

$e^{i(\omega_1 - \omega_q)n} \leq 2$ , gives

$$\begin{aligned} (A_1 A_1^* g, g) &\leq (MV_n(\omega_1) g, g) = \gamma_n(\omega_1, g) \\ &\leq (1/n^2)(E_n(\omega_1)^* T_n E_n(\omega_1) g, g) \\ &\leq (A_1 A_1^* g, g) + \frac{4}{n^2} \sum_{q=2}^m \frac{\|A_q^* g\|^2}{|1 - e^{i(\omega_1 - \omega_q)}|^2} \\ &\quad + \beta \|g\|^2 \frac{1}{n^\epsilon}. \end{aligned} \quad (17)$$

Since (17) holds for any  $g$  in  $\mathbb{C}^p$ , and the rightmost two terms go to zero as  $n \rightarrow \infty$ , it follows from (10) that  $MV_n(\omega_1) \rightarrow A_1 A_1^*$ . To show that (9) converges to zero if  $\omega$  is not a signal frequency, it suffices to set  $A_1 = 0$ . Then  $\omega_1$  is no longer a signal frequency. In this case, (17) is bounded below by zero and above by only the rightmost two terms, both of which go to zero as  $n \rightarrow \infty$ . This proves the theorem.  $\square$

This result justifies the use of the  $MV(n)$  spectrum in a wide variety of unknown stationary or nonstationary noise environments. The proof provided explicit error bounds, (17), associated with other signal components as well as the noise. Notice that the  $p \times p$  matrix  $(1/n^2)E_n(\omega_1)^* T_n E_n(\omega_1)$  is simply the weighted discrete Fourier transform (DFT) estimate of the signal point spectrum, where the signal covariance data has been weighted by a triangular window. It follows from (17) that the DFT "spectrum"  $(1/n^2)E_n(\omega_1)^* T_n E_n(\omega_1)$  also converges to the signal point spectrum as  $n \rightarrow \infty$  for a large class of nonstationary as well as stationary noise processes. If  $T_{Nn}$  corresponds to a nonstationary noise process, this quantity is not a spectrum in the usual sense. In any case, it also provides an upper bound for the  $MV(n)$  spectrum for each value of  $n$ .

#### REFERENCES

- [1] V. M. Adamjan, "Asymptotic properties for positive and Toeplitz matrices and operators," in *Operator Theory: Advances and Applications*, vol. 43. Basel: Birkhauser Verlag, 1990, pp. 17-38.
- [2] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, pp. 1408-1418, 1969.
- [3] C. Fofias, A. E. Frazho, and P. J. Sherman, "A geometric approach to the maximum likelihood spectral estimator for sinusoids in noise," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 1066-1070, Sept. 1988.
- [4] C. Fofias, A. E. Frazho, and P. J. Sherman, "A new approach for determining the spectral data of multichannel harmonic signals in noise," *Math. Contr., Signals, Syst.*, vol. 3, pp. 31-43, Jan. 1990.
- [5] L. Y. Geronimus, "Polynomials orthogonal on a circle and their application," *Zapiski Nauchno-issledovatel' skogo in-ta matematiki i mekhaniki i KhMO*, vol. 19, p. 35, 1948.
- [6] —, *Orthogonal Polynomials*, New York: Consultants Bureau, 1961.
- [7] P. R. Halmos, *Finite Dimensional Vector Spaces*. New York: Van Nostrand, 1974.
- [8] R. T. Lacoss, "Data adaptive spectral analysis methods," *Geophysics*, vol. 36, pp. 661-675, 1971.
- [9] L. S. Marple, Jr., *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [10] A. L. Sanovich and I. M. Spitkovskii, "On block Toeplitz matrices and properties of the Gauss model on the half-line related to them," *Teoret. Mat. Fiz.*, vol. 63, pp. 154-160, 1985 (in Russian).
- [11] V. S. Vladimirov and I. V. Volovich, "The diophantine moment problem, orthogonal polynomials, and some models of statistical physics," in *Lecture Notes in Mathematics*, vol. 1043. New York: Springer-Verlag, 1984, pp. 289-292.
- [12] —, "On the model of statistical physics," *Teoret. Mat. Fiz.*, vol. 54, no. 1, pp. 8-22, 1983.

## Gambling Using a Finite State Machine

Meir Feder

**Abstract**—Sequential gambling schemes in which the amount wagered on the future outcome is determined by a finite state (FS) machine are defined and analyzed. It is assumed that the FS machine determines the fraction of the capital wagered at each time instance  $i$  on the outcome at the next time instance,  $i+1$ , and that wagers are paid at even odds. The maximal capital achieved by any FS machine is found and its dependence on an empirical entropy measure,  $H^{FS}(x)$ , defined as the finite state complexity of  $x$  is shown. A specific gambling scheme is then proposed based on the Lempel-Ziv method for universal compression. The capital gained by this method is found and observed that, asymptotically, its exponential growth rate dominates the exponential growth rate achieved by gambling using any FS machine. Furthermore, this specific scheme suggests a class of gambling methods, based on a class of variable-to-variable length (VV) lossless compression methods, in which the capital is doubled for every bit compressed. These results emphasize the relation between gambling and data compression.

**Index Terms**—Gambling, finite state machine, data compression, Lempel-Ziv algorithm, prediction.

#### I. INTRODUCTION

Gambling and information measures are strongly related, as observed in [3]. Suppose that an outcome sequence is generated by a known probabilistic source and that the goal in sequential gambling is to maximize the expected logarithm of the capital. The optimal gambling procedure, investigated further in [4], wagers on the event that the next outcome  $x_{i+1}$  will be  $a$ , after observing  $x_1 \cdots x_i$ , a fraction  $b_{i+1}(a)$  of the capital, where

$$b_{i+1}(a) = \Pr\{x_{i+1} = a | x_i, x_{i-1}, \dots, x_1\}. \quad (1)$$

When wagers are paid at fair odds, the optimal expected logarithm of the capital satisfies

$$\frac{1}{n} E \left\{ \log \frac{S_n}{S_0} \right\} = \frac{1}{n} H(X_1, \dots, X_n) \rightarrow \mathcal{H}, \quad (2)$$

where  $H(X_1, \dots, X_n)$  is the joint entropy of the  $n$  outcomes of the source and  $\mathcal{H}$  is its entropy rate. We refer to [5], Ch. 4 for additional discussion about this relation.

