technique [8] was generalized to the first public key cryptosystem and is quite secure but computationally expensive. The current work describes a computationally efficient public key cryptosystem.

Recently, Rivest, Shamir, and Adleman [13] have proposed another public key cryptosystem that yields signatures more directly because the density of solutions in their problem is one. Their system also requires a smaller key (apparently 600 bits versus 20 kbits). Neither system's security has been adequately established, but when iterated, the trapdoor knapsack appears less likely to possess a chink in its armor. When used for obtaining signatures the trapdoor knapsack appears to be the weaker of the two. Both public key systems clearly need further certification and study.

## References


[1] E. Horowitz and S. Sahni, "Computing partitions with applications to the knapsack problem," *JACM*, vol 21, no. 2, pp. 277–292, Apr. 1974.

[2] R. Schroeppel, unpublished work.

[3] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.

[4] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, Nov. 1976, pp. 644–654.

[5] O. H. Ibarra and C. E. Kim, "Fast approximation algorithms for the knapsack and sum of subset problems," *JACM*, vol. 22, no. 4, pp. 463–468, Oct. 1975.

[6] E. L. Lawler, "Fast approximation algorithms for knapsack problems," Electronics Research Laboratory, College of Eng. U.C. Berkeley Memorandum UCB/ERL M77/45 21, June 1977.

[7] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF($P$) and its cryptographic significance," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 106–110, Jan. 1978.

[8] R. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299.

[9] M. V. Wilkes, *Time-Sharing Computer Systems*. New York: Elsevier, 1972.

[10] A. Evans, Jr., W. Kantrowitz, and E. Weiss, "A user authentication system not requiring secrecy in the computer," *Commun. ACM*, vol. 17, no. 8, Aug. 1974, pp. 437–442.

[11] G. B. Purdy, "A high security log-in procedure," *Commun. ACM*, vol. 17, no. 8, Aug. 1974, pp. 442–445.

[12] R. Merkle and J. Reeds, unpublished work.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[14] R. M. Karp "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, (1972), pp. 85–104.


# Compression of Individual Sequences via Variable-Rate Coding

JACOB ZIV, FELLOW, IEEE, AND ABRAHAM LEMPEL, MEMBER, IEEE

***Abstract*—Compressibility of individual sequences by the class of generalized finite-state information-lossless encoders is investigated. These encoders can operate in a variable-rate mode as well as a fixed-rate one, and they allow for any finite-state scheme of variable-length-to-variable-length coding. For every individual infinite sequence $x$ a quantity $\rho(x)$ is defined, called the compressibility of $x$, which is shown to be the asymptotically attainable lower bound on the compression ratio that can be achieved for $x$ by any finite-state encoder. This is demonstrated by means of a constructive coding theorem and its converse that, apart from their asymptotic significance, also provide useful performance criteria for finite and practical data-compression tasks. The proposed concept of compressibility is also shown to play a role analogous to that of entropy in classical information theory where one deals with probabilistic ensembles of sequences rather than with individual sequences. While the definition of $\rho(x)$ allows a different machine for each different sequence to be compressed, the constructive coding theorem leads to a universal algorithm that is asymptotically optimal for all sequences.**

Manuscript received June 10, 1977; revised February 20, 1978.

J. Ziv is with Bell Laboratories, Murray Hill, NJ 07974, on leave from the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

A. Lempel is with Sperry Research Center, Sudbury, MA 01776, on leave from the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

## I. Introduction

IN A RECENT paper [1], data-compression coding theorems and their converses were derived for the class of finite-state encoders that map at a fixed rate input strings drawn from a source of $\alpha$ letters into equally long strings over an alphabet of $\beta \leqslant \alpha$ letters. In the context of data-compression, the aim is to minimize the number of bits/symbol $\log_2 \beta$, while securing zero or negligibly small distortion. For every individual infinite sequence $x$, this minimal bit/symbol rate was shown in [1] to be equal to a quantity $H(x)$ that, in analogy with the Shannon entropy (which is defined for probabilistic ensembles rather than

0018-9448/78/0900-0530$00.75 

individual sequences), corresponds to the smallest coding rate for $x$ under which the decoding error-rate can be made arbitrarily small.

In this paper, compressibility of individual sequences is investigated with respect to a broader class of encoders that can operate in a variable-rate mode as well as in a fixed-rate one and that allow for any finite-state scheme of variable-length-to-variable-length coding.[1] On the other hand, no distortion is allowed, and the original data must be fully recoverable from the compressed image. This class of encoders can be modeled by the class of finite-state information-lossless generalized automata [2], [3].

In our model, an encoder $E$ is defined by a quintuple $(S,A,B,g,f)$ where $S$ is a finite set of states, $A$ is a finite input alphabet, $B$ is a finite set of output *words* over a finite output alphabet, $g$ is the "next-state" function that maps $S\times A$ into $S$, and $f$ is the output function that maps $S\times A$ into $B$.

