed from the training sequence, then the code length ℓ_n is[1]

$$\ell_n = M_n(\log M_n + 1) . \tag{1}$$

The pointwise redundancy r_n and its expected value $\overline{r}_n$ for a given source of the Lempel-Ziv code are respectively

$$\begin{aligned} r_n &= \frac{M_n(\log M_n + 1) - nh}{n} , & (2) \\ \overline{r}_n := Er_n &= \frac{E\{M_n(\log M_n + 1)\} - nh}{n} , & (3) \end{aligned}$$

where h is the entropy rate of the source. Plotnik, Weinberger and Ziv proved in [16] that the expected redundancy of the Lempel-Ziv code is $\overline{r}_n = O(\log\log n/\log n))$ for finite-state sources. But, the authors of [16] also noticed that for infinitely many sequences the pointwise

[1] Throughout the paper we shall write $\log(\cdot)$ for binary logarithm $\log_2(\cdot)$.

redundancy rate is bounded from the below by $2/\log n$. In this paper we prove that this lower bound is actually attainable for the expected redundancy rate $\overline{r}_n$, just closing the gap between the upper and the lower bounds. Moreover, we shall provide a precise asymptotic formula for $\overline{r}_n$ which will indicate that the coefficient at $1/\log n$ contains a fluctuating function.

In order to present our main results we must introduce some notation. Define for large (say, integer) x a function $\mu(x)$ as follows:

$$\mu(x) = \frac{x}{h}\log x - \frac{A}{h}x + O\left(\frac{\log x}{h}\right) \tag{4}$$

where $A = O(1)$ and will be specified below. Let x_n be a positive solution of the following equation

$$\mu(x_n) = n\ . \tag{5}$$

Observe that the above equation has the following asymptotic solution for large n

$$x_n = \frac{nh}{\log n}\left(1 + \frac{\log\log n}{\log n} + \frac{A - \log h}{\log n} + O\left(\frac{(\log\log n)^2}{\log^2 n}\right)\right)\ . \tag{6}$$

In the next section we prove the following main result. Hereafter, for the simplicity of the presentation we restrict our analysis to binary alphabet, but extension to any finite alphabet is straightforward.

Theorem. (i) *Consider a memoryless binary source with symbol "0" occuring with probability p and symbol "1" with probability $q = 1-p$. Let M_n be the number of phrases obtained after parsing a sequence of length n according to the Lempel-Ziv algorithm. Then, for any $k \geq 1$*

$$EM_n^k = x_n^k\left(1 + O\left(\sqrt{\frac{\log n}{n}}\right)\right) + O\left(\frac{n^{k-1}}{\log^{k-1} n}\right)\ . \tag{7}$$

More interestingly, the average redundancy of the Lempel-Ziv code becomes

$$\overline{r}_n = \frac{2h - h\gamma - \frac{1}{2}h_2 + h\alpha - h\delta_0(n)}{\log n} + O\left(\frac{\log\log n}{\log^2 n}\right)\ , \tag{8}$$

where $h = -p\log p - q\log q$ is the entropy, $\gamma = 0.577\ldots$ is the Euler constant, $h_2 = p\log^2 p + q\log^2 q$, and $\delta_0(n)$ is a fluctuating functions with small amplitude for $\log p/\log q$ rational, and zero otherwise. Finally, the constant α is defined as:

$$\alpha = -\sum_{k=1}^{\infty}\frac{p^{k+1}\log p + q^{k+1}\log q}{1 - p^{k+1} - q^{k+1}}\ , \tag{9}$$

(ii) *The above results hold for a Markovian source with h_2 and α expressed as in [7, 10].*

We point out that the above result should be compared with recent findings of Jacquet and Szpankowski [9] who proved that for a memoryless source $\Pr\{r_n > \varepsilon\} \leq A\exp(-a\varepsilon\sqrt{n})$ for some constants A, a, and small $\varepsilon > 0$. In passing, we note that the main result of [9] is instrumental for the proof of our Theorem.

Furthermore, the above redundancy result should also be compared to the average redundancy of another version of the Lempel-Ziv scheme [24], namely that of *fixed-database* or *sliding window* known also as LZ77. It is easy to see from recent results of Jacquet and Szpankowski [8] (cf. also [7]) that the average redundancy $\overline{r}_n$ for memoryless and Markovian sources becomes

$$\overline{r}_n = h\frac{\log\log n}{\log n} + O\left(\frac{\log\log n}{\log^2 n}\right) .$$

The redundancy of this scheme is larger than that of LZ78, thus the sliding-window version converges slower to the optimal compression ratio. Observe also that there is no fluctuating term in front of the leading term of $\overline{r}_n$ in the LZ77 scheme. In passing, we should mention that recently Wyner and Wyner [22] proved that a modification of the fixed-database scheme can achieve the redundancy of order $O(1/\log n)$. We conjecture that the coefficient at $1/\log n$ in this new scheme is not a constant but a fluctuating function, as in the case of LZ78 scheme.