Universal sequential gambling schemes which consider each specific (individual) outcome sequence and thus try to optimize the yield in gambling over that sequence have been analyzed in [1], where a relation between the achievable capital gain of any gambling scheme and the algorithmic complexity of the specific outcome sequence has been established. By algorithmic complexity we refer to the complexity measures of Kolmogorov [6], Chaitin [7], and Solomonoff [8], which are related to each other. The universal scheme in [1], for the binary case, wagers on the event that  $x_{i+1}$  will be "0," after observing  $x_1 \cdots x_i$ , a fraction

Manuscript received October 16, 1989; revised January 27, 1991. This work was presented in part at the Fourth Swedish-Soviet International Workshop on Information Theory, Gotland, Sweden, August 27-September 1, 1989 and at the IEEE International Symposium on Information Theory, San Diego, CA, January 14-19, 1990.

The author is with the Department of Electrical Engineering-Systems, Faculty of Engineering, Tel-Aviv University, Tel-Aviv, 69978, Israel.

IEEE Log Number 9100491.

of the capital  $b_{i+1}$ , such that

$$b_{i+1} = \frac{2^{-l(x_1 \cdots x_i, 0)}}{2^{-l(x_1 \cdots x_i, 0)} + 2^{-l(x_1 \cdots x_i, 1)}} = \frac{2^{-l(x_1 \cdots x_i, 0)}}{2^{-l(x_1 \cdots x_i)}}, \quad (3)$$

where  $x_1 \cdots x_i, 0$  is the sequence  $x_1 \cdots x_i$  extended by 0, and  $l(\nu)$  is the shortest codelength needed to represent the string  $\nu$ . This result is consistent with Solomonoff's probability estimate of a binary string  $\nu$  as  $p(\nu) = 2^{-l(\nu)} = 2^{-l(\nu, 0)} + 2^{-l(\nu, 1)}$ . When wagers are paid at even odds the capital available after  $n$  bets is

$$S_n = S_0 \cdot 2^{n - l(x_1 \cdots x_n)} = S_0 \cdot 2^{n \cdot [1 - K(x_1 \cdots x_n)]}, \quad (4)$$

where  $S_0$  is the initial capital and  $K(x_1^n) = l(x_1^n)/n$  is the shortest normalized codelength (Kolmogorov's complexity) of  $x_1^n = x_1 \cdots x_n$ .

In this correspondence we analyze universal gambling on each specific sequence in a constrained scheme where the amount wagered on the future outcome of the sequence is determined by a finite state (FS) machine. More precisely, let the first  $n$  outcomes of some experiment be denoted, as before,  $x_1^n = x_1 \cdots x_n$  where, for simplicity, we consider a binary outcome sequence although the results can easily be extended to any finite alphabet. Let  $b_{i+1}$  be the fraction of the capital wagered at time  $i$  on that the  $(i+1)$ st outcome is "0" (of course a fraction  $1 - b_{i+1}$  is wagered on that the outcome is "1"). A fair play is assumed, i.e., wagers are paid at even odds. A gambling procedure, defined by a finite state machine, is a function of the state  $s_i$  and the outcome  $x_i$  at time  $i$ :

$$b_{i+1} = f(s_i, x_i), \quad (5)$$

where the state sequence is generated according to

$$s_{i+1} = g(s_i, x_i). \quad (6)$$

In the first part of the correspondence, we find an explicit expression for the maximal capital achieved by any FS machine. This bound on the performance has the form

$$S_n \leq S_0 \cdot 2^{n[1 - H^{FS}(x_1^n)]}, \quad (7)$$

where  $H^{FS}(x_1^n)$ , which is a measure based on the empirical entropy of  $x_1^n$ , is defined here as the finite state complexity of  $x_1^n$ . Unlike Kolmogorov's complexity, this measure can be decided from the given observation sequence.

An alternative definition for the finite state complexity is provided through the work of Ziv and Lempel, [9], [10] on universal lossless compression. In this work, a universal compression algorithm (LZ algorithm) is presented and its performance is used to determine bounds on the compressibility of the given sequence using a finite state encoder. Intuitively, following the notion of [1], [2], compression is related to the capital gain in gambling. In the second part of the correspondence, we suggest a gambling scheme related to the LZ compression algorithm and calculate the capital gain associated with that scheme. This gambling scheme is based on the interpretation of the LZ algorithm observed in [11], [12]. Using properties of the LZ algorithm it will be shown that the exponential growth rate of the capital exceeds, asymptotically, the capital growth rate achieved by any FS machine. Note that, in addition to its theoretical importance, the previous gambling scheme provides a specific readily implementable sequential gambling scheme which can take advantage of the compressibility of the outcome sequence to achieve capital gain.

Another result, discussed in this correspondence, generalizes the idea used in deriving the gambling method induced by the LZ algorithm and suggests a class of practical gambling methods

based on a class of sequentially adaptive variable-to-variable length (VV) lossless compression methods. In any of these methods the capital is doubled for every bit compressed.

The sequential gambling problem raises, at least implicitly, the sequential prediction problem. The general finite state sequential prediction formula can be written as

$$\hat{x}_{i+1} = p(s_i, x_i), \quad (8)$$

where the state sequence is generated as in (6). The optimal FS prediction is also determined, and the relationship between the finite state complexity and the fraction of prediction errors along the given outcome sequence is investigated.

The general analysis of sequential gambling schemes, the maximal capital gain and the definition of the finite state complexity will be presented in the next section. The proposed gambling procedure, related to the Lempel-Ziv algorithm, its analysis, and the asymptotic behavior of the achievable capital gain of any FS machine will be discussed in Section III. In Section IV, the performance of the general class of gambling schemes based on a class of sequentially adaptive variable-to-variable length compression methods will be analyzed. The FS prediction will be discussed in Section V and conclusions are presented in Section VI.

