On the Complexity of Finite Sequences

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

Abstract—**A new approach to the problem of evaluating the complexity ("randomness") of finite sequences is presented. The proposed complexity measure is related to the number of steps in a self-delimiting production process by which a given sequence is presumed to be generated. It is further related to the number of distinct substrings and the rate of their occurrence along the sequence. The derived properties of the proposed measure are discussed and motivated in conjunction with other well-established complexity criteria.**

I. Introduction

ANY ATTEMPT to argue that a given finite sequence of fully specified numbers is a random sequence will be open to obvious criticism. On the other hand, in a wide variety of situations arising in digital data systems, it is adequate to have deterministically generated yet sufficiently "random looking" sequences that, while not being truly random, pass most of the agreed-upon randomness tests and allow for repeated experiments under similar "noise conditions."

Roughly speaking, the complexity of a finite fully specified sequence is a measure on the extent to which the given sequence resembles a random one. As in the case of randomness, any attempt to derive an allegedly absolute measure for the complexity of an individual sequence will raise understandable objections.

Early works in this area [1], [2] have linked the notion of complexity of a given sequence to that of an algorithm by which the sequence is supposed to be generated. Kolmogorov [1] has proposed to use the length of the shortest binary program, when fed into a given algorithm will cause it to produce a specified sequence, as a measure for the complexity of that sequence with respect to the given algorithm. Recent works in this field [3]–[6] have followed more or less along the same lines, while contributing to a further development and refinement of the pioneering ideas introduced by Kolmogorov and Chaitin.

In this paper, we propose and explore a new approach to the problem that links the complexity of a specific sequence to the gradual buildup of new patterns along the given sequence. Aside from introducing a variety of interesting combinatorial problems, the suggested approach leads, on one hand, to a new method of constructing "random looking" sequences and, on the other hand, to the identification of large families of sequences that yield themselves to efficient data compression.[1]

We do not profess to offer a new absolute measure for complexity which, as mentioned already, we believe to be nonexistent. Rather, we propose to evaluate the complexity of a finite sequence from the point of view of a simple self-delimiting learning machine which, as it scans a given n-digit sequence $S = s_1 s_2 \cdots s_n$ from left to right, adds a new word to its memory every time it discovers a substring of consecutive digits not previously encountered. The size of the compiled vocabulary, and the rate at which new words are encountered along S, serve as the basic ingredients in the proposed evaluation of the complexity of S.

The description of the exact mechanics of the outlined learning process requires some preparatory notation, definitions, and interpretation which take up all of Section II.

The proposed complexity measure is discussed in Section III, where it is also motivated and put to test against well-established test cases such as the de Bruijn sequences [8]. It is also shown in Section III that, under the proposed complexity measure, most sequences are complex, which is imperative for any complexity measure where "complex" is tantamount to "random." In particular, it is demonstrated that de Bruijn sequences are complex with respect to our definition of complexity. We further show that, under the proposed criterion, sequences from an ergodic source whose normalized entropy is less than unity do not qualify as complex ones, thereby demonstrating that the criterion is not too weak.

In Section IV we present a detailed study of the vocabulary of a sequence and examine the concept of complexity as related to the rate of vocabulary growth. We conclude by proposing yet another indicator of complexity which is related to the one discussed in Section III and which reflects more closely the rate of vocabulary growth.

II. Reproduction and Production of Sequences

Let A^* denote the set of all finite length sequences over a finite alphabet A. Let $l(S)$ denote the length of $S \in A^*$ and let

$$A^n = \{S \in A^* \mid l(S) = n\}, \qquad n \geq 0.$$

The null-sequence Λ, i.e., the "sequence" of length zero, is assumed to be an element of A^*.

A sequence $S \in A^n$ is fully specified by writing $S = s_1 s_2 \cdots s_n$; when S is formed from a single element $a \in A$, we write $S = a^n$. To indicate a substring of S that starts

Manuscript received January 3, 1975; revised April 17, 1975.

A. Lempel was with the Sperry Rand Research Center, Sudbury, Mass. He is now with the Department of Mathematical Sciences, Thomas J. Watson Research Center, Yorktown Heights, N.Y., on leave from the Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

J. Ziv was with Bell Laboratories, Murray Hill, N.J. He is now with the Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

[1] These applications will be treated separately in a forthcoming paper.

at position i and ends at position j, we write $S(i,j)$. That is, when $i \leq j$, $S(i,j) = s_i s_{i+1} \cdots s_j$ and $S(i,j) = \Lambda$, for $i > j$.

The concatenation of $Q \in A^m$ and $R \in A^n$ forms a new sequence $S = QR = q_1 q_2 \cdots q_m r_1 r_2 \cdots r_n \in A^{m+n}$, where $Q = S(1,m)$ and $R = S(m+1, m+n)$. We use the power notation $S^2 = SS$ to indicate concatenation of S with itself; more generally, $S^0 = \Lambda$ and $S^i = S^{i-1}S$, $i \geq 1$.