In concluding remarks of this paper, we extend Theorem to a generalized Lempel-Ziv parsing algorithm recently proposed by us in [13]. This new algorithm partitions a sequence into phrases such that the next phrase is the longest substring seen in the past by at most $b-1$ phrases. The case $b = 1$ corresponds to the original Lempel-Ziv parsing scheme. We indicate that this new scheme, at least for symmetric memoryless source (equal probabilities of symbol generations), can slightly improve the average redundancy of Lempel-Ziv-like codes (however, more research is need to verify this conclusion for other sources). We also briefly discuss a similar generalization of the sliding window Lempel-Ziv scheme.

2. ANALYSIS

The proof of Theorem is by reduction, that is, we reduce the problem under investigation to another one on digital trees that is easier to handle. We have already applied this strategy successfully in the past (cf. [9, 12]).

The reader is referred to [11, 14] for a discussion and the definition of digital trees. In short: the root of the tree is empty. All other phrases of the Lempel-Ziv parsing algorithm are stored in internal nodes. When a new phrase is created, the search starts at the root and proceeds down the tree as directed by the input symbols exactly in the same manner as in the digital tree construction, that is, symbol "0" in the input string means a move to the left and "1" means a move to the right. The search is complete when a branch is taken from an existing tree node to a new node that has not been visited before. Then, the edge and the new node are added to the tree. (cf. Figure 1 in [9, 12] and in Section 3).

Observe that for fixed n the number of nodes in the associated digital tree is random and equal to M_n. However, it is to our advantage to consider also a digital tree in which the number of nodes is fixed and equal to m. We call such a model the *digital tree model* while the original problem (i.e., with fixed length n of a word to parse) we name the *Lempel-Ziv model.*[2] The digital tree model was investigated in [3, 9, 12]. In the digital tree model, we denote by $D_m(i)$ the length of the path from the root to the ith node (the ith depth). Then, the internal path length L_m is defined as $L_m = \sum_{i=1}^{m} D_m(i)$.

In view of the above definitions, it is clear that M_n satisfies the following *renewal equation* (cf. [9])

$$M_n = \max\{m : \ L_m = \sum_{k=1}^{m} D_m(i) \leq n\} \ , \tag{10}$$

which directly implies that

$$\Pr\{M_n > m\} = \Pr\{L_m \leq n\} \ . \tag{11}$$

Indeed, consider building a dynamic digital search tree from phrases. Each time a phrase is created we add it as a new word to the digital tree. The tree grows, and we continue this process until for the first time the internal path length becomes n. Clearly, the number of inserted words at this time is M_n (cf. [3, 9, 12]).

The above relationship is crucial, and should lead to a complete characterization of M_n if one can analyze L_m. This follows from a result of Billingsley (cf. Theorem 17.3 in [1]) which claims that if

$$\frac{L_m - \mu_m}{\sigma_m} \to N(0,1) \ , \tag{12}$$

then

$$\frac{M_n - n/(\mu_m/m)}{\sqrt{n(\sigma_m/m)/(\mu_m/m)^3}} \to N(0,1) \tag{13}$$

where $N(0,1)$ is the standard normal distribution, and μ_m and σ_m are positive constants that under mild standard uniform integrability arguments can be asymptotically interpreted as the mean and the variance of L_m.

We concentrate on proving Theorem for a memoryless source (also known as the Bernoulli model). Recently, Jacquet and Szpankowski [9] proved that L_m appropriately normalized

[2] Hereafter, we shall consistently use n as the length of a single word to be parsed, and m as the number of words used to construct a digital search tree.

is normally distributed, so by Billingsley's result M_n is also normally distributed. But, to derive second order properties of the Lempel-Ziv algorithm (such as the redundancy), we need the rate of convergence to the normal distribution. The authors of [9] obtained such a rate for the path length L_m, but not for M_n (since (13) does not provide it). Therefore, we shall deal mostly with the path length L_m instead of M_n.

For the reader convenience we present below the main results of Jacquet and Szpankowski [9] that are necessary to prove our main results. Below, we write

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \tag{14}$$

fro the distribution function of the standard normal distribution. Now we are ready to recall main results of Jacquet and Szpankowski [9]:

Fact A. *Consider a digital search tree built from m independent words under the asymmetric memoryless source.*

(i) *Asymptotically the average value EL_m and the variance* Var L_m *become*

$$\begin{aligned} EL_m &= \frac{m}{h}\left(\log m + \frac{h_2}{2h} + \gamma - 1 - \alpha + \delta_0(\log m)\right) \\ &+ \frac{1}{h}\left(\log m + \frac{h_2}{2h} - \gamma - \log p - \log q + h\alpha\right) + O(1) \qquad (15) \\ \text{Var } L_m &= c_2 m \log m + O(m) \qquad (16) \end{aligned}$$

$c_2 = (h_2 - h^2)/h^3$, and $\delta_0(\log m)$ is a fluctuating functions for $\log p / \log q$ rational with small amplitude, and zero otherwise, and h, h_2 and α are defined in Theorem above.