## II. FINITE STATE GAMBLING: BOUNDS AND GENERAL RESULTS

### A. Finite State Gambling

Let the outcome sequence be a binary sequence as previously defined. Assume that the initial capital is  $S_0$  and define  $x_0$  arbitrarily to be 0. With these definitions the FS sequential gambling is totally determined by the functions  $f(\cdot, \cdot)$  and  $g(\cdot, \cdot)$  of (5) and (6). Note that the range of  $f(\cdot, \cdot)$  is the real interval  $[0, 1]$  while the range of  $g(\cdot, \cdot)$  is the finite state set, which for the case of  $K$  states can be the set  $\{1, \dots, K\}$ .

When the finite state machine used for gambling is given, the capital that we have after  $n$  wagers were made, observing the sequence  $x_1^n$ , is calculated as follows.

*Theorem 1:* Let  $n(s, x)$  be the number of times in the sequence that the state  $s_i = s$  and the corresponding outcome  $x_i = x$ . Let  $n(s, x, 0)$  be the number of times that the next outcome  $x_{i+1} = 0$ . Clearly  $n(s, x, 0) + n(s, x, 1) = n(s, x)$ . Define  $\alpha = n(s, x, 0)/n(s, x)$ . Then, the final capital at time  $n$  is given by,

$$S_n = S_0 \cdot 2^{n[1 - \sum_{s,x} n(s, x) / (n(-\alpha \log f(s, x) - (1-\alpha) \log(1-f(s, x))))]}. \quad (9)$$

*Proof:* Suppose that at time  $i$  the available capital is  $S_i$ . An amount of  $b_{i+1}S_i$  is wagered on "0." Since wagers are paid at even odds the capital available after the outcome  $x_{i+1}$  is known is

$$\begin{aligned} S_{i+1} &= 2b_{i+1}S_i, & \text{if } x_{i+1} = "0," \\ S_{i+1} &= 2(1-b_{i+1})S_i, & \text{if } x_{i+1} = "1," \end{aligned} \quad (10)$$

or  $S_{i+1} = 2S_i\delta(b_{i+1}, x_{i+1})$ , where  $\delta(b_{i+1}, x_{i+1}) = b_{i+1}$  if  $x_{i+1} = "0,"$   $\delta(b_{i+1}, x_{i+1}) = 1 - b_{i+1}$  if  $x_{i+1} = "1."$

We can now use (10) recursively and get the final  $S_n$  in terms of the initial capital:

$$S_n = S_0 \prod_{i=0}^{n-1} 2\delta(b_{i+1}, x_{i+1}) = S_0 \cdot 2^{n[1 + 1/n \sum_{i=0}^{n-1} \log \delta(b_{i+1}, x_{i+1})]}. \quad (11)$$

Arranging the summation in the exponent according to the states and recalling that  $b_{i+1} = f(s_i, x_i)$  leads to (9).  $\square$

The interesting problem, of course, is to find the maximal capital gain in this context of gambling using finite state machines. This requires the maximization of (9) with respect to  $f(\cdot, \cdot)$ ,  $g(\cdot, \cdot)$ , and the initial state. While it is hard to solve analytically, this general maximization problem, determining the wagering function  $f(\cdot, \cdot)$ , is an easy task given the state sequence.

**Theorem 2:** The finite state wagering function that yields the maximal capital gain for a given state sequence and a specific outcome sequence  $x_1^n$ , is given by

$$f(s, x) = \alpha(s, x) = \frac{n(s, x, 0)}{n(s, x)}. \quad (12)$$

The capital obtained after  $n$  wagers were made with this scheme is

$$S_n = S_0 \cdot 2^{n[1 - \sum_{s,x} n(s,x)/n \cdot h(\alpha(s,x))]}, \quad (13)$$

where  $h(\alpha) = -\alpha \log \alpha - (1-\alpha) \log(1-\alpha)$  is the binary entropy of the fraction  $\alpha$ .

*Proof:* In order to maximize the capital gain we have to minimize the summation in the exponent of (9). Since we assume that the state sequence is given,  $n(s, x)$  and  $\alpha(s, x)$  are known for the specific outcome sequence. Since  $\alpha$  can have a different value for each  $s$  and  $x$ , we can minimize each term in the summation separately, and get each optimal value of  $f(\cdot, \cdot)$ . Using Jensen's inequality (or directly),

$$\arg \min_f [-\alpha \log f - (1-\alpha) \log(1-f)] = \alpha \quad (14)$$

and the value of the minimum is just the binary entropy  $h(\alpha)$ .  $\square$

It is important to note that although the optimal gambling scheme derived in Theorem 2 above is sequential, as required, it is not found sequentially. The gambling specification requires the knowledge of quantities like  $n(s, x, j)$ , which depend on the entire sequence. This observation emphasizes the importance of the constructive result later in the correspondence, which provide a specific, nonanticipating, sequential gambling procedure.

### B. Finite State Complexity

Using the previous result (2) in analogy to the results in [1], we can define a state dependent complexity measure of the specific outcome sequence  $x_1^n$  based on its conditional empirical entropy with respect to the state sequence. This measure will be a function of the transition rule  $g(\cdot, \cdot)$  and the initial state  $s_0$ , and will be denoted as

$$H_{g, s_0}^{\text{FS}}(x_1^n) = - \sum_{s,x} \frac{N(s,x)}{n} \sum_{j=0,1} \frac{n(s,x,j)}{n(s,x)} \log \frac{n(s,x,j)}{n(s,x)}. \quad (15)$$