Q is called a *prefix* of $S \in A^*$, and S is called an *extension* of Q if there exists an integer i such that $Q = S(1,i)$; a prefix Q and its extension S are said to be *proper* if $l(Q) < l(S)$. When the length of a sequence S is not specified explicitly, it is convenient to identify prefixes of S by means of a special operator π according to $S\pi^i = S(1, l(S) - i)$, $i = 0,1,\cdots$. In particular, $S\pi^0 = S$ and $S\pi^i = \Lambda$, for $i \geq l(S)$.

The *vocabulary* of a sequence S, denoted by $v(S)$, is the subset of A^* formed by all the substrings, or *words*, $S(i,j)$ of S. For example,

$$v(0010) = \{\Lambda,0,1,00,01,10,001,010,0010\}.$$

A word $Q \in v(S)$ is called an *eigenword* of S if Q does not belong to the vocabulary of any proper prefix of S. The set of eigenwords of S, denoted by $e(S)$, is called the *eigenvocabulary* of S. For example,

$$e(0010) = \{10,010,0010\}.$$

Some simple relationships, implied directly by the above definitions, are recorded in the following lemmas.

Lemma 1: $v(S\pi) \subset v(S)$.

Lemma 2: $e(S) = v(S) - v(S\pi)$.

Further properties of the vocabulary and eigenvocabulary of a sequence S and their implications on the complexity of S are discussed in Section IV. At this point we proceed to describe the main mechanism by which we propose to evaluate the complexity of a given sequence.

When a sequence S is extended by concatenation with one of its words, say $W = S(i,j)$, the resulting sequence $R = SW$ can be viewed as being obtained from S through a simple copying procedure whereby $r_{m+l(S)} = w_m$ is copied from s_{i+m-1}, $m = 1,2,\cdots, j - i + 1$. With no additional effort, the same recursive copying procedure could be applied to generate an extension $R = SQ$ of S, which is much longer than warranted by any word in $v(S)$. The only provision is that Q be an element of $v(SQ\pi) = v(R\pi)$. Since $Q \in v(R\pi)$ implies the existence of a positive integer $p \leq l(S)$ such that $q_i = r_{p+i-1}$, $i = 1,2,\cdots,l(Q)$, we can start generating R from S by first copying the known symbol $r_p = s_p$ of S to obtain $q_1 = r_{1+l(S)}$; then we can obtain $q_2 = r_{2+l(S)}$ by copying r_{p+1} (which may still be a symbol of S or, if $p = l(S)$, the first and already known symbol of Q) and so on, until the last symbol of Q. Note that if $l(Q) > l(S) - p + 1$, then Q is simply a concatenation of several samples of $S(p,l(S))$, ending in a prefix thereof.

We say that an extension $R = SQ$ of S is *reproducible* from S, and write $S \to R$, if $Q \in v(R\pi)$. A position p of S such that $Q = R(p, l(Q) + p - 1)$ (i.e., a starting position of S that enables the copying procedure described above) is called a *pointer* for the reproduction $S \to R$. For example, $001 \to 00101010$ with $p = 2$.

A nonnull sequence S is said to be *producible* from its prefix $S(1,j)$, if $S(1,j) \to S\pi$ and $j < l(S)$. Producibility of S from $S(1,j)$ will be denoted by $S(1,j) \Rightarrow S$, and $S(1,j)$ will be referred to as a *base* of S. Note that every nonnull S has a base (e.g., $S\pi$) and that Λ is a base for every symbol $a \in A$ but for no other $S \in A^*$.

The distinction between producibility and reproducibility is best seen in the context of the recursive copying process that characterizes the latter. In addition to the simple copying, production allows for a single-symbol innovation at the end of the copying process. Thus if $R = SQ$ and $S \Rightarrow R$, then there exists a pointer $p \leq l(S)$ such that $q_i = r_{p+i-1}$, for $i = 1,2,\cdots, l(Q) - 1$, but the last symbol of Q need not be equal to $r_{p+l(Q)-1}$. (Any pointer for $S \to R\pi$ will also be referred to as a pointer for $S \Rightarrow R$.) For example, $01 \Rightarrow 0100$ with $p = 1$, but $01 \not\to 0100$, where $\not\to$ denotes the negation of $\to$.

The fact that every nonnull sequence S may be regarded as a production from some proper prefix of S, suggests the interpretation of the statement $Q \Rightarrow S$ as a mechanism of generating S from Q. In fact, any finite, nonnull sequence S can be interpreted as representing the final product of an iterative self-delimiting vocabulary-building process which, in its first step, performs $\Lambda = S(1,0) \Rightarrow S(1,1) = s_1$ and, having produced $S(1,h_i)$ from its base in step i, proceeds to perform $S(1,h_i) \Rightarrow S(1,h_{i+1})$, and so on until, after at most $l(S)$ steps, all of S has been produced. We call such a step-by-step mechanism of generating S a *production process* of S, and the result $S(1,h_i)$ of step i is called the ith *state* of the process.