By allowing the words in $B$ to be of different finite lengths, we allow for block-to-variable coding, and by including in $B$ the null-word $\lambda$ (i.e., the "word" of length zero), we allow for any finite-state scheme of variable-to-variable coding.

When an infinite sequence $x=x_1x_2\cdots$, $x_i\in A$ is fed into $E=(S,A,B,g,f)$, the encoder emits an infinite sequence $y=y_1y_2\cdots$, $y_i\in B$ while going through an infinite sequence of states $z=z_1z_2\cdots$, $z_i\in S$, according to

$$y_i=f(z_i,x_i)$$

$$z_{i+1}=g(z_i,x_i), \qquad i=1,2,\cdots$$

where $z_i$ is the state of $E$ when it "sees" the input symbol $x_i$.

A finite segment $x_ix_{i+1}\cdots x_j$, $1\leqslant i\leqslant j$, of $x$ will be denoted by $x_i^j$; similar notation will naturally apply to finite segments of other sequences. Following conventional practice, we shall extend the use of the encoder functions $f$ and $g$ to indicate the output sequence as well as the final state, which results from a given initial state and a finite sequence of input letters. For instance, we shall occasionally write $f(z_1,x_1^n)$ for $y_1^n$ and $g(z_1,x_1^n)$ for $z_{n+1}$.

An encoder $E$ is said to be *information lossless* (IL) if for all $z_1\in S$ and all $x_1^n\in A^n$, $n\geqslant 1$, the triple $\{z_1,f(z_1,x_1^n),g(z_1,x_1^n)\}$ uniquely determines $x_1^n$. $E$ is said to be *information lossless of finite order* (ILF) if there exists a finite positive integer $m$ such that for all $z_1\in S$ and all $x_1^m\in A^m$, the pair $\{z_1,f(z_1,x_1^m)\}$ uniquely determines the first input letter $x_1$. It is easy to verify [3] that if $E$ is ILF then it is also IL, but there exist IL encoders that are not ILF.[2]

[1]We regard block-to-block, block-to-variable, and variable-to-block as special cases of variable-to-variable coding.

[2]Even [3] presented an algorithm for determining whether a given automaton is either IL, ILF, or neither. For an automaton with $s$ states, the algorithm will take a number of steps that are proportional to $s^2$.

In the sequel, we assume the IL or the ILF property depending on which leads to a stronger result. For example, we prove a coding theorem (Theorem 2) by means of an ILF construction, while its converse (Theorem 1) is proved under the broader IL assumption.

To simplify the discussion without any real loss of generality, we also assume throughout that the output alphabet of the encoder is binary and that the initial state $z_1$ is a prescribed fixed member of the state-set $S$.

Given an encoder $E=(S,A,B,g,f)$ and an input string $x_1^n$, the *compression ratio* for $x_1^n$ with respect to $E$ is defined by

$$\rho_E(x_1^n)\triangleq\frac{L(y_1^n)}{n\log_2\alpha} \tag{1}$$

where $\alpha=|A|$, $y_1^n=f(z_1,x_1^n)$, $L(y_1^n)=\sum_{i=1}^n l(y_i)$, and $l(y_i)$ is the length in bits of $y_i\in B$. (Note that when $y_i=\lambda$, $l(y_i)=0$.)

The minimum of $\rho_E(x_1^n)$ over the class $E(s)$ of all finite-state IL encoders with $|A|=\alpha$ and $|S|\leqslant s$ is denoted by $\rho_{E(s)}(x_1^n)$. That is

$$\rho_{E(s)}(x_1^n)\triangleq\min_{E\in E(s)}\{\rho_E(x_1^n)\}. \tag{2}$$

Furthermore let

$$\rho_{E(s)}(x)\triangleq\limsup_{n\to\infty}\rho_{E(s)}(x_1^n) \tag{3}$$

and

$$\rho(x)\triangleq\lim_{s\to\infty}\rho_{E(s)}(x). \tag{4}$$

It is clear that for every individual sequence $x$, $0\leqslant\rho(x)\leqslant 1$. This normalized quantity $\rho(x)$ that depends solely on $x$ will be referred to as the (finite-state) *compressibility* of $x$. In Theorem 1 (the converse-to-coding theorem), we derive a lower bound on $\rho_{E(s)}(x_1^n)$ and show that in the limit this bound approaches the normalized Lempel–Ziv complexity [4] and becomes a lower bound on the compressibility of $x$. In Theorem 2 (the coding theorem), we demonstrate, using a variant of the author's universal compression algorithm [5], the existence of an asymptotically optimal universal ILF encoding scheme under which the compression ratio attained for $x$ tends in the limit to the compressibility $\rho(x)$ of $x$ for every $x$. It is important to note that apart from their asymptotic significance, the results of Theorems 1 and 2 also provide useful performance criteria for finite (and practical) data-compression tasks.

The concept of compressibility as defined here, like the quantity $H(\cdot)$ in [1], seems to play a role analogous to that of entropy in classical information theory where one deals with probabilistic ensembles of sequences rather than with individual sequences. This analogy is reinforced by Theorems 3 and 4, where our concept of compressibility is examined from a probabilistic point of view.