(ii) *For large m the following weak convergence takes place*

$$\Pr\{L_m \le EL_m + x\sqrt{\text{Var} L_m}\} = \Phi(x)\left(1 + O(1/\sqrt{m})\right) \tag{17}$$

for $x = o(\sqrt{m})$.

(iii) *The above is true for symmetric memoryless source (i.e., symbols occur with the same probability) if one replace the variance above by the following*

$$\text{Var } L_m^{sym} \sim m \cdot (C + \delta(\log_2 m)) \tag{18}$$

where $C = 0.26600\ldots$ and $\delta(x)$ is a fluctuating function with small amplitude.

Fact B. *In the asymmetric Bernoulli model, define $Z_n = \frac{M_n - EM_n}{\sqrt{\text{Var} M_n}}$. Then:*

(i) *The sequence of random variables Z_n converges weakly (i.e., in distribution) to $N(0,1)$. In addition, for all $r \geq 0$ the sequence $(Z_n)^r$ is uniformly integrable. Thus, all moments of Z_n exist and converge to the appropriate moments of the normal distribution. In particular,*

$$EM_n \sim \frac{nh}{\log(n)} \tag{19}$$

$$\mathrm{Var}\ M_n \sim \frac{c_2 h^3 n}{\log^2 n} \tag{20}$$

where $c_2 = (h_2 - h^2)/h^3$.

(ii) *For any $\varepsilon > 0$, there exist an integer $n_0 \geq 1$ such that for all $n > n_0$*

$$\Pr\{|M_n - EM_n| > \varepsilon EM_n\} \leq A \exp\left(-a\varepsilon\sqrt{n}\right) \tag{21}$$

for some positive constants $A, a > 0$.

(iii) *The above results are also true for the symmetric (i.e., unbiased) memoryless source if one replaces the variance by*

$$\mathrm{Var}\ M_n^{sym} \sim \frac{n(C + \delta(\log_2 n))}{\log_2^3 n} \tag{22}$$

where the constant $C = 0.26600\ldots$. In (ii) one must replace $\sqrt{n}$ by $\sqrt{n/\log n}$.

Now, we are in position to prove Theorem (i) for a memoryless source. From (11) we easily derive the kth moment of M_n. Indeed:

$$\begin{aligned} EM_n^{k+1} &= (k+1)\sum_{m\geq 0} m^k \Pr\{M_n > m\} = (k+1)\sum_{m \geq 0} m^k \Pr\{L_m \leq n\} \\ &= (k+1)\int_0^\infty x^k \Pr\{L_x \leq n\} dx + O(EM_n^k) \end{aligned} \tag{23}$$

where the last estimate follows from the Euler-Maclaurin formula [11]. To verify, it suffices to do some elementary algebra on the Euler-Maclaurin formula that is recalled below: *For any function $f(k)$, we have*

$$\begin{aligned} \sum_{k=a}^{b} f(k) &= \int_a^b f(x)dx - \frac{f(b) - f(a)}{2} + \sum_{k=1}^{j} \frac{B_{2k}}{(2k)!}\left(f^{(2k-1)}(b) - f^{(2k-1)}(a)\right) \\ &+ O((2\pi)^{-2j}\int_a^b |f^{(2j)}(x)|dx \end{aligned}$$

where $f^{(j)}(x)$ is the jth derivative of $f(x)$, and B_k is the kth Bernoulli number.

Thus, by Fact A and the above we arrive at

$$\begin{aligned} EM_n^{k+1} &= (k+1)\int_0^{x_n} x^k(1+O(1/\sqrt{x}))\Phi\left(\frac{n-\mu(x)}{\sigma(x)}\right)dx \\ &+ (k+1)\int_{x_n}^{\infty} x^k(1+O(1/\sqrt{x}))\Phi\left(\frac{n-\mu(x)}{\sigma(x)}\right)dx + O(EM_n^k) \end{aligned}$$

where we use simplified notation $\mu(x) = EL_x$ and $\sigma(x) = \text{Var } L_x$. The quantity x_n above is the same as in (5), that is, $n = \mu(x_n)$. Note that x_n is given asymptotically by (6).