We then define the finite state (actually the  $K$ -state) complexity measure of the given outcome sequence as

$$H^{\text{FS}, K}(x_1^n) = \min_{g \in G_{K, s_0}} H_{g, s_0}^{\text{FS}}(x_1^n), \quad (16)$$

where  $G_K$  is the set of all  $K^{2K}$  transition functions that correspond to FS machines with  $K$  states. This minimization can always be accomplished since we have to search over a finite set. Clearly, the capital in any  $K$ -state sequential gambling scheme,

after  $n$  bets, is upper bounded by

$$S_n \leq S_0 \cdot 2^{n \cdot [1 - H^{\text{FS}, K}(x_1^n)]}. \quad (17)$$

Define now the following limit supremum:

$$H^{\text{FS}, K}(\mathbf{x}) = \limsup_{n \rightarrow \infty} H^{\text{FS}, K}(x_1^n), \quad (18)$$

where  $\mathbf{x} = x_1 \cdots x_n \cdots$  is an infinite sequence. The limit supremum (18) provides an upper bound for the asymptotic growth rate of the capital as the length of the outcome sequence becomes large, i.e.,

$$\begin{aligned} \liminf_{n \rightarrow \infty} \frac{1}{n} \log_2 \frac{S_n}{S_0} &\leq 1 - \limsup_{n \rightarrow \infty} H^{\text{FS}, K}(x_1^n) \\ &= 1 - H^{\text{FS}, K}(\mathbf{x}) \doteq r_{\text{FS}, K}(\mathbf{x}). \end{aligned} \quad (19)$$

This growth rate depends on the number of states,  $K$ . As in [9] we consider the asymptotic performance as the number of states goes to infinity, and we define the finite state complexity as

$$\begin{aligned} H^{\text{FS}}(\mathbf{x}) &= \lim_{K \rightarrow \infty} \left[ \limsup_{n \rightarrow \infty} H^{\text{FS}, K}(x_1^n) \right] \\ &= 1 - \lim_{K \rightarrow \infty} r_{\text{FS}, K}(\mathbf{x}) = 1 - r_{\text{FS}}(\mathbf{x}), \end{aligned} \quad (20)$$

where the limit for  $K$  always exists since the empirical entropy, for each  $n$  and thus for its limit supremum, monotonically decreases with  $K$ .

Following the relation between the gambling yield and the complexity of the outcome sequence, as discussed in [1], the definition, (20), of the asymptotic minimum of the state-dependent conditional empirical entropy as the finite state complexity of  $\mathbf{x}$  is motivated. The similarities to the definition of the compression ratios in [9] are also noted.

### III. GAMBLING USING THE LEMPEL-ZIV COMPRESSION ALGORITHM

The universal data compression algorithm suggested by Lempel and Ziv encodes variable length source strings according to a dictionary that contains past strings of the given sequence. The dictionary is updated sequentially according to the source symbols that have just been encoded. During the encoding process the input string is parsed into substrings. The complexity measure of a given sequence, as defined by Ziv and Lempel, is based on the number of parsed substrings, denoted  $c(x_1^n)$ , in the given sequence,  $x_1^n$ . A more detailed description of the Lempel-Ziv algorithm can be found in [9].

In an interpretation of this procedure observed in [11], [12], the parsing is performed according to a sequence of dictionaries, each satisfies the prefix condition and thus can be represented by leaves of a tree. This representation of variable-to-block coding methods was first suggested in [13]. Now in the LZ method, for binary strings, the initial tree is  $\{0, 1\}$ , i.e., a tree containing a root and two leaves. At each step we use the current dictionary to parse a string by following the path from the root to the leaf that corresponds to the source symbols. After a string is parsed, a new dictionary is generated by splitting the leaf that corresponds to the string  $\nu$  just parsed, making it an internal node and adding to the dictionary the two words  $\{\nu 0, \nu 1\}$  corresponding to the leaves descending from it.

As an example, the sequence  $00101010100 \cdots$  is parsed into  $\{0, 01, 010, 1, 0100, \cdots\}$ . The first few dictionaries, represented by trees, generated by this sequence are shown in Fig. 1. We

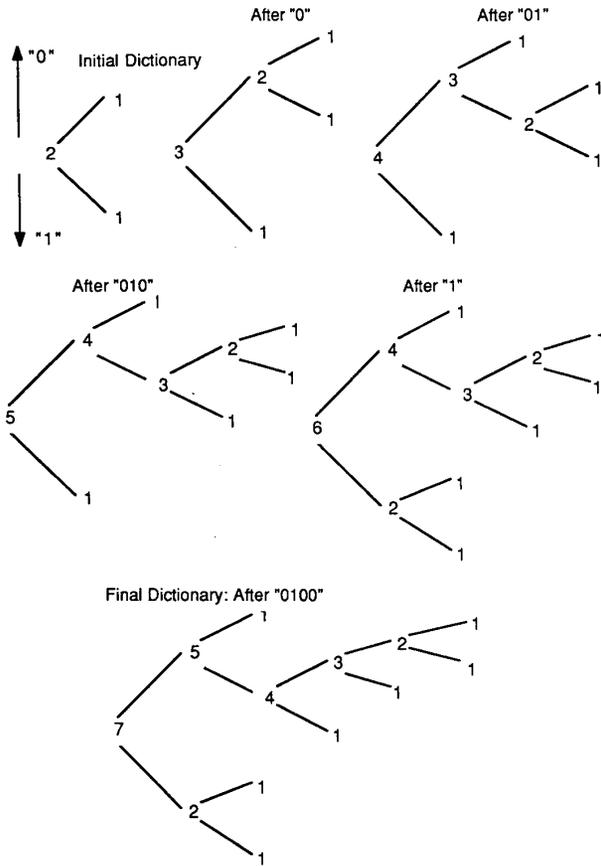


Fig. 1. Sequence of Lempel-Ziv dictionary trees.

associate to each node in the dictionary trees a weight that is the number of leaves that belong to the subtree whose root is this node. Thus, all the leaves in the tree get the number 1, the root gets the total number of leaves, and the weight associated with each node is the sum of the weights associated with all its descendants.

With this interpretation, the following gambling scheme is proposed. As previously described, the source symbols define paths in the dictionary trees; thus following  $\nu = x_1 \cdots x_i$  we reach some node in a dictionary tree. Each possible new outcome,  $x_{i+1}$ , determines a specific direction in the dictionary tree to continue the path. The amount  $b_{i+1}$  wagered on the event that, say, this outcome, is "0" will be the ratio between the weight of the child node that correspond to " $\nu 0$ " and the weight associated with the current node. After the bet is made and  $x_{i+1}$  is known, the pointer moves to the corresponding child node, or, if this child node is a leaf, the tree is extended according to the LZ algorithm and the pointer moves to the root (in the last case the outcome  $x_{i+1}$  is the last symbol of a parsed substring).