The remainder of this paper contains two parts: descriptive part (Section II) where all the results are stated

and discussed and a formal part (Section III) where all proofs except that of Theorem 2 are given. The proof of Theorem 2, which is constructive and thus informative, is presented in the mainstream of Section II.

## II. Statement and Discussion of Results

Our first result establishes a lower bound on the compression ratio attainable by any encoder $E$, from the class $E(s)$ of IL encoders with no more than $s$ states, for any finite input string $x_1^n$ over an alphabet $A$ of $\alpha$ letters. In the theorem below and elsewhere in the sequel, $\log k$ means $\log_2 k$.

*Theorem 1 (Converse-to-Coding Theorem):* For every $x_1^n \in A^n$

$$\rho_{E(s)}(x_1^n) \geq \frac{c(x_1^n)+s^2}{n \log \alpha} \log \frac{c(x_1^n)+s^2}{4s^2} + \frac{2s^2}{n \log \alpha} \quad (5)$$

where $c(x_1^n)$ is the largest number of *distinct* strings (or "phrases") whose concatenation forms $x_1^n$. (The proof of this theorem is given in Section III.)

It was shown in an earlier paper [4, Th. 2] that for all $x_1^n \in A^n$

$$c_{LZ}(x_1^n) - 1 \leq c(x_1^n) < \frac{n \log \alpha}{(1-\epsilon_n) \log n} \quad (6)$$

where $c_{LZ}(x_1^n)$ is the Lempel–Ziv complexity of $x_1^n$ and $\epsilon_n$ satisfies $\lim_{n\to\infty} \epsilon_n = 0$. From (5) and (6) it follows that

$$\rho_{E(s)}(x) = \limsup_{n\to\infty} \rho_{E(s)}(x_1^n) \geq \limsup_{n\to\infty} \frac{c(x_1^n) \log c(x_1^n)}{n \log \alpha}$$

which implies the bound (7a) of Corollary 1. The bound (7b) of this corollary follows directly from (5) and (6).

*Corollary 1 (Compressibility Bounds):*

$$\rho(x) = \lim_{s\to\infty} \rho_{E(s)}(x) \geq \limsup_{n\to\infty} \frac{c(x_1^n) \log c(x_1^n)}{n \log \alpha} \quad (7a)$$

$$\rho(x) \geq \limsup_{n\to\infty} \limsup_{k\to\infty} \frac{1}{kn \log \alpha} \cdot \sum_{i=0}^{k-1} c\left(x_{in+1}^{(i+1)n}\right) \log c\left(x_{in+1}^{(i+1)n}\right). \quad (7b)$$

It also follows from (6) and (7) that $\lim_{n\to\infty} \sup (c_{LZ}(x_1^n) \log n)/(n \log \alpha)$, the normalized Lempel–Ziv complexity of $x$, serves as a lower bound on the compressibility $\rho(x)$ of $x$.

An interesting application of the compressibility bound is the use of (7) to identify certain infinite sequences $x$ that, while being rather easy to describe, satisfy $\rho(x) = 1$ and thus are incompressible by any finite-state IL encoder. To illustrate this point, let $u(k)$ denote the binary sequence of length $k2^k$ that lists, for example, in lexicographic order, all the $2^k$ binary words of length $k$, and let

$$u_1^{n(k)} = u(1)u(2)\cdots u(k)$$

where

$$n(k) = \sum_{i=1}^{k} i2^i = (k-1)2^{k+1} + 2.$$

It is easy to verify that when each $u(i)$ is parsed into its $2^i$ distinct $i$-tuples, we obtain a parsing of $u_1^{n(k)}$ into a maximum number of distinct phrases, namely,

$$c\left(u_1^{n(k)}\right) = \sum_{i=1}^{k} 2^i = 2^{k+1} - 2.$$

For example, $u_1^{n(3)}$ is parsed as follows:

$$u_1^{n(3)} = 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111. \quad (8)$$

For this particular sequence $u$, inequality (7a) implies

$$\rho(u) \geq \lim_{k\to\infty} \frac{(2^{k+1}-2) \log (2^{k+1}-2)}{(k-1)2^{k+1}+2} = 1.$$

In our next result we employ a variant of the authors' universal compression algorithm [5] to demonstrate the existence of an ILF compression scheme under which, for every $x$, the compression ratio attained for $x_1^n$ tends to $\rho(x)$ as $n$ tends to infinity.