III. Production Complexity of a Sequence

Consider an m-step production process of a sequence S and let $S(1,h_i)$, $i = 1,2,\cdots,m$, be the m states of the process. Note that $h_1 = 1$ and $h_m = l(S)$. The parsing of S into

$$H(S) = S(1,h_1)S(h_1 + 1, h_2) \cdots S(h_{m-1} + 1, h_m)$$

is called the (production) *history* of S and the m words $H_i(S) = S(h_{i-1} + 1, h_i)$, $i = 1,2,\cdots,m$, where $h_0 \equiv 0$, are called the *components* of $H(S)$.

A component $H_i(S)$ and the corresponding production step $S(1,h_{i-1}) \Rightarrow S(1,h_i)$ are called *exhaustive* if $S(1,h_{i-1}) \not\to S(1,h_i)$; a history (or, production process) is called exhaustive if each of its components, with a possible exception of the last one, is such.

It is easy to verify that every nonnull sequence S has a unique exhaustive history. For instance, the exhaustive history of the sequence $S = 0001101001000101$ is given by the following parsing of S

$$0 \cdot 001 \cdot 10 \cdot 100 \cdot 1000 \cdot 101$$

where successive components are separated by dots and where the absence of a dot at the end of the sequence indicates that the last component is not exhaustive. Henceforth, the exhaustive history of a sequence S will be denoted by $E(S)$.

Now we define the proposed measure $c(S)$ for the production complexity of a sequence S. Let $c_H(S)$ denote the number of components in a history $H(S)$ of S. Then

$$c(S) = \min \{c_H(S)\}$$

where the minimization is over all histories of S. In other words, $c(S)$ is the least possible number of steps in which S can be generated according to the rules of a production process.

The significance of $c(S)$ as a complexity indicator is further strengthened by the following observation. Any sequential encoding procedure employs a parsing rule by which a long string of data is broken down into words that are individually mapped into codewords. The production process defined here is an example of such a class of parsing rules. Since every codeword consists of at least one symbol (one bit, in the binary case), it follows that under the self-delimiting ground rules of production, the coded representation of a sequence S consists of at least $c(S)$ symbols.

Theorem 1: $c(S) = c_E(S)$, where $c_E(S)$ is the number of components in $E(S)$, the exhaustive history of S.

We postpone the proof of Theorem 1 to Section IV. This result is quite intuitive in view of the extremal innovative nature of an exhaustive component. For, in any given state of a production process, the next production step is longest when the resulting component is exhaustive. Nevertheless, Theorem 1 is not trivial since it has to be shown that waiving the exhaustive option at some step of the production process does not enable us to shorten the remainder of the process.

Let $\alpha = |A|$ denote the size of the alphabet A. In the sequel, unless explicitly stated otherwise, log (x) means the logarithm of x to the base α.

Theorem 2: For every $S \in A^n$,

$$c(S) < \frac{n}{(1 - \varepsilon_n) \log (n)}$$

where

$$\varepsilon_n = 2 \frac{1 + \log \log (\alpha n)}{\log (n)}.$$

Proof: By Theorem 1, $c(S)$ equals the number of components in the exhaustive history of S. From the definition of an exhaustive component it follows that all the components of $E(S)$, with a possible exception of the last one, are distinct. Let N denote the maximum possible number of distinct words into which a sequence of length n over A can be parsed. Clearly, $c(S) \leq N + 1$. Consider first the special case in which $n = n_k = \sum_{i=0}^{k} i\alpha^i$, for some non-negative integer k. It is easy to verify that for such length, the maximum number of distinct words is given by

$$N_k = \sum_{i=1}^{k} \alpha^i = \frac{\alpha}{\alpha - 1} (\alpha^k - 1).$$

(Think of the sequence formed by all distinct words of length one, followed by all distinct words of length two, and so forth up to all distinct words of length k.)

