JOURNAL OF COMPUTER AND SYSTEM SCIENCES 37, 324–336 (1988)

An Unpredictability Approach to Finite-State Randomness*

MARY G. O'CONNOR[1]

Applied Research, Bell Communications Research,
Morristown, New Jersey

Received July 2, 1986; revised January 21, 1988

This paper investigates the concept of randomness within a complexity theoretic framework. We consider an unpredictability approach for defining randomness in which the preditions are carried out by finite-state automata. Our model of a finite-state predicting machine (FPM) reads a binary sequence from left to right and depending on the machine's current state will generate, at each point, one of three possible values: 0, 1, or #. A response of 0 or 1 is to be taken as the FPMs prediction of the next input. A # means no prediction of the next input is made. We say that an infinite binary sequence appears random to an FPM if no more than half of the predictions made of the sequence's terms by the FPM are correct. The main result of this paper is to establish the equivalence of the sequences which appear random to all FPMs and the ∞-distributed sequences, where a binary sequence is called ∞-distributed if every string of length k occurs in the sequence with frequency 2^{-k}, for all positive integers k. We also explicitly construct machines that exhibit success in predicting the sequences which are not ∞-distributed. Finally, we show that for any given ∞-distributed sequence, all infinite subsequences which are constructible from FPMs are also ∞-distributed. © 1988 Academic Press, Inc.

1. INTRODUCTION

The concept of randomness plays an important role in diverse fields. Some of the many applications of random processes include sampling, probabilistic algorithms, simulations, signal processing, cryptographic systems, and numerical methods. Most often, however, methods for generating pseudo-random numbers are based on ad hoc procedures and involve little theory. The development of a definition of randomness from a theoretical foundation remains a significant problem for current mathematical research.

One approach to such a definition is to consider randomness with respect to various computational models (see [2, 9, 14, 15, 19, 23, 34, 35]). Suppose X is an arbitrary infinite binary sequence. One method of measuring the randomness of X is to assess the predictability of subsequent terms of X based on its earlier terms. With this method, the randomness of X depends on the type of prediction schemes

* The work reported in this paper was included in a doctoral dissertation written under the direction of Professor Michael Sipser at the Massachusetts Institute of Technology, 1985.

[1] Current address: Staff Director, Marketing & Technology, The New England Telephone Company, Boston, Massachusetts.

0022-0000/88 $3.00

applied. This paper proposes a definition of randomness using an unpredictability approach in which the predictions are carried out by finite-state automata.

We have chosen to focus on the finite automaton to provide a class of prediction schemes because of its simplicity as a minimal computational model. One contention of this paper is that by considering such a restrictive computational model, the relationship between the notion of randomness and complexity theory can be investigated more thoroughly. We establish the equivalence of sequences which are not predictable by finite automata and those that are ∞-distributed, thereby highlighting distributivity as a basic feature of randomness.

2. Predicting Machine Model

Our model of a finite-state predicting machine is one which reads a binary sequence from left to right and, depending on the machine's current state, will generate at each point one of three possible values: 0, 1, or #. A response of 0 to 1 is to be taken as the FPMs prediction of the next input. *A* # means no prediction of the next input is made. More formally, a *finite-state predicting machine* (FPM) is an ordered 6-tuple (I, R, S, s_0, f, g), where

I is the input alphabet $\{0, 1\}$;

R is the response alphabet $\{0, 1, \#\}$;

S is a finite non-empty set called the set of states;

s_0 is the machine's initial state;

f is the transition function which maps $S \times I$ into S;

g is the response function which maps S into R.

For convenience, we extend the domain of the transition function f such that the expression $f(s, W)$ signifies the state to which the machine goes if it is in state s and receives the string W as input.

Given machine M, its *complement* M^c, is found by replacing each state's prediction response having value 1 by 0, and vice versa. A prediction of # is left unchanged.