*Theorem 2 (Coding Theorem):* For every $n > 0$ there exists a finite-state ILF encoder $\mathcal{E}$ with $s(n)$ states that implements a block-to-variable code with the following performance characteristics.

i) For any given block length $n$ and every input block $x_1^n$, the compression-ratio attained by $\mathcal{E}$ satisfies

$$\rho_{\mathcal{E}}(x_1^n) \leq \frac{c(x_1^n)+1}{n \log \alpha} \log \left(2\alpha(c(x_1^n)+1)\right); \quad (9)$$

the compression ratio attained for successive blocks $x_{(i-1)n+1}^{in}$, $i = 1, 2, \cdots$, satisfies the same inequality with $x_{(i-1)n+1}^{in}$ replacing $x_1^n$.

ii) For every finite $s$,

$$\rho_{\mathcal{E}}(x_1^n) \leq \rho_{E(s)}(x_1^n) + \delta_s(n) \quad (10)$$

where

$$\lim_{n\to\infty} \delta_s(n) = 0.$$

iii) Given an infinite input sequence $x$, let $\rho_{\mathcal{E}}(x,n)$ denote the compression ratio attained for $x$ by $\mathcal{E}$ while feeding $\mathcal{E}$ with successive input blocks of length $n$. Then for any $\epsilon > 0$

$$\rho_{\mathcal{E}}(x,n) \leq \rho(x) + \delta_\epsilon(x,n) \quad (11)$$

where

$$\lim_{n\to\infty} \delta_\epsilon(x,n) = \epsilon.$$

*Proof:* The proof is constructive, and before going into computational details, we present a short outline of the construction. For the encoder $\mathcal{E}$, we employ an ILF finite-state machine that realizes a concatenated coding scheme by combining a fixed block-to-variable outer code with a state-dependent variable-to-variable inner code. The inner code is used to encode sequentially and state

dependently growing segments of an input block of relatively large length $n$. Upon completion of a block the machine returns to its initial state, thus "forgetting" all past history before starting to encode the next input block.

The segments of a block that serve as input words of the inner code are determined according to a so-called *incremental parsing* procedure. This procedure is sequential, and it creates a new phrase as soon as a prefix of the still unparsed part of the string differs from all preceding phrases. The parsing is indicated as

$$x_1^n = x_{n_0+1}^{n_1} x_{n_1+1}^{n_2} x_{n_2+1}^{n_3} \cdots x_{n_p+1}^{n_{p+1}}, \qquad n_0 \triangleq 0,\ n_{p+1} \triangleq n, \tag{12}$$

and is called incremental if the first $p$ words $x_{n_{j-1}+1}^{n_j}$, $1 \leqslant j \leqslant p$, are all distinct and if for all $j = 1,2,\cdots,p+1$ when $n_j - n_{j-1} > 1$ there exists a positive integer $i < j$ such that $x_{n_{i-1}+1}^{n_i} = x_{n_{j-1}+1}^{n_j - 1}$.

It is clear from this definition that if (12) is an incremental parsing of $x_1^n$, then $n_1 = 1$. The last word $x_{n_p+1}^n$ may or may not be distinct from the first $p$ words, and for every word of length $l > 1$, its prefix of length $l-1$ can be found as an earlier word of the parsing. For example, (8) is an incremental parsing of $u_1^{n(3)}$.

Now let $x_{n_{-1}+1}^{n_0} \triangleq \lambda$, the word of length zero, and (since $x_1^n = \lambda x_1^n$) let us adopt the convention that $x_{n_{-1}+1}^{n_0}$ is always the initial word of an incremental parsing. Also given a word $w$, let $d(w)$ denote the word obtained by deleting the last letter of $w$. It follows that for every $j = 1,2,\cdots,p+1$ there exists a unique nonnegative integer $\pi(j) = i < j$ such that $d(x_{n_{j-1}+1}^{n_j}) = x_{n_{i-1}+1}^{n_i}$.

The incremental parsing of a given block $x_1^n$ and the coding of the words determined by it are executed sequentially as follows. To determine the $j$th word, $1 \leqslant j \leqslant p+1$, we take $n_j$ to be the largest integer, not exceeding $n$, for which $d(x_{n_{j-1}+1}^{n_j})$ equals some earlier word, for example, $x_{n_{i-1}+1}^{n_i}$, and we set $\pi(j) = i$ (e.g., for $j = 1$, $n_1 = 1$, $x_1^1 = x_1$, $d(x_1) = \lambda$, and $\pi(1) = 0$). Having determined $x_{n_{j-1}+1}^{n_j}$, it is encoded into the base-2 expansion of the integer

$$I(x_{n_{j-1}+1}^{n_j}) \triangleq \pi(j)\alpha + I_A(x_{n_j})$$

where $I_A$ is a predetermined mapping from the input alphabet $A$ onto the set of integers 0 through $\alpha - 1$. Since $0 \leqslant \pi(j) \leqslant j-1$, it follows that

$$0 \leqslant I(x_{n_{j-1}+1}^{n_j}) \leqslant (j-1)\alpha + \alpha - 1 = j\alpha - 1,$$

and the number of bits required to encode the $j$th word is $L_j = \lceil \log (j\alpha) \rceil$, the least integer that is not smaller than $\log (j\alpha)$.