In order to estimate the above integrals, we first recall definition of the error function $\text{erf}(x)$:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \ .$$

Observing that $\Phi(x) = 1/2 + 1/2\text{erf}(x/\sqrt{2})$, we estimate the first integral above as (with the error term $1+O(1/\sqrt{x})$ dropped for the simplicity of presentation)

$$\int_0^{x_n} x^k\Phi\left(\frac{n-\mu(x)}{\sigma(x)}\right)dx = \frac{x_n^{k+1}}{k+1} - \frac{1}{2}\int_0^{x_n} x^k\left(1-\text{erf}\left(\frac{n-\mu(x)}{\sqrt{2}\sigma(x)}\right)\right)dx \ .$$

In a similar fashion, the second integral becomes

$$\int_{x_n}^{\infty} x^k\Phi\left(\frac{n-\mu(x)}{\sigma(x)}\right)dx = \frac{1}{2}\int_{x_n}^{\infty} x^k\left(1-\text{erf}\left(\frac{\mu(x)-n}{\sqrt{2}\sigma(x)}\right)\right)dx \ .$$

In summary, we have

$$\begin{aligned} EM_n^{k+1} &= x_n^{k+1}(1+O(1/\sqrt{x_n})) - \frac{k+1}{2}\int_0^{x_n} x^k(1+O(1/\sqrt{x}))\left(1-\text{erf}\left(\frac{n-\mu(x)}{\sqrt{2}\sigma(x)}\right)\right)dx \\ &+ \frac{k+1}{2}\int_{x_n}^{\infty} x^k(1+O(1/\sqrt{x}))\left(1-\text{erf}\left(\frac{\mu(x)-n}{\sqrt{2}\sigma(x)}\right)\right)dx + O(x_n^k) \ . \qquad (24) \end{aligned}$$

We compute the above two integrals separately. Let us denote them as follows

$$\begin{aligned} I_1 &= \int_0^{x_n} x^k\left(1-\text{erf}\left(\frac{n-\mu(x)}{\sqrt{2}\sigma(x)}\right)\right)dx \ , \\ I_2 &= \int_{x_n}^{\infty} x^k\left(1-\text{erf}\left(\frac{\mu(x)-n}{\sqrt{2}\sigma(x)}\right)\right)dx \ . \end{aligned}$$

We make the following change of variables

$$y = \frac{\mu(x)-n}{\sigma(x)}$$

and we simplify $\mu(x) \sim c_1 x\log x$ and $\sigma(x) \sim \sqrt{c_2 x\log x}$ where we write $c_1 = 1/h$. Observe that x is a function of y, so we often write $x(y)$. It satisfies the following equation $c_1 x\log x -$

$y\sqrt{c_2 x \log x} - n = 0$, which implies

$$\begin{aligned}
x(y)\log x(y) &= \left(\frac{y\sqrt{c_2}+\sqrt{c_2 y^2+4nc_1}}{2c_1}\right)^2 \\
&= nh\left(1+\frac{c_2hy^2}{2n}+\frac{y\sqrt{hc_2}}{\sqrt{n}}\sqrt{1+\frac{hc_2y^2}{4n}}\right) \\
&= nh\left(1+\frac{y\sqrt{hc_2}}{\sqrt{n}}+\frac{y^2c_2h}{2n}+\frac{y^3c_2h\sqrt{hc_2}}{8n^{3/2}}+O(y^5/n^{5/2})\right) .
\end{aligned}$$

From the above and after some further Taylor's expansion of $\log x(y)$ we derive the following (with the help of MAPLE)

$$\frac{dx(y)}{dy} = \sqrt{n}(1+G(n))\left(\frac{h^{3/2}\sqrt{c_2}}{\log n}+\frac{yh^2c_2}{\sqrt{n}\log n}+O\left(\frac{1}{n}\right)\right)$$

and

$$x(y) = n(1+G(n))\left(\frac{h}{\log n}+\frac{yh^{3/2}\sqrt{c_2}}{\sqrt{n}\log n}+\frac{y^2h^2c_2}{2n\log n}+O\left(\frac{y^3}{n^{3/2}}\right)\right)$$

where

$$G(n) = \frac{\log\log n}{\log n}+\cdots .$$

The above substitutions are needed to compute the above integrals which become

$$\begin{aligned}
I_1 &= \int_0^\infty x^k(-y)(1-\mathrm{erf}(y/\sqrt{2}))\frac{dx(-y)}{dy}dy , \\
I_2 &= \int_0^\infty x^k(y)(1-\mathrm{erf}(y/\sqrt{2}))\frac{dx(y)}{dy}dy .
\end{aligned}$$