This specific gambling scheme can be analyzed explicitly.

**Theorem 3:** The capital after  $n$  bets using the gambling scheme satisfies

$$S_n \geq S_0 \frac{2^n}{(c(x_1^n) + 1)!} \geq S_0 \cdot 2^{n \cdot [1 - 1/(nc(x_1^n) \log(c(x_1^n) + 1))]} \quad (21)$$

*Proof:* Suppose that at the beginning of the  $j$ th parsed string, after  $i$  source symbols, the available capital is  $S_i$ . Suppose that the LZ algorithm parses now a new string of length  $l_j$ . Also assume that the current ( $j$ th) dictionary size (which is the number assigned to the root node of the dictionary tree) is  $k_j$ . Following the gambling method, the capital at the beginning of the next substring,  $S_{i+l_j}$ , is

$$S_{i+l_j} = S_i \frac{2^{l_j}}{k_j} \quad (22)$$

since it is the product of  $l_j$  terms of the form  $2b_m$ ,  $m = i + 1, \dots, i + l_j$ , and the product of the weights,  $\{b_m\}$ , along the path is a telescopic product, which equals  $1/k_j$  independently of the particular path.

The initial dictionary size,  $k_1$ , is 2; after gambling on each parsed string it increases by 1, i.e.,  $k_j = j + 1$ . Applying (22) recursively (and ignoring end effect, i.e., assuming that the outcome sequence ends with a parsed substring), the capital at the end of the outcome sequence can be expressed as

$$S_n = S_0 \prod_{j=1}^{c(x_1^n)} \frac{2^{l_j}}{k_j} = S_0 \frac{2^n}{(2 \cdot 3 \cdot \dots \cdot (c(x_1^n) + 1))} \quad (23)$$

Since  $(c(x_1^n) + 1)! \leq (c(x_1^n) + 1)^{c(x_1^n)}$ , (21) is readily proved.  $\square$

As we recall, in the previous section an upper bound on the gambling yield, using any FS machine, was derived. This upper bound was not explicit; it was in terms of a minimum over all state sequences, see (16), and it was not related to the LZ complexity. We can now use a property of the LZ parsing, referred to as Ziv's inequality, that provides a lower bound for the finite state complexity, (15), and thus an upper bound on the finite state gambling yield, in terms of the quantity  $c(x_1^n)$ . This bound holds for any state sequence and thus it will also be a bound for (16), for all finite  $K$ . This result is summarized in the following theorem (originally shown in [14]) presented here in the context of gambling.

**Theorem 4:** Let the number of states in a finite state machine be  $K$ . For any  $n$  and any finite sequence  $x_1^n = x_1 \cdots x_n$ , the finite state complexity,  $H^{\text{FS},K}(x_1^n)$  is lower-bounded by

$$H^{\text{FS},K}(x_1^n) \geq \frac{1}{n} c(x_1^n) \log c(x_1^n) - \delta(c(x_1^n), K, n) \quad (24)$$

and thus the capital achieved by the gambling scheme based on the LZ algorithm, after  $n$  bets, satisfies

$$S_n \geq S_0 2^{n[1 - H^{\text{FS},K}(x_1^n) - \delta(c(x_1^n), K, n)]} \quad (25)$$

where, for a fixed  $K$ ,  $\delta(c(x_1^n), K, n) \rightarrow 0$  and decays uniformly for all  $x_1^n$  as  $O(\log \log n / \log n)$ .

The proof of this theorem is provided in [15] and [14]. A simpler and more intuitive proof, based on the interpretation of [11], can also be constructed as shown in [16].

Summarizing the results of the previous theorems we see that first, from Theorem 5, for each finite sequence length,  $n > 0$ , there exists a finite state gambling scheme of  $K(n)$  states, the scheme based on the Lempel-Ziv algorithm, for which, the exponential growth rate,  $r_{\text{LZ}}(x_1^n)$  is at least  $1 - n^{-1} c(x_1^n) \log [c(x_1^n) + 1]$ . Second, from Theorem 6, when we compare this scheme to any scheme with a finite, fixed, number of states, then  $r_{\text{LZ}}(x_1^n) \geq r_{\text{FS},K}(x_1^n) - \delta(c(x_1^n), K, n)$ . As we let  $n$

go to infinity, we get

$$\begin{aligned} r_{LZ}(x) &\geq 1 - \limsup_{n \rightarrow \infty} H^{FS,K}(x_1^n) - \lim_{n \rightarrow \infty} \delta(\cdot, \cdot, n) \\ &= 1 - H^{FS,K}(x) = r_{FS,K}(x), \end{aligned} \quad (26)$$

where  $x = x_1, \dots, x_n, \dots$  is now an infinite sequence. This is true for any finite  $K$ . Letting  $K$  go to infinity in (26), we get (since the left-hand side is independent of  $K$ )

$$r_{LZ}(x) \geq 1 - \lim_{K \rightarrow \infty} H^{FS,K}(x) = r_{FS}(x), \quad (27)$$

where  $r_{FS}(x) = 1 - H^{FS}(x)$  and  $H^{FS}(x)$  is defined in (20).

Still, one can say that the domination of the scheme based on the Lempel–Ziv compression is due to an unfair comparison between an infinite state machine and a finite state one. To make this comparison fair, we investigate a “blocked” scheme based on the Lempel–Ziv algorithm in which the dictionary construction process is initialized at the beginning of each block. When the block size is  $n$ , this blocked scheme has a finite number of states, even when we gamble on the infinite sequence  $x$ . The growth rate of the blocked scheme will be denoted  $r_{LZ}(x, n)$ , and in the following theorem it is compared to the finite state exponential growth rate,  $r_{FS}(x)$ .