It is easy to verify that this code is uniquely decipherable with bounded delay. Given a binary sequence $b = b_1 b_2 \cdots$ that begins with the coded image of $x_1^n$, we can determine $x_1^n$ sequentially by reversing the encoding procedure. Namely, $b$ is parsed into the codewords $b_{k_0+1}^{k_1}, b_{k_1+1}^{k_2}, \cdots$, which are decoded into the original phrases according to the following recursive procedure. Initially, set $j = 0$, $k_j = 0$, and $n_j = 0$. Given the current values of $j$, $k_j$, and $n_j < n$, procede with the following.

i) Set $k_{j+1} = k_j + \lceil \log ((j+1)\alpha) \rceil$.

ii) Take $I(x_{n_j+1}^{n_{j+1}})$ to be the integer whose base-2 representation is given by $b_{k_j+1}^{k_{j+1}}$, and determine the nonnegative integers $i$ and $r$ that satisfy

$$I(x_{n_j+1}^{n_{j+1}}) = i\alpha + r, \qquad 0 \leqslant r \leqslant \alpha - 1.$$

iii) Take $a = I_A^{-1}(r)$, and if $n_j + (n_i - n_{i-1} + 1) \geqslant n$, set $n_{j+1} = n$, take $x_{n_j+1}^n$ to be the word formed by the first $n - n_j$ letters of $x_{n_{j-1}+1}^{n_j} a$, and stop. Otherwise, set $n_{j+1} = n_j + n_i - n_{i-1} + 1$, take $x_{n_j+1}^{n_{j+1}} = x_{n_{i-1}+1}^{n_i} a$, increment $j$, and return to (*i*).

For continuous decoding of successive blocks, the "stop" instruction in iii) should be replaced by the "reset" instruction: set $j$, $k_j$, and $n_j$ to their initial zero value and return to i).

The total number of bits that go into the coding of an input block $x_1^n$ that is parsed incrementally into $p+1$ words is given by

$$L = \sum_{j=1}^{p+1} L_j = \sum_{j=1}^{p+1} \lceil \log (j\alpha) \rceil.$$

Hence

$$L \leqslant \sum_{j=1}^{p+1} \log (2\alpha j) \leqslant (p+1)(\log (p+1) + \log (2\alpha)).$$

Since $p \leqslant c(x_1^n)$, the maximum number of distinct words into which $x_1^n$ can be parsed, it follows that the compression ratio attainable by the described encoder $\mathcal{E}$ for $x_1^n$ satisfies

$$\rho_{\mathcal{E}}(x_1^n) \leqslant \frac{c(x_1^n)+1}{n \log \alpha} \log \left[2\alpha(c(x_1^n)+1)\right],$$

which proves (9).

From (5) and (9) after some manipulation, we obtain

$$\rho_{\mathcal{E}}(x_1^n) \leqslant \rho_{E(s)}(x_1^n) + \frac{\Delta(c)}{n \log \alpha} \tag{13}$$

where $c = c(x_1^n)$ and

$$\Delta(c) = (c+1) \log \frac{2\alpha(c+1)}{c+s^2} - (s^2-1) \log (c+s^2) + (c+s^2) \log (4s^2).$$

It is easy to verify that $\Delta(c)$ increases with $c$, which by (6) implies

$$\delta_s(n) \triangleq \frac{1}{n \log \alpha} \Delta\left(\frac{n \log \alpha}{(1-\epsilon_n) \log n}\right) \geqslant \frac{\Delta(c)}{n \log \alpha}. \tag{14}$$

It is also easy to verify that

$$\lim_{n \to \infty} \delta_s(n) = 0,$$

which together with (13) and (14) proves (10).

Finally, the compression attained by our encoder $\mathcal{E}$ for an infinite sequence $x$ is by definition:

$$\rho_{\mathcal{E}}(x,n) = \limsup_{k \to \infty} \frac{1}{kn \log \alpha} \sum_{i=1}^{k} L_i \tag{15}$$

where $L_i$ is the number of bits used to encode the $i$th block $x_{(i-1)n+1}^{in}$ of $x$. Since (9) and (10) hold for any input block of length $n$, we can write

$$\frac{L_i}{n \log \alpha} = \rho_{\mathcal{E}}(x_{(i-1)n+1}^{in}) \leqslant \rho_{E(s)}(x_{(i-1)n+1}^{in}) + \delta_s(n). \quad (16)$$

From (15) and (16) we obtain

$$\rho_{\mathcal{E}}(x,n) \leqslant \delta_s(n) + \limsup_{k\to\infty} \frac{1}{K} \sum_{i=1}^{k} \rho_{E(s)}(x_{(i-1)n+1}^{in}),$$

which reduces to

$$\rho_{\mathcal{E}}(x,n) \leqslant \delta_s(n) + \rho_{E(s)}(x). \quad (17)$$

where $\lim_{n\to\infty} \delta_s(n) = 0$. By (4), we can write $\rho_{E(s)}(x) = \rho(x) + \delta_s^*(x)$, where $\lim_{s\to\infty} \delta_s^*(x) = 0$ for all $x$. Hence given $x$ and any $\epsilon > 0$, there always exists a sufficiently large finite $s$ for which $\delta_s^*(x) \leqslant \epsilon$. Since (17) holds for every finite $s$, it follows that for any $\epsilon > 0$, we can write

$$\rho_{\mathcal{E}}(x,n) \leqslant \rho(x) + \epsilon + \delta_s(n),$$

which proves (11) with $\delta_\epsilon(x,n) \triangleq \epsilon + \delta_s(n)$ and completes the proof of the theorem. Q.E.D.