We have

$$n_k = \sum_{i=1}^{k} i\alpha^i = \frac{\alpha}{\alpha - 1} \left[\alpha^k \left(k - \frac{1}{\alpha - 1}\right) + \frac{1}{\alpha - 1}\right],$$

and hence

$$c(S) \leq N_k + 1 = \frac{\alpha}{\alpha - 1} (\alpha^k - 1) + 1 < \frac{\alpha}{\alpha - 1} \alpha^k$$

$$< \frac{n_k}{k - \frac{1}{\alpha - 1}} \leq \frac{n_k}{k - 1}.$$

Now any positive integer n can be expressed in the form $n = n_k + \Delta_k$, where $0 \leq \Delta_k < (k + 1)\alpha^{k+1}$. The increase in the number of distinct words caused by a Δ_k increment in length is clearly not greater than $\Delta_k/(k + 1)$, and hence, for any length n, we have

$$c(S) < \frac{n_k}{k - 1} + \frac{\Delta_k}{k + 1} < \frac{n_k + \Delta_k}{k - 1} = \frac{n}{k - 1}.$$

Since

$$k < \log (n) < \log \left(\sum_{i=1}^{k+1} i\alpha^i\right) < k + 1 + 2 \log (k + 1)$$

we have

$$k - 1 \geq \log (n) - 2 - 2 \log (k + 1)$$

$$> \log (n) - 2 - 2 \log (1 + \log (n))$$

or

$$k - 1 > \log (n) - 2[1 + \log \log (\alpha n)]$$

$$= \log (n) - \varepsilon_n \log (n).$$

Therefore,

$$c(S) < \frac{n}{(1 - \varepsilon_n) \log (n)}.$$

Q.E.D.

We shall now demonstrate that under our definition of complexity, "almost all sequences of sufficiently large length n are complex." That is, for almost all $S \in A^n$, $c(S)$ is very close to the upper bound given in Theorem 2. This property is typical (in fact, imperative) of any complexity measure where "complex" is tantamount to "random." More precisely, assigning the same probability measure α^{-n} to each element of A^n, we obtain the following.

Theorem 3: For every positive ε,

$$\lim_{n \to \infty} \Pr \left[c(S) < \frac{n(1 - \varepsilon)}{\log (n)} \,\middle|\, l(S) = n\right] = 0.$$

Proof: Let $\lceil x \rceil$ denote the least integer not smaller than x and let $\lfloor x \rfloor$ denote the largest integer not exceeding x. Consider a sequence $S \in A^n$ and a parsing of S into

$$L(S) = S(1,l)S(l + 1,2l)S(2l + 1,3l) \cdots S(kl + 1,n)$$

where $1 \leq l \leq n$ and $k = \lfloor n/l \rfloor$. Let

$$\delta_S(i) = \begin{cases} 0, & \text{if } S(1,(i - 1)l) \to S(1,il) \\ 1, & \text{otherwise,} \end{cases}$$

and let $S(1,l_i)$ be the sequence obtained in a single exhaustive production step from $S(1,(i-1)l)$. Then it is clear that

$$\delta_S(i) = 1, \qquad \text{if and only if } il \geq l_i$$

and hence

$$c(S) = c_E(S) \geq \sum_{i=1}^{k} \delta_S(i).$$

Denoting by $\bar{x}$ the expectation of x over the ensemble A^n, we obtain

$$\overline{c(S)} = \alpha^{-n} \sum_{S \in A^n} c(S) \geq \overline{\sum_{i=1}^{k} \delta_S(i)} = \sum_{i=1}^{k} \overline{\delta_S(i)}.$$

Now

$$\begin{aligned}\overline{\delta_S(i)} &= \Pr\left[S(1,(i-1)l) \nrightarrow S(1,il)\right] \\ &= 1 - \Pr\left[S(1,(i-1)l) \rightarrow S(1,il)\right] \\ &\geq 1 - \sum_{j=1}^{(i-1)l} \Pr\left[S((i-1)l+1,il) = S(j, l+j-1)\right].\end{aligned}$$

It is easy to verify that for all $j = 1,2,\cdots,(i-1)l$

$$\Pr\left[S((i-1)l+1,il) = S(j,l+j-1)\right] = \alpha^{-l}.$$

Hence

$$\overline{\delta_S(i)} \geq 1 - (i-1)l\alpha^{-l}$$

and

$$\begin{aligned}\overline{c(S)} &\geq \sum_{i=1}^{k} \left[1 - (i-1)l\alpha^{-l}\right] = k\left[1 - \frac{k-1}{2} l\alpha^{-l}\right] \\ &\geq \left(\frac{n}{l} - 1\right)\left[1 - \tfrac{1}{2}n\alpha^{-l}\right].\end{aligned}$$

Choosing for l the value $l = \lceil \log(n) + \log\log(n) \rceil$ we obtain

$$\overline{c(S)} \geq \frac{n}{\log(n)}\left(1 - 0\left(\frac{\log\log(n)}{\log(n)}\right)\right)$$

where $0(x)$ is a number of the same order of magnitude as x. The last inequality and Theorem 2 imply Theorem 3.

Q.E.D.

Theorem 3 shows that the proposed notion of complexity is not too restrictive. As has been rightly pointed out by the anonymous reviewers, it is equally important to demonstrate that our criterion is not too weak, i.e., that it does not allow "too many" sequences with structural regularities or vocabulary deficiencies to qualify as complex ones. In the context of finite sequencies, this is quite a delicate task since one can find some kind of regularity in any specific sequence of finite length.