A predicting machine can be represented by a *state diagram*. Each state, s, along with its response value, $g(s)$, is represented by a labeled square. A double square

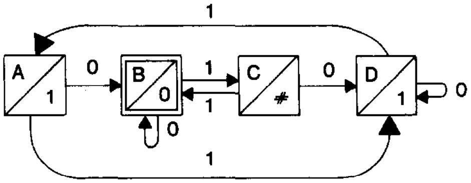
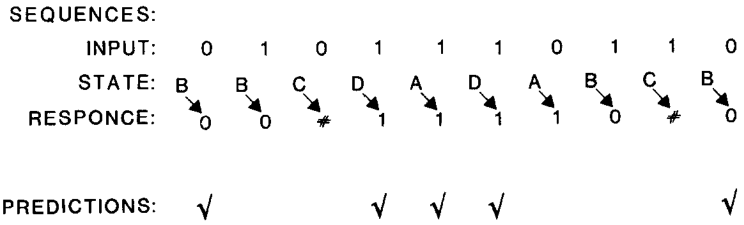


Fig. 1. State diagram of an FPM.



SEQUENCES:

INPUT:	0	1	0	1	1	1	0	1	1	0
STATE:	B	B	C	D	A	D	A	B	C	B
RESPONCE:	0	0	#	1	1	1	1	0	#	0
PREDICTIONS:	√			√	√	√				√

FIG. 2. Results of an FPM processing an input.

identifies the initial state. Each square has two outgoing arrows labeled 0 and 1 whose destinations are the states specified by the transition function f, see Fig. 1 for an example. Accordingly, for the string $W = 0101110110$ the scenario of Fig. 2 takes place.

One indication of how well this machine has predicted W is the ratio of the number of correct predictions to the number of predictions made. In this example a total of eight predictions were made, of these five were correct, thus yielding the ratio $\frac{5}{8}$.

3. Prediction Ratios

We now consider predictability in the more general case that the input to a predicting machine is an infinite binary string X. The number of predictions made could be either finite or infinite. In the latter case, to assess the predictability of X we use the upper limit of the prediction ratio as the number of predictions goes to infinity. Assuming that machine M makes an infinite number of predictions on X, the *prediction ratio* Φ is

$$\Phi(M, X) = \limsup_{p \to \infty} \frac{c(p)}{p},$$

where $c(p)$ is the number of correct predictions made among the first p predictions.

If $\Phi(M, X) > \frac{1}{2}$, then M will be designated as a *Predictor* of X, since M exhibits some success in making correct predictions on input X. Because we want the success to be a phenomenon in the long run, rather than on just some initial portion of X, a Predictor of X by definition must make infinitely many predictions when processing X. We say that X appears *random* to machine M iff M is not a Predictor of X.

4. Predictors for $\langle k \rangle$-Distributed Sequences

One type of infinite sequence relevant to our investigation of randomness is a k-distributed sequence. A binary sequence X is *k-distributed* if

$$\Pr(X_n, X_{n+1}, \ldots, X_{n+k-1} = x_1, x_2, \ldots, x_k) = 1/2^k$$

for all binary strings $x_1, x_2, ..., x_k$. In the above definition, we use the expression $\Pr(X_n, X_{n+1}, ..., X_{n+k-1} = x_1, x_2, ..., x_k)$ to mean $\lim_{n \to \infty} (f(n)/n)$, where $f(n)$ is the number of occurrences of the string $x_1, x_2, ..., x_k$ among the first n terms in X. A $\langle k \rangle$*-distributed* sequence is one that is k-distributed but not $(k+1)$-distributed.

THEOREM 1. *Every $\langle k \rangle$-distributed sequence has a $(k+1)$-state Predictor.*

Proof. Assume X is $\langle k \rangle$-distributed. Since X is not $(k+1)$-distributed, there exists at least one string of length $(k+1)$ for which its probability of occurring in X either does not exist, or exists but does not equal $2^{-(k+1)}$. To account for the possibility that a probability is not well-defined, we introduce below the notation $\overline{\Pr}(X_{n+k} = z \mid X_n X_{n+1} \cdots X_{n+k-1} = W)$. Let

$$\overline{\Pr}(X_{n+k} = z \mid X_n X_{n+1} \cdots X_{n+k-1} = W) = \limsup_{m \to \infty} \frac{v(m)}{m},$$

where $v(m)$ equals the number of times that the term $z \in \{0, 1\}$ has followed an occurrence of W, among the first m occurrences of W in X.