*Theorem 5:* For any infinite outcome sequence  $x = x_1 \dots x_n \dots$  and any  $\epsilon > 0$ , there exists a finite  $K(\epsilon)$ , such that the rate of the blocked LZ scheme satisfies

$$r_{LZ}(x, n) \geq r_{FS}(x) - \delta_\epsilon(K(\epsilon), n), \quad (28)$$

where  $\lim_{n \rightarrow \infty} \delta_\epsilon(K(\epsilon), n) = \epsilon$ .

*Proof:* The exponential capital growth of the blocked LZ gambling method is, by definition,

$$r_{LZ}(x, n) = \liminf_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \left( 1 - \frac{1}{n} \log [c(x^i) + 1] \right), \quad (29)$$

where  $x^i$  is the  $i$ th block. Now from Theorem 6,

$$1 - \frac{1}{n} \log [c(x^i) + 1] \geq 1 - H^{FS,K}(x^i) - \delta(K, n). \quad (30)$$

Also,

$$H^{FS,K}(x^1 \dots x^k) \geq \frac{1}{k} \sum_{i=1}^k H^{FS,K}(x^i), \quad (31)$$

where  $x^1 \dots x^k$  is the concatenation of the  $k$  blocks, and (31) is true since, by definition, each term in the RHS is minimized separately. Thus, combining (20) and (31), we get

$$\begin{aligned} r_{LZ}(x, n) &\geq 1 - \limsup_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k H^{FS,K}(x^i) - \delta(K, n) \\ &\geq 1 - \limsup_{n \rightarrow \infty} H^{FS,K}(x) - \delta(K, n). \end{aligned} \quad (32)$$

Now from the definition, (20), for any  $\epsilon > 0$  there exists  $K(\epsilon)$  such that  $H^{FS,K}(x) \leq H^{FS}(x) + \epsilon$ . Thus, we get

$$\begin{aligned} r_{LZ}(x, n) &\geq 1 - H^{FS}(x) - \epsilon - \delta(K(\epsilon), n) \\ &= r_{FS}(x) - \delta_\epsilon(K(\epsilon), n), \end{aligned} \quad (33)$$

where  $\delta_\epsilon(K(\epsilon), n) = \epsilon + \delta(K(\epsilon), n)$ , whose limit as  $n$  goes to infinity is  $\epsilon$ .  $\square$

We note the similarity between this theorem and Theorem 2 in [9]. We also recall that the entropy estimate of the Lempel–Ziv compression algorithm, for each sequence, is  $H^{LZ}(x) = n^{-1}c(x) \log c(x)$ . Thus, the results of this section state that, asymptotically, the optimal but yet achievable finite state sequential gambling scheme provides an increase of the capital by a factor of  $\exp_2(1 - H^{LZ}(x))$  per each bet, an intuitively appealing result.

As a final note we can also use an additional property of the LZ data compression algorithm that states that if the outcome sequence is generated by a stationary ergodic source then  $n^{-1}c(x) \log c(x) \rightarrow \mathcal{H}$  with probability 1. Thus, when we use the universal gambling scheme to gamble on the outcome of an ergodic source we achieve, with probability 1, the optimal exponential growth rate without having to know in advance the source’s probability model.

#### IV. GAMBLING USING A CLASS OF VARIABLE-TO-VARIABLE LENGTH (VV) COMPRESSION METHODS

The gambling scheme based on the LZ compression algorithm just presented was very useful in determining the asymptotic performance of finite state gambling schemes. However, its main importance lies, maybe, in the fact that it provides a specific, readily implementable, gambling procedure whose performance was analyzed and shown to have desired properties. This analysis can be extended to a class of variable-to-variable length, lossless, compression methods. In this section, we describe the class and show that, for each of these schemes, the capital is doubled for every bit compressed.

Variable-to-variable length (VV) source coding can be achieved by encoding source words, of variable size, using code words of variable size. The encoded source words are listed in a dictionary while the codewords are listed in a codebook. In the class of VV source coding methods, considered here, the source word dictionary is “complete,” i.e., the lengths of source words in the dictionary satisfy Kraft’s inequality, with equality. We also allow “adaptive” coding in which the dictionary and the codebook can vary after each source word is encoded in a sequential fashion. An example of a dictionary tree and a codebook for encoding the dictionary entries is shown in Fig. 2.

The gambling procedure based on any VV compression method from the class previously described and its performance, is presented as follows.

*Theorem 6:* For each VV compression method, as described above, there exist a sequential gambling scheme such that if a source sequence of length  $n$  is encoded by a sequence of codewords of total size  $l$ , the capital  $S_n$  after gambling on the outcome of the source sequence satisfies

$$S_n \geq S_0 \cdot 2^{n-l}, \quad (34)$$

where  $S_0$  is the initial capital.

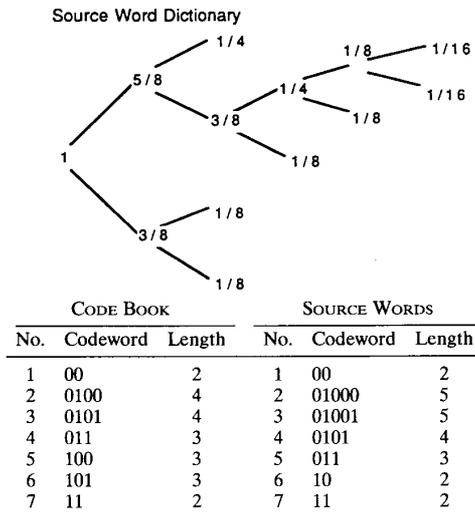


Fig. 2. Dictionary tree and codebook of a VV compression.