There are several plausible ways to overcome this difficulty. The one we choose here is to regard a given sequence, as emanating from a discrete ergodic source and to link the complexity of the sequence with the entropy of the source.

According to the asymptotic equipartition property of ergodic sources [7, p. 60], sequences of sufficiently large length l from such an α-symbol source with normalized entropy h, $0 \leq h \leq 1$, can be partitioned into two subsets: a subset σ of "typical" sequences whose cardinality is roughly α^{hl}, and its complement $\sigma' = A^l - \sigma$, where the probability of σ tends to unity as $l \rightarrow \infty$. Furthermore, due to the ergodicity of the source, in almost all sufficiently long sequences the fraction of nontypical substrings is negligible.

In view of this observation, we may, in the proof of Theorem 2, replace α by α^h to obtain the asymptotic upper bound

$$c(S) \leq \frac{hn}{\log(n)}$$

for almost all n-sequences from an α-symbol ergodic source with normalized entropy h. Hence although the set of complex sequences has measure 1, it includes almost none from an ergodic source with $h < 1$.

Another property expected from any complexity measure is the following.

Theorem 4—(Subadditivity): $c(QS) \leq c(Q) + c(S)$.

Proof: Let $E(Q)$ and $E(S)$ be the exhaustive histories of Q and E, respectively. It is easy to see that the parsed concatenate $H(QS) = E(Q) \cdot E(S)$ is a history of QS. Hence, by Theorem 1,

$$c(QS) \leq c_H(QS) = c_E(Q) + c_E(S) = c(Q) + c(S).$$

Q.E.D.

A sequence S of length $n = \alpha^k + k - 1$ is called a *de Bruijn sequence* [8] if the α^k substrings $S(i, i+k-1)$, $i = 1,2,\cdots,\alpha^k$, form all the distinct words of length k over A and $S(1, k-1) = S(\alpha^k + 1, n)$. (More commonly, a de Bruijn sequence is defined to be a *closed* sequence of length α^k with distinct cyclicly successive substrings of length k.) With respect to most commonly accepted criteria, de Bruijn sequences are considered to be "good" finite approximations of complex sequences. As shown below, the same is true with respect to the complexity measure defined here.

Theorem 5: If S is a de Bruijn sequence of length

$$n = \alpha^k + k - 1$$

then

$$c(S) \geq \frac{n}{\log(n)}.$$

Proof: Consider a prefix $S(1,j)$ of S, where $1 \leq j < \alpha^k$. Since all substrings of length k are distinct, $S(1,j) \nrightarrow S(1, j+k)$, therefore, the length of every component starting in position $j < \alpha^k$ cannot exceed k. Hence

$$\begin{aligned}c(S) = c_E(S) &\geq \frac{\alpha^k + k - 1}{k} \\ &= \frac{n}{\log(n-k+1)} \geq \frac{n}{\log(n)}.\end{aligned}$$

Q.E.D.

From Theorems 2 and 5 it is easily seen that de Bruijn sequences are indeed complex.

We conclude this section by raising a question that might have occurred already to the reader: Why have we chosen

to define complexity in terms of a production process using the simple device of copying plus a unit innovation, rather than a more sophisticated function of the current state of the process? As it turns out, the merits of the model presented here extend beyond the mere, though important, aspect of simplicity. For the sake of critical comparison, we have considered the following generalized model.

Let f be an arbitrary mapping from A^m into A, where m is fixed, and let the relation $S \xrightarrow{f} SQ$ mean that there exists a position (pointer) p such that $1 \leq p \leq l(S) - m + 1$ and

$$q_i = f[R(p + i - 1, p + i + m - 2)], \qquad i = 1,2,\cdots,l(Q)$$

where $R = SQ$.

It is easy to see that $\xrightarrow{f}$ is a straightforward generalization of the reproduction relation $\rightarrow$, which corresponds to the identity mapping of A onto A. Similarly, we define an f-production $\overset{f}{\Longrightarrow}$ according to $S \overset{f}{\Longrightarrow} R$, if $S \xrightarrow{f} R\pi$ and $l(S) < l(R)$. Also, $S \overset{f}{\Longrightarrow} R$ is said to be exhaustive if $S \not\xrightarrow{f} R$.

Omitting the details, for the sake of brevity, it can be shown that results analogous to those of Theorems 1, 3, 4, and practically all those of Section IV are also valid with respect to any mapping f.

Furthermore, let m be a fixed integer and define $S \overset{m}{\Longrightarrow} R$ to mean that there exists a mapping f from A^i into A, with $1 \leq i \leq m$, such that $S \overset{f}{\Longrightarrow} R$. Then, even under these relaxed conditions, Theorems 2 and 3 still remain valid.