For some $W = w_1 w_2 \cdots w_k$, to be specified momentarily, let the prediction scheme of M in its entirety be that a prediction of a 1 is made after every occurrence of W. Note that

$$\Phi(M, X) = \overline{\Pr}(X_{n+k} = 1 \mid X_n \cdots X_{n+k-1} = W)$$

and

$$\Phi(M^c, X) = \overline{\Pr}(X_{n+k} = 0 \mid X_n \cdots X_{n+k-1} = W).$$

Since X is $\langle k \rangle$-distributed, there exists some W for which

$$\overline{\Pr}(X_{n+k} = 0 \mid X_n \cdots X_{n+k-1} = W) > \tfrac{1}{2} \quad \text{or} \quad \overline{\Pr}(X_{n+k} = 1 \mid X_n \cdots X_{n+k-1} = W) > \tfrac{1}{2}.$$

Let such a string W be the one given in the construction of M. Accordingly, either M or M^c is a Predictor of X. Assume the states of M are labeled $s_1, s_2 \cdots s_k, s_{k+1}$, with s_1 as the initial state. Let $g(s_{k+1}) = 1$, and $g(s_i) = \#$ for $1 \leqslant i \leqslant k$. Let $f(s_j, z) = S_{L(j,z)}$, where $L(j, z) = 1 +$ (the maximum length of a suffix of string $(w_1, w_2 \cdots w_{j-1}, z)$ which is a prefix of W). Machine M having $k+1$ states has now been constructed. ▮

Although this theorem provides a $(k+1)$-state Predictor for any $\langle k \rangle$-distributed sequence, it is possible that a Predictor with fewer states exists. For example, the infinite sequence having period 1111001011010000 is a $\langle 4 \rangle$-distributed sequence which has a 2-state Predictor.

5. Block Structure

We now evaluate $\Phi(M, X)$, where X is an ∞-distributed sequence and M is an arbitrary FPM which makes an infinite number of predictions on X. An *∞-distributed sequence* is one that is k-distributed for all positive integers k. Henceforth, we assume that M is an n-state machine, in which the states are labeled $s_1, s_2, ..., s_n$.

One aspect of our approach for evaluating $\Phi(M, X)$ is to section X into blocks of some judiciously chosen size. The sectioning of X into blocks of length k can be viewed as setting up barriers between certain terms of X, starting at the beginning of X, and spaced a distance k apart. An important property of sectioning X into blocks is given below and is due to John Maxfield [24].

Lemma 2. *For any positive integer k, if an ∞-distributed sequence is sectioned into blocks of length k, then blocks containing any particular string of length k occur with a frequency of 2^{-k}.*

With respect to machine M, the block structure of X for block size k is furthered developed by labeling each barrier with the state of M which occurs at that junction of processing X. We use the terminology "a block initialized with state s_j" to signify a block whose left barrier is labeled s_j.

6. Special Subclass of FPMs

Besides the use of a block structure, another basic aspect of our method for establishing that ∞-distributed sequences appear random to all FPMs is to single out for consideration just a certain subclass of FPMs. In this section, we define connected non-stop FPMs and show that if there exist Predictors for ∞-distributed sequences there would also exist some connected non-stop Predictors for ∞-distributed sequences.

For any given machine M and states s_i and s_j, we say that state s_j is *reachable* from state s_i if there exists a string P such that s_j is the state that M goes to, if it is in state s_i and receives the input P. Machine M is said to be *connected* if any state is reachable from any other.