Let us first estimate I_2 for $k=0$. We obtain

$$\begin{aligned}
I_2 &= \int_0^\infty (1-\mathrm{erf}(y/\sqrt{2}))\frac{dx(y)}{dy}dy \\
&= (1+G(n))\int_0^\infty (1-\mathrm{erf}(y/\sqrt{2}))\left(\sqrt{\frac{n(h_2-h^2)}{\log^2(n)}}+\frac{yc_2h^2}{\log(n)}+O\left(\frac{y^2}{\sqrt{n}}\right)\right)dy \\
&= (1+G(n))\left(\sqrt{\frac{n(h_2-h^2)}{\pi\log^2(n)}}+\frac{c_2h^2}{2\log(n)}+O\left(\frac{1}{\sqrt{n}}\right)\right) .
\end{aligned}$$

To compute the above integral we used the following well known identity (cf. [4] Eq. (6.281))

$$\int_0^\infty (1-\mathrm{erf}(px))x^{2q-1} = \frac{\Gamma(q+1/2)}{2\sqrt{\pi}qp^{2q}}$$

for $p>0$ and $q>0$. We also need $\Gamma(3/2)=\sqrt{\pi}/2$ (cf. [4]). In a similar manner we can compute I_1. Indeed, we have

$$I_1 = (1+G(n))\left(\sqrt{\frac{n(h_2-h^2)}{\pi\log^2(n)}}-\frac{c_2h^2}{2\log(n)}+O\left(\frac{1}{\sqrt{n}}\right)\right) .$$

Thus, putting everything together we get

$$EM_n = x_n \left(1 + O\left(\sqrt{\frac{\log n}{n}}\right)\right) + O(1)$$

This proves Theorem for $k = 0$, and following the same lines of arguments we prove Theorem for any k.

To prove the second part of Theorem concerning the average redundancy, we must evaluate $E\{M_n(\log M_n + 1)\}$. Let N be a random variable distributed as the standard normal distribution. Then by Fact B we can write $M_n = \overline{M}_n + N\sigma_n + X$ where $X = o(\sqrt{n})$ (pr.) (in fact, (21) suggests that $X = O(1)$ but we use only the above weaker condition to derive our results). We also use the abridge notation: $EM_n = \overline{M}_n \sim nh/\log n$, and $\sigma_n = \sqrt{\mathrm{Var} M_n} \sim \sqrt{c_2 h^3 n/\log^2 n}$. Using (19)-(20), one easily proves the following

$$\begin{aligned} M_n \log M_n &= (\overline{M}_n + N\sigma_n + X)\log \overline{M}_n \left(1 + \frac{N\sqrt{c_2 h}}{\sqrt{n}} + \frac{X}{\overline{M}_n}\right) \\ &= (\overline{M}_n + N\sigma_n + X)\left(\log \overline{M}_n + \frac{N\sqrt{c_2 h}}{\sqrt{n}} + \frac{X}{\overline{M}_n} - \frac{N^2 c_2 h}{2n}\right) \end{aligned}$$

After some algebra and noting that $EN = EX = EN^3 = 0$ we obtain

$$E\{M_n(\log M_n + 1)\} = \overline{M}_n(\log \overline{M}_n + 1) + O(1/\log n) \ .$$

Now observe that $\overline{M}_n = x_n(1 + O(\sqrt{\log n/n})$ where x_n is defined in (5). Using the asymptotic expansion (6) of x_n, we finally prove formula (8) on the average redundancy. Note that the term $\log\log n/\log n$ cancels! This completes the proof of Theorem (i).

The proof of Theorem (ii) for Markovian sources follows the same footsteps as above with Facts A and B already anticipated in [9], and will be formally proved in a forthcoming paper [10].

3. EXTENSIONS AND CONCLUDING REMARKS

In [12] we have introduced the following generalization of the Lempel-Ziv parsing scheme: the next phrase is the longest phrase seen in the past by *at most* $b - 1$ phrases. The case $b = 1$ corresponds to the original Lempel-Ziv algorithm. For example, the sequence discussed above is partitioned as (1)(1)(0)(0)(10)(10)(00)(100)(01)(00)(11) for $b = 2$. For a similar generalization of the sliding window version of the Lempel-Ziv scheme (LZ77) the reader is referred to Szpankowski [20].

A data compression code for this new algorithm may consists of pairs of number: one being a pointer to the previous occurrence of the prefix of the phrase, and the second

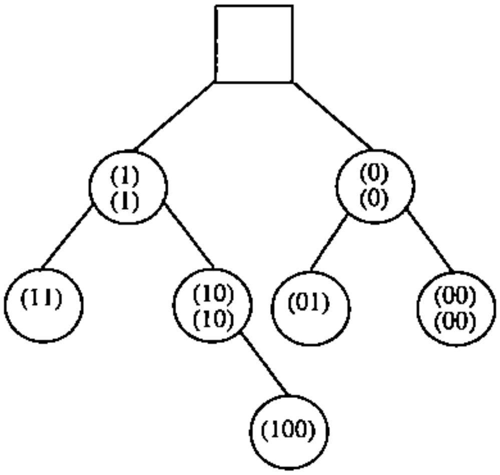

Figure 1: A 2-digital search tree representation of the generalized Lempel-Ziv parsing for the string 1100101000100010011