IV. Eigenvocabulary and Further Properties of Production Histories

In this section we examine the exhaustive history of a sequence as related to the rate of growth of its vocabulary and discuss another production history that more closely reflects the rate of vocabulary growth.

We begin by investigating the eigenvocabulary of a sequence as defined in Section II. Let $e(S)$ be the eigenvocabulary of a given sequence S. The cardinality of $e(S)$, denoted by $k(S)$, will be referred to as the *eigenvalue* of S.

Lemma 3: $k(\Lambda) = 0$ and for $S \neq \Lambda$, $1 \leq k(S) \leq l(S)$.

Proof: Since $e(\Lambda)$ is empty, $k(\Lambda) = 0$. For $S \neq \Lambda$, it follows from Lemma 2 (see Section II), that every eigenword of S is of the form $S(i,l(S))$ and that $S \in e(S)$. Hence, for $S \neq \Lambda$, the eigenvalue of S is a positive integer not exceeding the length of S. Q.E.D.

Note that for $S \neq \Lambda$, $k(S) = l(S)$, if and only if the last symbol of S differs from all of its predecessors. In particular, if $l(S) = 1$ then $k(S) = 1$.

Given the eigenvalue of S, we can explicitly identify the eigenwords of S as follows.

Theorem 6: $e(S) = \{S(i,l(S)) \mid 1 \leq i \leq k(S)\}$.

Proof: The theorem is trivially true for $S = \Lambda$, since in this case the range of i is empty. For $l(S) = n \geq 1$, let m be the largest integer such that $S(m,n) \notin v(S\pi)$. Clearly, $m \leq n$, or else, $S(m,n) = \Lambda \in v(S\pi)$. Since $S(i,n) \in v(S\pi)$ implies $S(i+1,n) \in v(S\pi)$ and since $S(1,n) = S \in e(S)$, it follows that $S(i,n) \in e(S)$, if and only if $1 \leq i \leq m$, and that $m = k(S)$. Q.E.D.

Lemma 4: $k(S\pi) \leq k(S)$.

Proof: If $l(S) = 0$, the lemma is trivial, and if $l(S) = 1$, it follows directly from Lemma 3. If $l(S) = n \geq 2$ and $k(S\pi) = m$, then $1 \leq m \leq n - 1$ and $S(m, n-1) \notin v(S\pi^2)$. This implies $S(m,n) \notin v(S\pi)$ which, in turn, implies $k(S) \geq m$. Q.E.D.

Thus when $S\pi$ is extended into S, the correspondingly extended eigenwords of $S\pi$ become eigenwords of S and, in addition, S may have some newly formed eigenwords due to the last symbol of S.

From the differential nature of $e(S)$, as expressed by Lemma 2, it is clear that $e(S)$ represents the per-symbol growth of the vocabulary of S at the last position of S. Lemma 4 shows that the rate of vocabulary growth at a given position is at least as high as at the preceding position, and the eigenvalue of S indicates the latest position of S at which there was an increase in the rate of vocabulary growth.

Lemma 5: $S(1,j) \rightarrow S$, if and only if $k(S) \leq j \leq l(S)$.

Proof: Let $l(S) = n$. By Theorem 6, $k(S)$ is the largest integer m such that $S(m,n) \notin v(S\pi)$. Hence $S(i,n) \in v(S\pi)$, if and only if $i \geq m + 1$ and, therefore, $S(1,j) \rightarrow S$, if and only if $m \leq j \leq n$. (Note that j is bounded by n because, by definition, $S(1,j)$ must be a prefix of S.) Q.E.D.

Thus the length of the shortest prefix of a sequence S from which S is reproducible equals the eigenvalue $k(S)$ of S.

Recalling the distinction between reproduction and production, one can readily verify the following analogue of Lemma 5.

Lemma 6: $S(1,j) \Rightarrow S$, if and only if $k(S\pi) \leq j \leq l(S) - 1$.

Consider now a history $H(S)$ of S. In Section III we have been mostly concerned with the exhaustive history of S. Another interesting history type, also characterized by an extremal property of its components, is the following.

A history component $H_i(S) = S(h_{i-1} + 1, h_i)$ is called *primitive* if h_i is the least integer such that the eigenvalue of $S(1,h_i)$ is greater than that of $S(1,h_{i-1})$; a history (or, production process) is called primitive if each of its components, with the possible exception of the last one, is primitive.