Lemma 3. *If there exists a Predictor of an ∞-distributed sequence, then there exists a connected Predictor of a, not necessarily the same, ∞-distributed sequence.*

Proof. Let X be an ∞-distributed sequence, and suppose M is a Predictor of it. Consider the following equivalence relation among the states of M: two states are in the same equivalence class iff each is reachable from the other. Let a component of the state diagram of M consist of the states in an equivalence class along with the outgoing arrows that remain in that class. Because of the finite number of components, and the fact that once M leaves a component it may never return to it, in the course of processing any infinite binary string, M will eventually remain in one component. Let B refer to the component that M ultimately remains in when processing X. Since any outgoing arrow that leaves B will never be used when X is processed by M, we can w.l.o.g. assume that all outgoing arrows from states in B are assigned only to states in B. Let M_B denote the machine consisting of component B, with initial state being the first state in component B visited by M when processing X. Because of the equivalence relation by which the components were induced, M_B is a connected machine. Let X_B be the remaining terms of X to be processed when M enters component B. Since eventually all subsequent predictions made by M while processing X will emanate from component B, then $\Phi(M, X) = \Phi(M_B, X_B)$. Moreover, since $\Phi(M, X) > \frac{1}{2}$, then M_B is a connected Predictor of the ∞-distributed sequence X_B. ▮

For any given machine M, state s, and input sequence X, let

$$\overline{\Pr}(s) = \limsup_{m \to \infty} \frac{v(m)}{m},$$

where $v(m)$ is the number of times that M is in state s when the first m terms of X are processed. Because of its positive value, the following bound is sufficient for our purposes. It should be noted, however, that a much stronger bound is provable.

LEMMA 4. *Let M be a connected n-state predicting machine, and X be ∞-distributed. For any state s, $\overline{\Pr}(s) \geqslant 2^{-n(n-1)}$.*

Proof. The principal technique of the proof is to derive a finite string, Q, with the property that regardless of the state which M is in when Q begins, in processing Q, there will be at least one visit to state s. W.l.o.g. assume that s_1 is the given state. Define Q as $P_2 P_3 P_4 \cdots P_n$, where the P_i's are determined inductively as follows. Let P_2 be a string of length at most n such that M goes to state s_1 if it is in state s_2 and receives the input P_2. For $3 \leqslant i \leqslant n$, let P_i be a string of length at most n such that M goes to state s_1 if it is in state $f(s_i, P_2 P_3 \cdots P_{i-1})$ and receives the input P_i.

Letting q denote the length of Q, then $q \leqslant (n-1)(n)$, since Q consists of $(n-1)$ P_i-terms each of length at most n. Because X is ∞-distributed, we obtain

$$\Pr(Q \text{ occurring in } X) = 2^{-q} \geqslant 2^{-n(n-1)}.$$

Since each occurrence of Q signifies at least one visit to s_1, then $\overline{\Pr}(s_1) \geqslant 2^{-n(n-1)}$. ▮

We now consider machines that have the property that no state is assigned the prediction response #. A machine with this property will be referred to as a *nonstop predicting machine*, since such a machine makes a prediction of every input bit. One sequence of interest resulting from processing a sequence by a nonstop FPM is a record of the correctness or incorrectness of the predictions. For a given nonstop machine M and input W, the *assessment sequence* is a binary sequence (of the same length as W) whose ith term is 1 if w_i was correctly predicted, and whose ith term is 0 if w_i was incorrectly predicted.

An important relationship exists between input sequences and assessment sequences of the same length.

LEMMA 5. *For any given nonstop M and positive integer k, there is a* 1-1 *correspondence between the set of input sequences of length k and the set of assessment sequences obtained by processing these inputs on M.*

Proof. Since M is given, its initial state s_0, transition function f, and response function g are known. Furthermore, the response function of M^c, denoted here by g_c, can be determined. Given any string A of length k, a sequence W is derived below which when processed by M yields A as the assessment sequence. The terms of W, determined sequentially are:

For $i=1$: if $a_i=1$ then $w_i=g(s_0)$, else $w_i=g_c(s_0)$.
For $i>1$: if $a_i=1$ then $w_i=g(f(s_0, w_1 \cdots w_{i-1}))$, else $w_i=g_c(f(s_0, w_1 \cdots w_{i-1}))$.