It is easy to verify that inequality (7b) and the proof of Theorem 2 also imply the following corollary.

*Corollary 2:* Let $p(x_1^j)$ denote the number of phrases in the incremental parsing of $x_1^j$. Then for every infinite sequence $x$,

$$\rho(x) = \limsup_{n\to\infty} \limsup_{k\to\infty} \frac{1}{kn \log \alpha} \cdot \sum_{i=0}^{k} p(x_{in+1}^{(i+1)n}) \log p(x_{in+1}^{(i+1)n}).$$

We proceed now to examine the concept of compressibility from a different point of view. Given $w \in A^l$, an arbitrary word of length $l$ over $A$, and an input string $x_1^n \in A^n$, let

$$\delta(x_{i+1}^{i+l}, w) = \begin{cases} 1, & \text{if } x_{i+1}^{i+l} = w \\ 0, & \text{otherwise} \end{cases} \quad 0 \leqslant i \leqslant n-l. \quad (18)$$

Then the relative frequency of occurence of $w$ in $x_1^n$ is

$$P(x_1^n, w) = \frac{1}{n-l+1} \sum_{i=0}^{n-l} \delta(x_{i+1}^{i+l}, w), \quad (19)$$

which can also be interpreted as a "probability measure" of $w \in A^l$ relative to $x_1^n$. The corresponding normalized "entropy" is then given by

$$\hat{H}_l(x_1^n) = -\frac{1}{l \log \alpha} \sum_{w\in A^l} P(x_1^n, w) \log P(x_1^n, w) \quad (20)$$

where by definition $P \log P = 0$ whenever $P = 0$. Now we take

$$\hat{H}_l(x) = \limsup_{n\to\infty} \hat{H}_l(x_1^n) \quad (21)$$

and

$$\hat{H}(x) = \lim_{l\to\infty} \hat{H}_l(x). \quad (22)$$

The existence of this limit is established in Section III where we also prove the following results.

*Theorem 3:* For every infinite sequence $x$

$$\rho(x) = \hat{H}(x). \quad (23)$$

*Theorem 4:* If $x$ is drawn from an ergodic source with entropy $H$ then

$$\Pr[\rho(x) = H] = 1. \quad (24)$$

*Corollary 3:* If $x$ is drawn from a stationary source with entropy $H$ then

$$E\rho(x) = H \quad (25)$$

where $E$ denotes expectation.

In a recent paper [1], compressibility of an individual infinite sequence $x$ with respect to an IL *fixed-rate* encoder is measured in terms of a "finite-state complexity" $H(x)$, and results similar to those obtained here for $\rho(x)$ are established there for $H(x)$. Thus the roles played by $\rho(x)$ and $H(x)$ for individual sequences are analogous to that played by the Shannon entropy for probabilistic ensembles of sequences, and both $E\rho(x)$ and $EH(x)$ are appropriate candidates for a concept of "generalized entropy" in the nonstationary case.

## III. Proofs

*Proof of Theorem 1:* Given an encoder $E \in E(s)$ and an input string $x_1^n$, let

$$x_1^n = x_1^{n_1} x_{n_1+1}^{n_2} x_{n_2+1}^{n_3} \cdots x_{n_{c-1}+1}^{n}$$

be a parsing of $x_1^n$ into $c = c(x_1^n)$ distinct phrases, and let $c_j$ denote the number of phrases $x_{n_{i-1}+1}^{n_i}$, $1 \leqslant i \leqslant c$, (where $n_0 \triangleq 0$ and $n_c \triangleq n$) for which $y_{n_{i-1}+1}^{n_i}$, the corresponding output phrase, is $j$-bits long. Since the input phrases are all distinct, it follows from the IL property of $E$ that $c_j \leqslant s^2 2^j$ for all $j$. It is also clear that to obtain a lower bound on the length $L(y_1^n)$ in bits of $y_1^n$, we may overestimate $c_j$, $j = 0, 1, \cdots$, at the expense of $\sum_{i>j} c_i$, provided the sum of all $c_j$ remains equal to $c$. Thus if $q$ and $r$ are the nonnegative integers satisfying $c = qs^2 + r$, $0 \leqslant r < s^2$, and if

$$q = \sum_{j=0}^{k} 2^j + \Delta_k, \qquad 0 \leqslant \Delta_k < 2^{k+1},$$

then we may assume that $c_j = s^2 2^j$ for $0 \leqslant j \leqslant k$, $c_{k+1} = s^2\Delta_k + r$, and $c_j = 0$ for $j > k+1$. Therefore