In comparison with an exhaustive production, we observe that both primitive and exhaustive production steps are innovative in the sense that both result in a larger eigenvocabulary. However, among all possible innovative productions, the shortest one is called primitive, while the longest production is the exhaustive one. As an example, consider the sequence S and the associated eigenvalues of its prefixes given by

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
s_i	0	0	0	1	1	0	1	0	0	1	0	0	0	1	0	1

and

$$k(S(1,i)) = 1\ 1\ 1\ 4\ 4\ 5\ 5\ 6\ 7\ \ 7\ \ 8\ \ 8\ 10\ 10\ 11\ 13.$$

The first component of any history of S must be $S(1,1) = s_1 = 0$, which is both primitive and exhaustive. The longest prefix of S, which is producible from $S(1,1)$, is $S(1,4) = 0^3 1$ (with pointer $p = 1$). Hence the second component can be either one of the following: $S(2,2) = 0$, $S(2,3) = 0^2$, and $S(2,4) = 0^2 1$. The first two of these possibilities are noninnovative, while the third one is again primitive and exhaustive. Suppose now that $S(1,9)$ is a state in some production process of S. Then since $S(1,13)$ is the longest one-step production from $S(1,9)$, the possible components are the following:

1) $S(10,10) = 1$, which is noninnovative;
2) $S(10,11) = 10$, which is primitive (but not exhaustive);
3) $S(10,12) = 10^2$, which is (innovative, but) neither primitive nor exhaustive;
4) $S(10,13) = 10^3$, which is exhaustive (but not primitive).

It is easy to verify that every nonnull sequence has a unique primitive history. For instance, the primitive history of the sequence S of the last example is given by the following parsing of S:

$$0 \cdot 001 \cdot 10 \cdot 10 \cdot 0 \cdot 10 \cdot 00 \cdot 10 \cdot 1 \cdot$$

where the dot at the end of the sequence indicates that the last component is also a primitive one.

To indicate the eigenvalue of every prefix of a given sequence S, it is convenient to associate with S a so-called *eigenfunction* g_S, defined for every $i = 0,1,\cdots,l(S)$, where $g_S(i)$ is the eigenvalue of $S(1,i)$, i.e., $g_S(i) = k(S(1,i))$. From Lemmas 3 and 4, it is easy to see that g_S is monotonically nondecreasing along S and that, for each i, $g_S(i) \leq i$.

Lemma 7: Let $H_i(S) = S(h_{i-1} + 1,h_i)$ be a component of a history $H(S)$. Then $H_i(S)$ is exhaustive, if and only if $h_i = \min \{h \mid g_S(h) > h_{i-1}\}$.

Proof: Since $H_i(S)$ is a history component we have $S(1,h_{i-1}) \Rightarrow S(1,h_i)$. Hence, by Lemma 6, $g_S(h_i - 1) \leq h_{i-1}$. For $H_i(S)$ to be exhaustive, it is necessary and sufficient that $S(1,h_{i-1}) \nrightarrow S(1,h_i)$ which, by Lemma 5, holds, if and only if $g_S(h_i) > h_{i-1}$. Hence it follows that $H_i(S)$ is exhaustive, if and only if h_i is equal to the least integer h such that $g_S(h) > h_{i-1}$. Q.E.D.

The following theorem is an immediate consequence of Lemma 7 and the maximal-length property of exhaustive components.

Theorem 7: A parsing of a nonnull $S \in A^*$ into $P(S) = S(1,p_1)S(p_1 + 1,p_2)\cdots S(p_{m-1} + 1,p_m)$ is a history of S, if and only if, for all $i = 1,2,\cdots,m$, $p_{i-1} < p_i \leq \min \{p \mid g_S(p) > p_{i-1}\}$, where $p_0 = 0$ and $p_m = l(S)$.

For later reference and comparison purposes, we use the formal form of a theorem to characterize the primitive and exhaustive histories of a sequence in a form similar to that of Theorem 7.

Theorem 8: Let $H(S) = S(1,h_1)S(h_1 + 1,h_2)\cdots S(h_{m-1} + 1,h_m)$ be a history of S. Then $H(S)$ is primitive if and only if, for $i = 1,2,\cdots,m-1$, $h_i = \min \{h \mid g_S(h) > g_S(h_{i-1})\}$, and for $h_m = l(S)$, $g_S(h_m - 1) = g_S(h_{m-1})$. $H(S)$ is exhaustive if and only if, for $i = 1,2,\cdots,m-1$, $h_i = \min \{h \mid g_S(h) > h_{i-1}\}$ and $g_S(h_m - 1) \leq h_{m-1}$.

Proof: For $i = 1,2,\cdots,m-1$, the exhaustive part of the theorem follows from Lemma 7, and the primitive part follows from the fact that $S(h_{i-1} + 1,h_i)$ is primitive, if and only if $g_S(h_i) > g_S(h_{i-1}) = g_S(h_i - 1)$. (The inequality guarantees innovation and the equality assures the minimality of the innovation.) For $i = m$, both parts of the theorem merely take into account a possible end-effect that may prevent the last component from being of the same type as the rest. Q.E.D.