number is either empty space or the last bit of the phrase in the case it is the bth phrase. Observe that the length ℓ_n of such a code depends on two parameters: Namely, the number of phrases $M_n(b)$, and the number of *distinct* phrases $M_n'(b)$. With this notation in mind, we express the length ℓ_n as a function of $M_n(b)$ and $M_n'(b)$ as follows:

$$\ell_n(X_1^n) = M_n(b)(\log M_n'(b) + I(b)) \tag{25}$$

where $I(b)$ is equal to one if the phrase consists of a previous prefix and one more bit (i.e., already $b-1$ phrases have occurred), and zero otherwise.

As in the case $b = 1$, to study this generalization of the Lempel-Ziv scheme we construct a special digital search tree called b-digital search tree which stores up to b phrases in a node (cf. Figure 1). We also consider the case when the number of phrases is fixed and equal to m (as before we call such a model the digital tree model to distinguish it from the Lempel-Ziv model). The details of the tree construction can be found in [11, 12].

The analysis of b-digital trees, even with fixed number of strings, is much more complicated than for the case $b = 1$, as explained in [2, 12]. While for $b = 1$ some recurrence equations have explicit solutions, this is not any longer true for $b > 1$. The digital tree model for unbiased (symmetric) Bernoulli model was first investigated by Flajolet and Richmond [2] (cf. [5]). In a forthcoming paper [10] it will be presented a full characterization of b-digital trees. Let us summarize some of our anticipated results.

For fixed (number of strings) m, let S_m and L_m denote respectively the size (number of nodes) of a tree and the internal path length (sum of all depths to *all strings*). In

[2, 5, 10, 21] one can extract the following results

$$\begin{aligned} ES_m &= m[q_0(b) + \delta_1(m)] + O(1) , & (26) \\ EL_m &= \frac{m}{h}\left(\log m + \frac{h_2}{2h} - H_{b-1} + \omega + \gamma - 1 + \delta_2(m)\right) + O(\log m) , & (27) \end{aligned}$$

where $q_0(b)$ and ω are some constants, and H_b is the harmonic number. The functions $\delta_1(m)$, $\delta_2(m)$, and $\delta_3(m)$ are fluctuating functions with small amplitudes. For example, for the symmetric (unbiased coin tossing) Bernoulli model Flajolet and Richmond [2] computed

$$q_0(b) = \frac{1}{\log 2}\int_0^\infty \left(\frac{1+t}{Q(t)}\right)^b \frac{dt}{1+1} , \qquad (28)$$

where $Q(t) = \prod_{j=0}^\infty (1 + t2^{-j})$. Clearly, $q_0(1) = 1$, and the authors of [2] computed $q_0(2) = 0.5747$, $q_0(3) = 0.4069$, and so on. For large b one easily derives from (28) that $q_0(b) \sim 1/(b \log 2)$ as $b \to \infty$ (cf. [2]).

Moreover, in [10] we shall prove that L_m and S_m after proper normalizations are normally distributed. Thus, an equivalence of Fact A holds for b-digital search trees, however, the proof is much more complicated.

As before, the parameters of the Lempel-Ziv model can be expressed in terms of the corresponding parameters of the digital tree model. In particular:

$$\begin{aligned} M_n(b) &= \max\{m : \; L_m \leq n\} , \\ M_n'(b) &= S_{M_n} . \end{aligned}$$

Furthermore, from the construction of the compression code we observe that the second number in the code is nonempty (i.e., it is equal to a single bit) whenever an overflow occurs in the associated digital search tree, that is, a new phrase arrives to a full node. Thus, $EI(b) = E(M_n'(b) - 1)/(EM_n) \sim q_0(b)$.

Using the above anticipated results, we are in position to establish the average redundancy $\overline{r}_n(b)$ of the generalized Lempel-Ziv code. Finally, after some algebra similar to the one performed for the $b = 1$ case, we obtain

$$\overline{r}_n(b) = h\frac{1 - \gamma - \frac{h_2}{2h} - \omega + H_{b-1} + q_0(b) + \log q_0(b) - \delta(n)}{\log n} + O\left(\frac{\log\log n}{\log^2 n}\right) , \qquad (29)$$

where $\delta(n)$ is a fluctuating function with a small amplitude, and the other quantities are defined as before.