$$c = s^2 \sum_{j=0}^{k} 2^j + s^2\Delta_k + r = s^2(2^{k+1} + t) \quad (26)$$

where

$$t = \Delta_k - 1 + \frac{r}{s^2}, \quad (27)$$

and

$$\begin{aligned} L(y_1^n) &\geqslant s^2 \sum_{j=0}^{k} j2^j + (k+1)(s^2\Delta_k + r) \\ &= s^2[(k-1)2^{k+1} + 2] + (k+1)(s^2\Delta_k + r) \\ &= s^2(k-1)(2^{k+1} + t) + s^2(k+3+2t) \\ &= (k-1)(c+s^2) + 2s^2(t+2). \end{aligned} \quad (28)$$

From (26) we have

$$k-1=\log\frac{c-s^2t}{s^2}-2=\log\frac{c+s^2}{4s^2}-\log\left[1+\frac{(t+1)s^2}{c-s^2t}\right],$$

which together with (28) yields

$$L(y_1^n)\geqslant(c+s^2)\left(\log\frac{c+s^2}{4s^2}+\tau\right) \tag{29}$$

where

$$\tau=\frac{2s^2(t+2)}{c+s^2}-\log\left[1+\frac{(t+1)s^2}{c-s^2t}\right].$$

Let $\phi=((t+1)s^2)/(c-s^2t)$. Then

$$\tau=\frac{2s^2}{c+s^2}+\frac{2\phi}{1+\phi}-\log(1+\phi),$$

and by (26) and (27) we have

$$\phi=\left(\Delta_k+\frac{r}{s^2}\right)2^{-(k+1)}.$$

From the definitions of $\Delta_k$ and $r$ it follows that $0\leqslant\phi<1$, and one can readily verify that over this interval, $2\phi\geqslant(1+\phi)\log(1+\phi)$. Hence $\tau\geqslant(2s^2)/(c+s^2)$, which together with (29) yields

$$L(y_1^n)\geqslant(c+s^2)\log\frac{c+s^2}{4s^2}+2s^2.$$

Dividing both sides of this inequality by $n\log\alpha$, we obtain the bound of Theorem 1. Q.E.D.

*Lemma 1:* The limit of (22) exists.

*Proof:* By (20) and (21), we can write

$$\begin{aligned}(l+m)\hat{H}_{l+m}(x)&=\frac{-1}{\log\alpha}\\&\quad\cdot\limsup_{n\to\infty}\sum_{w\in A^{l+m}}P(x_1^n,w)\log P(x_1^n,w)\\&=\frac{-1}{\log\alpha}\limsup_{n\to\infty}\sum_{u\in A^l}\sum_{v\in A^m}P(x_1^n,uv)\\&\quad\cdot\left[\log\frac{P(x_1^n,uv)}{P(x_1^n,u)}+\log P(x_1^n,u)\right]\\&=l\hat{H}_l(x)+\frac{1}{\log\alpha}\limsup_{n\to\infty}\sum_{v\in A^m}P(x_1^n,v)\\&\quad\cdot\sum_{u\in A^l}\frac{P(x_1^n,uv)}{P(x_1^n,v)}\log\frac{P(x_1^n,u)}{P(x_1^n,uv)}.\end{aligned}$$

By the convexity of the logarithm function, we can further write

$$\begin{aligned}(l+m)\hat{H}_{l+m}(x)&\leqslant l\hat{H}_l(x)+\frac{1}{\log\alpha}\\&\quad\cdot\limsup_{n\to\infty}\sum_{v\in A^m}P(x_1^n,v)\log\sum_{u\in A^l}\frac{P(x_1^n,u)}{P(x_1^n,v)}\end{aligned}$$

which reduces to

$$(l+m)\hat{H}_{l+m}(x)\leqslant l\hat{H}_l(x)+m\hat{H}_m(x).$$

Hence $l\hat{H}_l$ is subadditive in $l$, and since $0\leqslant\hat{H}_l(x)\leqslant1$, the limit of (22) exists. Q.E.D.

*Lemma 2 (Generalized Kraft Inequality):* For any given IL encoder $E$ with $s=|S|$ states,

$$K\triangleq\sum_{w\in A^l}2^{-L(w)}\leqslant s^2\left(1+\log\frac{s^2+\alpha^l}{s^2}\right) \tag{30}$$

where

$$L(w)=\min_{z\in S}\{L(f(z,w))\} \tag{31}$$

and $L(f(z,w))$ is the length in bits of $f(z,w)$, the output from $E$ when started in state $z$ and fed with $w$.