*Proof:* Consider the tree that represents the dictionary of source words, and assign a weight of  $2^{-l_c}$  to each leaf where  $l_c$  is the length of the codeword that encodes the source word represented by this leaf. Assign to each node a weight equal to the sum of the weights assigned to its children. Now, when gambling on the outcome of a specific source word, consider the path in the tree that corresponds to that word. The amount wagered on each bit is in proportion to the relative weights in the tree along the path, similar to the procedure used in the LZ gambling. The multiplication of the weights along each path  $s$  in the source dictionary, leading from the root to some leaf, is a telescopic product and it is exactly  $2^{-l_c(s)}/\sum_s 2^{-l_c(s)}$  where  $l_c(s)$  is the codeword length associated with the leaf that ends the path  $s$ . The capital gain when gambling on a string that corresponds to this path  $s$  is

$$2^{l_s} \cdot \frac{2^{-l_c(s)}}{\sum_s 2^{-l_c(s)}}, \quad (35)$$

where  $\sum_s 2^{-l_c(s)}$  is the root weight that is the sum of the weights of all the leaves. Since  $\sum_s 2^{-l_c(s)} \leq 1$  due to Kraft's inequality (for the codebook), the capital gain is lower bounded by  $2^{l_s - l_c}$ .

Now, the sum of the lengths of the source words is the total source length  $n$ . The sum of the lengths of the codewords is the total code length  $l$ . The total capital gain is a multiplication of terms like (35); noticing that this multiplication results in a summation in the exponent, the total capital gain is lower bounded by  $2^{n-l}$ .  $\square$

We note that variable source length compression algorithms can also be based on dictionaries that do not satisfy the prefix condition. The tree representing the source strings may not be complete; in this case some source words are represented by internal nodes and not by leaves. It is easy to extend the result above to this case by assigning to an internal node that is also a source dictionary word, the sum of the weights assigned to its descendants plus a weight  $2^{-l_c}$  where  $l_c$  is the length of the codeword that encodes this source word. We note that this extension may be useful for gambling using, say, Welch's modification of the Lempel-Ziv algorithm [17], which leads to non-complete dictionary trees.

## V. FINITE STATE PREDICTION

Gambling involves a notion of prediction. However, in the gambling problem we end up estimating the probability of the next outcome without committing ourselves to a specific guess. It is thus natural to investigate explicitly the universal finite state prediction problem, where, as defined by (8), the goal for a good prediction is to minimize the number of prediction errors. The relation between FS gambling and FS prediction is also an interesting question.

We now find the optimal FS prediction when the state sequence is given.

*Theorem 7:* Given the state sequence (i.e., given  $g(\cdot, \cdot)$  and  $s_0$ ), the prediction function  $p(\cdot, \cdot)$ , which generates the smallest number of prediction errors in a given outcome sequence  $x_1^n$  is

$$\hat{x}_{i+1} = p(s_i, x_i) = \begin{cases} \text{"0,"} & \text{if } n(s_i, x_i, 0) > n(s_i, x_i, 1), \\ \text{"1,"} & \text{otherwise.} \end{cases} \quad (36)$$

*Proof:* The total number of prediction errors  $n_e$  out of  $n$  (the size of the outcome sequence) can be written as

$$n_e = n - n_c = n - \sum_{s,x} n(s, x, \hat{x}) = n - \sum_{s,x} n(s, x, p(s, x)), \quad (37)$$

where  $n_c$  is the number of correct predictions. Since the prediction is a function of the current state and current outcome, we can minimize each term in the sum separately and get (36).  $\square$

The finite-state prediction rule (36) assumes that the state sequence is given. To get the optimal FS prediction it is also required to find the best, in terms of minimum prediction errors, state sequence. It is hard to find this minimum analytically; however, the minimum number of prediction errors, or its complement the maximum number of correct predictions, can be bounded in terms of the optimal gambling performance, i.e., in terms of the finite state empirical entropy,  $H^{\text{FS},K}(x_1^n)$ . This relation is presented in the following theorem.

*Theorem 8:* for a specific outcome sequence  $x_1^n = x_1, \dots, x_n$ , of size  $n$ , the maximum number of correct predictions over  $x_1^n$  of any finite state predictor with  $K$  states is bounded by

$$n_c^{\text{FS},K}(x_1^n) \leq h^{-1}(H^{\text{FS},K}(x_1^n)) \cdot n, \quad (38)$$

where  $h^{-1}(x)$ ,  $1/2 \leq x \leq 1$ , is the inverse of the binary entropy function.

*Proof:* Let  $g$  and  $s$  define a state sequence of a FS machine used for prediction and denote by  $n_c = \sum_{s,x} n(s, x, \hat{x})$  the number of correct guesses when this state sequence together with (36) is used for prediction. Because the binary entropy function is concave, we can write

$$\begin{aligned} H_{g,s_0}^{\text{FS}}(x_1^n) &= \sum_{s,x} \frac{n(s,x)}{n} h\left(\frac{n(s,x,\hat{x})}{n(s,x)}\right) \\ &\leq h\left(\sum_{s,x} \frac{n(s,x)}{n} \frac{n(s,x,\hat{x})}{n(s,x)}\right) \\ &= h\left(\frac{n_c}{n}\right). \end{aligned} \quad (39)$$

Now,  $1/2 \leq n_c/n \leq 1$ , and since  $h(x)$  is monotonically decreas-

ing for  $1/2 \leq x \leq 1$ , we get

$$n_c \leq h^{-1}(H_{g',s'}^{FS}(x_1^n)). \quad (40)$$

Let  $g'$  and  $s'$  be the state sequence generator and the initial state of the  $K$ -state machine that leads, together with (36), to the maximum number of correct predictions,  $n_c^{FS,K}$ . Let  $H_{g',s'}^{FS}(x_1^n)$  be the empirical entropy associated with this state sequence. From the definition, (16),  $H^{FS,K}(x_1^n) \leq H_{g',s'}^{FS}(x_1^n)$ . Using this, the monotonicity of  $h^{-1}(\cdot)$ , and (40), we get

$$n_c^{FS,K} \leq h^{-1}(H_{g',s'}^{FS}(x_1^n)) \cdot n \leq h^{-1}(H^{FS,K}(x_1^n)) \cdot n. \quad \square \quad (41)$$