Let $H(S)$ be a history of S and let H_S denote the set $\{h_1,h_2,\cdots,h_m\}$ of positions of S that characterize $H(S)$. The elements of H_S will be referred to as the *terminals* of the components of $H(S)$. A position i of S, $1 \leq i \leq l(S)$, is said to be *primitive in* S if $g_S(i) > g_S(i-1)$. Let p_S denote the set of all positions that are primitive in S and let $P_S = p_S \cup \{l(S)\}$.

It is clear from Theorem 8 that a history $H(S)$ is primitive, if and only if $H_S = P_S$. With regard to terminals of exhaustive components we have the following.

Lemma 8: If $S(h_{i-1} + 1,h_i)$ is an exhaustive component of a history $H(S)$, then $h_i \in p_S$.

Proof: If $S(h_{i-1} + 1,h_i)$ is exhaustive then, by Lemma 7, $h_i = \min \{h \mid g_S(h) > h_{i-1}\}$ or $g_S(h_i) > h_{i-1} \geq g_S(h_i - 1)$. Hence $g_S(h_i) > g_S(h_i - 1)$ and h_i is primitive in S. Q.E.D.

It follows from Lemma 8 that if $H(S)$ is an exhaustive history then $H_S \subseteq P_S$. A history $H(S)$ is said to be a *refinement* of another history $H'(S)$ if, for every component $S(h_{i-1} + 1,h_i)$ of $H(S)$, there exists a component $S(h'_{j-1} + 1,h_j')$ of $H'(S)$ such that $h_{i-1} \geq h'_{j-1}$ and $h_i \leq h_j'$. In other words, $H(S)$ is a refinement of $H'(S)$ if every component of $H'(S)$ is a concatenation of components of $H(S)$. In terms of component terminals, this means that $H(S)$ is a refinement of $H'(S)$, if and only if $H_S' \subseteq H_S$.

It is clear from our discussion that the following holds.

Theorem 9: For every nonnull $S \in A^*$, the primitive history of S is a refinement of the exhaustive history of S.

Now we present the proof of Theorem 1, referred to in Section III.

Proof of Theorem 1: Let $H_S = \{h_1,h_2,\cdots,h_m\}$ and $E_S = \{e_1,e_2,\cdots,e_k\}$, where $m = c_H(S)$, $k = c_E(S)$, $1 = h_1 < h_2 < \cdots < h_m = l(S)$, and $1 = e_1 < e_2 < \cdots < e_k = l(S)$. Let η be a mapping from E_S into H_S, defined by

$$\eta(e_i) = \max \{h \in H_S \mid h \leq e_i\}, \qquad i = 1,2,\cdots,k.$$

Obviously, $\eta(e_1) = h_1 = 1$ and $\eta(e_k) = h_m = l(S)$. If $k > 2$, consider any i such that $2 \leq i \leq k-1$ and let $\eta(e_i) = h_j$. It is clear that $j < m$ and that $e_i < h_{j+1}$. Since position

h_{j+1} is reached in a single production step from position h_j and since e_i is the farthest position reachable in one step from e_{i-1}, it follows that $e_{i-1} < h_j$. Hence for each i such that $2 \leq i \leq k-1$, we have $e_{i-1} < \eta(e_i) \leq e_i$, therefore, $\eta(e_i)$, $i = 1,2,\cdots,k$, is a one-to-one mapping from the set E_S onto a subset of H_S. Q.E.D.

We conclude by observing that one could use the primitive history of a sequence, rather than (or, in addition to) the exhaustive one, as a means for evaluating the complexity of a given sequence S. It is quite obvious that $c_E(S)$ is a rather conservative estimate of the complexity of S since it accounts for a unit "production cost" per each production step. The refinement relationship between the exhaustive and primitive histories of S, as expressed by Theorem 9, suggests that the number of primitive steps going into a single exhaustive step is probably a more realistic cost per exhaustive step.

ACKNOWLEDGMENT

The authors wish to thank Dr. S. Winograd and Dr. L. Shepp for many helpful discussions.

REFERENCES

[1] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Inform. Transmission*, vol. 1, pp. 1–7, 1965.
[2] P. Martin-Löf, "The definition of random sequences," *Inform. Contr.*, vol. 9, pp. 602–619, 1966.
[3] ——, "Complexity oscillations in infinite binary sequences," *Z. Wahrscheinlichkeit verw. Geb.*, vol. 19, pp. 225–230, 1971.
[4] C. P. Schnorr, "The process complexity and effective random tests," *J. Comput. Syst. Sci.*, vol. 7, pp. 376–388, 1973.
[5] G. Chaitin, "A theory of program size formally identical to information theory," IBM, Yorktown Heights, N.Y., Rep. RC 4805, Apr. 1974.
[6] ——, "Information-theoretic limitations of formal systems," *J. Ass. Comput. Mach.*, vol. 21, pp. 403–424, 1974.
[7] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[8] S. W. Golomb, *Shift Register Sequences*. San Francisco: Holden-Day, 1967.