*Proof:* Let $k_j$ denote the number of $w\in A^l$ for which $L(w)=j$. Then $K=\Sigma_j k_j2^{-j}$ and $\alpha^l=\Sigma_j k_j$. By the IL property of $E$, it is clear that $k_j\leqslant s^2 2^j$. It is also clear that to obtain an upper bound on $K$, we may overestimate $k_j$, $j=0,1,\cdots$, at the expense of $\Sigma_{i>j}k_i$, provided the sum of all the $k_j$ remains equal to $\alpha^l$. Thus we can write

$$K\leqslant\sum_{j=0}^{m}(s^2 2^j)2^{-j}=s^2(m+1) \tag{32}$$

where $m$ is the integer satisfying

$$\sum_{j=0}^{m-1}s^2 2^j<\alpha^l\leqslant\sum_{j=0}^{m}s^2 2^j.$$

Furthermore

$$2^m=1+\sum_{j=0}^{m-1}2^j\leqslant\frac{\alpha^l}{s^2}+1$$

which together with (32) yields (30). Q.E.D.

*Proof of Theorem 3:* From the definition of $L(w)$ in (31), it is clear that for any IL encoder with $s$ states

$$\begin{aligned}\rho_{E(s)}(x_1^n)&=\frac{1}{n\log\alpha}\sum_{i=1}^{n}L(f(z_i,x_i))\\&=\frac{1}{ln\log\alpha}\sum_{i=1}^{n}lL(f(z_i,x_i))\\&\geqslant\frac{1}{ln\log\alpha}\sum_{i=0}^{n-l}L\left(f\left(z_{i+1},x_{i+1}^{i+l}\right)\right)\\&\geqslant\frac{1}{ln\log\alpha}\sum_{i=0}^{n-l}L\left(x_{i+1}^{i+l}\right).\end{aligned} \tag{33}$$

By (19), we can rewrite (33) as

$$\rho_{E(s)}(x_1^n)\geqslant\frac{n-l+1}{ln\log\alpha}\sum_{w\in A^l}P(x_1^n,w)L(w),$$

and we obtain

$$\rho_{E(s)}(x)\geqslant\frac{1}{l\log\alpha}\limsup_{n\to\infty}\sum_{w\in A^l}P(x_1^n,w)L(w).$$

By (20) and (21), we have

$$\begin{aligned}\hat{H}_l(x)-\rho_{E(s)}(x)&\leqslant\frac{1}{l\log\alpha}\limsup_{n\to\infty}\\&\quad\cdot\sum_{w\in A^l}P(x_1^n,w)\left(\log\frac{1}{P(x_1^n,w)}-L(w)\right)\\&=\frac{1}{l\log\alpha}\limsup_{n\to\infty}\sum_{w\in A^l}P(x_1^n,w)\log\frac{2^{-L(w)}}{P(x_1^n,w)}\end{aligned}$$

which by the convexity of the logarithm function and Lemma 2 reduces to

$$\hat{H}_l(x)-\rho_{E(s)}(x)\leqslant\frac{1}{l\log\alpha}\limsup_{n\to\infty}\log\sum_{w\in A^l}2^{-L(w)}$$

$$=\frac{s^2}{l\log\alpha}\left(1+\log\frac{\alpha^l}{s^2}\right).$$

Taking the limit as $l$ approaches infinity, we obtain

$$\hat{H}(x)-\rho_{E(s)}(x)\leqslant 0, \tag{34}$$

and since (34) holds for every finite $s$, we have

$$\hat{H}(x)\leqslant\rho(x) \tag{35}$$

for every infinite sequence $x$.

Using Huffman's coding scheme for input blocks of length $l$, it is easy to show [6] that

$$\rho(x)\leqslant\hat{H}_l(x)+\frac{\log\alpha}{l},$$

which when $l$ tends to infinity becomes

$$\rho(x)\leqslant\hat{H}(x) \tag{36}$$

for all $x$. Combining (35) with (36) completes the proof. Q.E.D.

*Proof of Theorem 4:* Since $x$ is drawn from an ergodic source, it follows that for every positive integer $l$ and for every $w\in A^l$

$$\Pr\left[P(x,w)=\Pr(w)\right]=1$$

where $P(x,w)=\lim_{n\to\infty}P(x_1^n,w)$ and $\Pr(w)$ is the probability measure of $w$. Therefore, if $H_l=1/l\Sigma_{w\in A^l}\Pr(w)\log\Pr(w)$, we obtain

$$\Pr\left[\hat{H}_l(x)=H_l\right]=1,$$

which when $l$ approaches infinity becomes

$$\Pr\left[\hat{H}(x)=H\right]=1.$$

From this and Theorem 3, we obtain Theorem 4. Q.E.D.

## Acknowledgment

The authors are thankful to M. Cohn, H. D. Shapiro, D. Slepian, and A. D. Wyner for many helpful discussions.

## References

[1] J. Ziv, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, IT-24, pp. 405–412, July 1978.
[2] S. Even, "Generalized automata and their information losslessness," *Switching Circuit Theory and Logical Design, AIEE Special Publ.*, S-141, pp. 144–147, 1961.
[3] S. Even, "On information lossless automata of finite order," *IEEE Trans. Electronic Computers*, vol. EC-1 pp. 561–569, Aug. 1965.
[4] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 75–81, Jan. 1976.
[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May 1977.
[6] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.