By showing that every string of length k is an assessment string for some input to machine M, we have established the 1-1 correspondence between input sequences and assessment sequences of the same length. ▌

Let (M, s_i), for $1 \leqslant i \leqslant n$, denote the machine that results by resetting the initial state of M to s_i.

LEMMA 6. *For a given nonstop M, consider a string of length k for which at least one (M, s_i) makes precisely c correct predictions given the string as input. There are at most $n \times \binom{k}{c}$ such strings.*

Proof. The set of assessment sequences corresponding to inputs of length k for which any particular (M, s_i) makes precisely c correct predictions consists of the binary strings of length k which have precisely c terms equal to 1. By Lemma 5, each of these assessment sequences corresponds to a different input sequence. Therefore, there are $\binom{k}{c}$ inputs of length k on which (M, s_i) make precisely c correct predictions, and there are n (M, s_i) machines to be considered. ▌

7. NO PREDICTORS FOR ∞-Distributed Sequences

In this section we establish that there are no Predictors for ∞-distributed sequences.

LEMMA 7. *If there esists a Predictor of an ∞-distributed sequence, then there exists a nonstop connected Predictor of an ∞-distributed sequence.*

Proof. Assume M is a Predictor of some ∞-distributed sequence X. By Lemma 3, we can assume w.l.o.g. that M is connected. Let S_1 be the set of states of

M with prediction responses of 0 or 1, and S_2 the remaining states. If S_2 is empty there is no need to continue.

Let M' be derived from M by resetting each prediction response of the states in S_2 with 0. We proceed assuming that both $\Phi(M', X) = \frac{1}{2}$ and $\Phi(M'^{c}, X) = \frac{1}{2}$, since if the contrary were true then either M' or M'^{c} would be an example of a connected nonstop Predictor of X. We note that $\Phi(M', X)$ is not determined entirely by the predictions made by the states in S_2 since, by Lemma 4, $\overline{\Pr} > 0$, for each s in S_1. Further note that with machine M', the success of the states in S_1 in making correct predictions (as indicated by $\Phi(M, X) > \frac{1}{2}$) is offset by incorrect predictions made by states in S_2 (as indicated by $\Phi(M', X) = \frac{1}{2}$). Therefore, a connected nonstop Predictor of X is obtained from M' by changing the prediction responses of the states in S_2 to their opposite value, since the intervals of success for S_1 which were previously negated by incorrect predictions by S_2 will now be enhanced by correct predictions by S_2.

In summary, by applying the result that each state of a connected machine contributes to the overall prediction ratio and by reassigning some of the prediction responses of M, we have shown that if there exists a Predictor of an ∞-distributed sequence, then there exists a nonstop connected Predictor of an ∞-distributed sequence. ▌

THEOREM 8. *If X is ∞-distributed, then it has no Predictor.*

Proof. Our proof is by contradiction. Suppose that M is a Predictor of some ∞-distributed sequence X. By Lemma 7, we may assume that M is a connected nonstop machine. Since M is a Predictor of X, then there is some positive valued ε, say $\varepsilon_{M,X}$, so that $\Phi(M, X) = \frac{1}{2} + \varepsilon_{M,X}$.

We construct a sequence Y, based on the block structure of X induced by M and appropriate block size k, to obtain an upper bound of $\Phi(M, X)$. Although k is yet unspecified, assume that $k \geqslant \log_2 n$. In deriving Y, we define below functions β and α. For each string W of length k, let

$$\beta_W = \max_{S_i \in S} \{\text{proportion of correct predictions made by } (M, s_i) \text{ on input } W\},$$

where S is the set of states of M. Rank the strings of length k from 1 to 2^k, such that for any two strings (of length k) U and V, the rank of U is less than the rank of V iff $\beta_U \geqslant \beta_V$. Assign an α value to each W as follows:

- If the rank of W is in the range $[1, n]$, then set $\alpha_W = 1$.
- Otherwise, determine the value of j for which the rank of W falls in the range

$$\left[1 + n \times \sum_{i=0}^{j-1} \binom{k}{i},\ n \times \sum_{i=0}^{j} \binom{k}{i}\right] \quad \text{and} \quad \text{set } \alpha_w = \frac{k-j}{k}.$$

Let Y be the sequence of α values formed by mapping each block of X to its α value.