It might be interesting to compare the average redundancy for different values of b hoping that there exists an optimal value of b. At this point, we have full understanding (and computations) for the unbiased memoryless source (cf. [2]). Our computation show that

$$\begin{aligned}\overline{r}_n(1) &= \frac{2.68+\delta(n)}{\log n}+O\left(\frac{\log\log n}{\log^2 n}\right) ,\\ \overline{r}_n(\infty) &= \frac{1.87+\delta(n)}{\log n}+O\left(\frac{\log\log n}{\log^2 n}\right) .\end{aligned}$$

Thus, (at least for the unbiased case) there exists an optimal value of b which minimizes the average redundancy. We are planning to investigate this problem more deeply by extending our computations to unbiased memoryless and Markovian sources, and by performing some experiments on real data.

Finally, we consider a similar extension of the sliding window Lempel-Ziv (LZ77) scheme announced in Szpankowski [20]. The idea is as follows: Let X_1^n be a fixed database (training) sequence. We search now for the longest prefix of X_{n+1}^{∞} that occurs *at most* b times in the database, and we denote the length of this longest prefix as $L_n(b)$. The compression code contains the pointer to the *first* occurrence of the prefix in the database and the length $L_n(b)$ of the longest prefix. Clearly, $L_n(b)$ is smaller then $L_n(1)$ (which is bad) but due to (at most) b repetitions we do not need $\log n$ bits to store pointers but less (hopefully $\log(n/b)$ – which is not really correct as we shall see below).

Our goal is to estimate the average redundancy $\overline{r}_n(b)$ and compare it with the redundancy of LZ78 code as well as for different values of b in order to select the best b. To estimate the number of bits necessary to store pointers to the database, we need to represent the database as the so called b-suffix tree introduced in [20]. This is an ordinary suffix tree that allows to store up to b (sub)strings in an external node (similar to the b extension of the digital tree as discussed above). Observe as in [20] that $L_n(b)$ is just the depth of insertion, while the number of *distinct* pointers to the database is the *number of external nodes* in the associated b-suffix tree. We denote the latter quantity by $S_n(b)$. Then, as in [22] we can write

$$\overline{r}_n(b)=\frac{E\log S_n(b)+E\log L_n(b)}{EL_n(b)}-h . \tag{30}$$

To estimate $\overline{r}_n(b)$ above, we must evaluate $L_n(b)$ and $S_n(b)$. As proved in [8], the above parameters of a suffix tree do not differ to much from the corresponding parameters of a trie built from n *independent* strings (see [8] for a more precise statement). Thus, using the results of [18] we immediately obtain

$$EL_n(b)=\frac{1}{h}\log n-\frac{1}{h}H_{b-1}+\frac{\gamma}{h}+\frac{h_2}{2h^2}+\delta_3(n)+O\left(\frac{1}{n}\right)$$

with the same notation as before, and H_b denoting the bth Harmonic sum.

The average size ES_n seemed not to be analyzed before except for the symmetric case (cf. [15]). But it is easy to see that it satisfies the following recurrence: $ES_0(b) = 0$, $E_1(b) = \ldots = ES_b(b) = 1$ and

$$ES_n(b) = \sum_{k=0}^{n} \binom{n}{k} p^k q^{n-k} (ES_k(b) + ES_{n-k}(b)) \ .$$

This recurrence equation can be solved using a general result of Szpankowski [18], and we obtain

$$ES_n(b) = n - \sum_{k=2}^{n} (-1)^k \binom{n}{k} \frac{\sum_{r=2}^{b} (-1)^r \binom{k}{r} (1 - p^r - q^r)}{1 - p^k - q^k} \ .$$

Using Mellin-like approach or Rice's method (cf. [12, 18, 19]), we easily derive an asymptotic expansion which becomes

$$ES_n(b) = n\left(1 - \left(1 - \frac{1}{b} - \sum_{r=2}^{b} \frac{p^r + q^r}{r(r-1)}\right) + \delta_4(n)\right) + O(1) = n \cdot c(b) + O(1)$$

where $\delta_4(n)$ is another fluctuating function with a small amplitude. Observe that $c(b) < 1$.

Finally, using Jacquet and Régnier [6] and Jacquet and Szpankowski [8] we conclude – using the same arguments as above for LZ78 scheme – that $E \log S_n(b) \sim \log ES_n(b)$ and $E \log L_n(b) \sim \log EL_n(b)$. Thus, putting everything together we finally obtain

$$\overline{r}_n(b) = h \frac{\log(\log n - H_{b-1}) + \log c(b)}{\log n - H_{b-1}} + O\left(\frac{1}{\log n}\right) \ . \tag{31}$$

For large b, we can approximate $H_b \sim \log b$, and then

$$r_n(b) = h \frac{\log \log(n/b) - \log(b(1-h))}{\log(n/b)} + O\left(\frac{1}{\log n}\right) \ . \tag{32}$$

From the above, we conclude that the redundancy of LZ78 code is better than LZ77. Furthermore, the extension discussed in this section is more relevant for the LZ78 code than for LZ77. From (32) we also observe that the above extension can slightly improve the redundancy of the sliding window Lempel-Ziv scheme for not too large n, that is, when the term $\log(b(1-h))$ is comparable to $\log\log(n/b)$.