In general, the optimal state sequence for gambling, i.e., the state sequence that achieves  $H^{FS,K}(x_1^n)$ , may not be the state sequence that minimizes the number of prediction errors. Suppose we use the state sequence that corresponds to the optimal  $K$ -state machine for gambling, (i.e., the state sequence that is found in the minimization of (16)) together with the prediction function (36) to make predictions and let  $\tilde{n}_c^{FS,K}$  be the number of correct prediction associated with this prediction scheme. Define  $\tilde{p}^{FS,K} = \tilde{n}_c^{FS,K}/n$  and  $p^{FS,K} = n_c^{FS,K}/n$  as the fraction of correct predictions associated with the state sequence for gambling and the optimal state sequence for prediction, respectively. By definition,  $\tilde{p}^{FS,K} \leq p^{FS,K}$ , but from (41)

$$\tilde{p}^{FS,K} \leq p^{FS,K} \leq h^{-1}(H^{FS,K}(x_1^n)). \quad (42)$$

Thus, given the optimal state sequence for gambling, we can calculate upper and lower bounds on the optimal fraction of correct predictions. Depending on the diversity of the correct prediction fractions that correspond to each state, these bounds may be loose or tight. Since the number of correct predictions is an integer, then if  $h^{-1}(H^{FS,K}(x_1^n)) \cdot n \leq \tilde{p}^{FS,K} \cdot n + 1$ , we must have  $\tilde{p}^{FS,K} = p^{FS,K}$ , i.e., in this case the optimal state sequence for gambling leads, with (36), to the minimum number of prediction errors and thus it is the optimal state sequence for prediction as well.

We have provided here only a preliminary analysis of the FS prediction problem, where we have focused on the relation between the FS prediction and the FS gambling. The FS prediction problem is investigated in depth in [16].

## VI. SUMMARY AND CONCLUSION

Sequential gambling using a finite state machine has been defined and analyzed and the optimal achievable capital gain has been obtained. The exponential growth rate of the capital, i.e.,  $1/n \log S_n/S_0$ , has been considered. The quantities given by (15)–(20), which are based on the empirical conditional entropy of the outcome sequence, have been used to define the finite state complexity  $H^{FS}$  and, as expected, it was shown that the exponential growth rate of the capital has the form of  $1 - H^{FS}$ .

A specific sequential gambling scheme based on the Lempel–Ziv compression algorithm has been presented. The explicit determination of the probability estimate induced by the Lempel–Ziv compression algorithm, as noted in [11] and [12], is emphasized by the results presented here. The exponential growth rate of the capital has, asymptotically, the form  $1 - n^{-1}c(x) \log c(x)$ , where  $c(x)$  is the number of distinct phrases, generated by the Lempel–Ziv parsing algorithm. By utilizing properties of the Lempel–Ziv compression (Ziv's inequality) it was shown that the exponential growth rate of this gambling scheme dominates the growth rate of any finite state gambling scheme where the number of states is fixed. Also, using addi-

tional properties of the LZ data compression scheme, if the outcome sequence is generated by an ergodic source, then the universal gambling scheme presented here achieves, with probability 1, the optimal capital growth rate, despite the fact that the source probabilistic model is not available.

Another result presented in the correspondence is a method for sequential gambling that can be based on any compression algorithm from a class of variable-to-variable length lossless compression algorithms. This method will double the capital for every bit compressed but it will also reduce the capital by half for every extra bit. The gambling method also reveals the probability estimate, in the sense of Solomonoff, induced by each of these compression algorithms. In fact the results here confirm and extend the work reported in [1], [2], [18], and elsewhere on the relation between gambling and data compression.

Several topics related to the work presented here call for further investigation. It will be interesting to achieve an explicit expression for the minimization of (16), and thus for the finite state complexity, as defined here, of an individual sequence. Also, the finite state prediction problem, introduced here, is currently investigated and it is analyzed in depth in [16].

## REFERENCES

- [1] T. M. Cover, "Universal gambling schemes and the complexity measures of Kolmogorov and Chaitin," Tech. Rep. 12, Dept. of Statistics, Stanford Univ., Stanford, CA, Oct. 1974.
- [2] T. M. Cover and R. King, "A convergent gambling estimate of the entropy of English," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 4, pp. 413–421, July 1978.
- [3] J. L. Kelly Jr., "A new interpretation of information rate," *Bell Syst. Tech. J.*, vol. 35, pp. 917–926, 1956.
- [4] L. Breiman, "Optimal gambling systems for favorable games," *Proc. 4th Berkeley Symp. Math. Stat. Prob.*, vol. 1, 1961 pp. 65–78.
- [5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, to be published, 1991.
- [6] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Inform. Trans.*, vol. 1, pp. 4–7, 1965.
- [7] G. J. Chaitin, "A theory of program size formally identical to information theory," *J. ACM*, vol. 22, pp. 329–340, 1975.
- [8] R. J. Solomonoff, "A formal theory of inductive inference, Parts 1 and 2," *Inform. Contr.*, vol. 7, pp. 1–22 and 224–254, 1964.
- [9] J. Ziv and A. Lempel, "Compression of individual sequences via variable rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 5, pp. 530–536, Sept. 1978.
- [10] J. Ziv, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 405–412, 1978.
- [11] G. G. Langdon, Jr., "A note on the Lempel–Ziv model for compressing individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 2, pp. 284–287, Mar. 1983.
- [12] J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 5, pp. 656–664, Sept. 1983.
- [13] B. P. Tunstall, "Synthesis of noiseless compression codes," Ph.D. thesis, Georgia Inst. of Technol., Atlanta, GA, 1968.
- [14] E. Plotnik, M. J. Weinberger, and J. Ziv, "Upper bounds on the probability of sequences emitted by finite state sources and the redundancy of the Lempel–Ziv algorithm," submitted to *IEEE Trans. Inform. Theory*.
- [15] N. Merhav, M. Gutman, and J. Ziv, "On the estimation of the order of a Markov chain and universal data compression," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 1014–1019, Sept. 1989.
- [16] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, to be submitted, 1991.
- [17] T. A. Welch, "A technique for high performance data compression," *IEEE Comput. Mag.*, vol. 17, pp. 8–19, 1984.
- [18] P. H. Algoet and T. M. Cover, "Asymptotic optimality and asymptotic equipartition properties of log-optimal investment," *Ann. Probab.*, vol. 16, pp. 876–898, 1988.