ACKNOWLEDGEMENT

We would like to thank M. Feder, P. Jacquet, J. Kieffer and P. Shields for helpful comments and discussions regarding this work.

References

[1] P. Billingsley, *Convergence of Probability Measures*, John Wiley & Sons, New York 1968.

[2] P. Flajolet and B. Richmond, Generalized Digital Trees and Their Difference-Differential Equations, *Random Structures & Algorithms*, 3, 305-320, 1992.

[3] E. Gilbert and T. Kadota, The Lempel-Ziv Algorithm and Message Complexity, *IEEE Trans. Information Theory*, 38, 1839-1842, 1992.

[4] I. Gradshteyn and I. Ryznik, *Tables, Integrals, Series, and Products*, Academic press, New York 1980.

[5] F. Hubalek, Further Results on Generalized Digital Trees – The Mellin Convolution Approach, *Theoretical Computer Science*, to appear.

[6] P. Jacquet and M. Régnier, Normal Limiting Distribution of the Size of Tries, *Proc. Performance'87*, 209-223, North Holland, Amsterdam 1987

[7] P. Jacquet and W. Szpankowski, Analysis of Digital Tries with Markovian Dependency, *IEEE Trans. Information Theory*, 37, 1470-1475, 1991.

[8] P. Jacquet and W. Szpankowski, Autocorrelation on Words and Its Applications. Analysis of Suffix Trees by String-Ruler Approach, *J. Combin.Theory Ser. A*, 66, 237-269, 1994.

[9] P. Jacquet and W. Szpankowski, Asymptotic Behavior of the Lempel-Ziv Parsing Scheme and Digital Search Trees, *Theoretical Computer Science*, 144, 161-197, 1995.

[10] P. Jacquet and W. Szpankowski, Asymptotic Behavior of Generalized Lempel-Ziv Parsing Scheme for Markovian Sources, in preparation.

[11] D. Knuth, *The Art of Computer Programming. Sorting and Searching*, Addison-Wesley, 1973.

[12] G. Louchard and W. Szpankowski, Average Profile and Limiting Distribution for a Phrase Size in the Lempel-Ziv Parsing Algorithm, *IEEE Trans. Information Theory*, 41, 478-488, 1995.

[13] G. Louchard and W. Szpankowski, Generalized Lempel-Ziv Parsing Scheme and its Preliminary Analysis of the Average Profile, *Proc. Data Compression Conference*, 262-271, Snowbird, 1995.

[14] H. Mahmoud, *Evolution of Random Search Trees*, John Wiley & Sons, New York 1992.

[15] H. Mahmoud and T. Papadakis, A Probabilistic Analysis of Fixed and Elastic Buckets in Tries and Patricia Trees, *Proc. 30th Allerton Conference*, 874-883, Monticello, 1992.

[16] E. Plotnik, M.J. Weinberger, and J. Ziv, Upper Bounds on the Probability of Sequences Emitted by Finite-State Sources and on the Redundancy of the Lempel-Ziv Algorithm, *IEEE Trans. Information Theory*, 38, 66-72, 1992.

[17] P. Shields, Universal Redundancy Rates Do Not Exist, *IEEE Information Theory*, 39, 520-524, 1993.

[18] W. Szpankowski, Some Results on V-ary Asymmetric Tries, *J. Algorithms*, 9, 224-244, 1988.

[19] W. Szpankowski, A Characterization of Digital search Trees From the Successful Search Viewpoint, *Theoretical Computer Science*, 85, 117-134, 1991.

[20] W. Szpankowski, A Generalized Suffix Tree and Its (Un)Expected Asymptotic Behaviors, *SIAM J. Computing*, 22, 1176-1198, 1993.

[21] W. Szpankowski and J. Tang, Analysis of a Digital Search Tree with Applications to a Generalized Lempel-Ziv Algorithm, *Proc. 33-rd Annual Allerton Conference*, 1995.

[22] A.D. Wyner and A.J. Wyner, Improved Redundancy of a Version of the Lempel-Ziv Algorithm, *IEEE Trans. Information Theory*, 41, 723-732, 1995.

[23] J. Ziv, Compression, Test of Randomness, and Estimating the Statistical Model of Individual Sequences, *SEQUENCES*, R. Capocelli, Ed. New York: Springer-Verlag, 366-373, 1990.

[24] J. Ziv and A. Lempel, A Universal Algorithm for Sequential Data Compression, *IEEE Trans. Information Theory*, 23, 337-343, 1977.

[25] J. Ziv and A. Lempel, Compression of Individual Sequences via Variable-rate Coding, *IEEE Trans. Information Theory*, 24, 530-536, 1978.