By applying Lemma 6, we have that $\beta_W \leqslant \alpha_W$, for any W. Hence, regardless of which state of M initializes a block consisting of string W, the proportion of correct prediction made within the block does not exceed α_W. Accordingly, the average value of the terms of Y is an upper bound of $\Phi(M, X)$. Let D_1 be the set of all strings of length k with ranks in the range $[1, n \times \lfloor 2^k/n \rfloor]$. Let D_2 be the set of remaining strings of length k.

Let h_k be the average proportion of heads among the $1/n$th most successful outcomes of tossing a fair coin k times. Since it is possible to associate uniquely with each such outcome n strings of length k from D_1 each of whose α value equals the probability of the outcome, the average α value of the strings in D_1 equals h_k. Since the average α value of the strings in D_2 does not exceed that of the strings in D_1, the strings in D_2 have an average α value no more than h_k. Accordingly, since all k-length strings are equally likely among the blocks of X, then h_k is an upper bound for the average value of the terms of Y. Hence, $\Phi(M, X) \leqslant h_k$.

By the well-known properties of the binomial distribution, the upper $1/n$th area of the distribution curve of the proportion of heads among k tosses of a fair coin becomes arbitrarily close to $\frac{1}{2}$, as k increases. Hence there exists a k such that $h_k < \frac{1}{2} + \varepsilon_{M,X}$. Accordingly, for such a k, $\Phi(M,X) \leqslant h_k < \frac{1}{2} + \varepsilon_{M,X}$. But this contradicts our original assumption that $\Phi(M, X) = \frac{1}{2} + \varepsilon_{M,X}$. ▌

8. Characterization of Finite-State Randomness

Using some of the results derived above, we now prove the following:

Theorem 9. *The sequences that appear random to the class of finite-state predicting machines are precisely the ∞-distributed ones.*

Proof. Since a sequence which is not ∞-distributed is $\langle k \rangle$-distributed for some k, then by Theorem 1, any sequence which is not ∞-distributed has a Predictor. By Theorem 8, any sequence which is ∞-distributed has no Predictor. ▌

To apply this theorem in an interesting setting, imagine that the prediction schemes provided by FPMs are those adhered to by gamblers. The game is to occasionally make predictions of the terms of an infinite sequence supplied by the house. The rules are simple, the gambler must pay $ 1.00 for every prediction he makes and will receive $ 2.00 for a correct prediction. If the sequence supplied by the house is ∞-distributed, then in the long run, assuming the gambler makes an infinite number of predictions, no matter what finite-state prediction scheme he chooses, he will not beat the system. Perhaps the only good news for the gambler who is doing poorly is that by continuing to gamble, in the long run he will break even. In contrast, if the sequence made available by the house is not ∞-distributed, then there is a finite-state predicting scheme the gambler can follow to beat the system.

9. Subsequences Constructible from FPMs

For a given machine M and sequence X, define subsequence $C_{M,X}$ as the terms of X taken in sequential order for which machine M has tried to predict. A subsequence of X is said to be *constructible by an* FPM if the subsequence equals $C_{M,X}$ for some M. We now prove that if X appears random to all FPMs, then all infinite subsequences of X constructible from FPMs also appear random. This is an appealing property since one criterion used in critiquing the merits of a definition of randomness is if the infinite subsequences of a random sequence are also random.

Theorem 10. *If X is an ∞-distributed sequence and M makes an infinite number of predictions on X, then $C_{M,X}$ is also an ∞-distributed sequence.*

Proof. If $C_{M,X}$ were not ∞-distributed, then it would be $\langle k\rangle$-distributed for some k. Suppose this is the case. We show how to construct machine T based on $k+1$ copies of M for which either T or T^{c} is a Predictor of X. This will contradict Theorem 8 and prove our result.

Since $C_{M,X}=C_{M',X}$, where M' is obtained from M by replacing each state's prediction response having value 0 to 1, we can assume w.l.o.g. that all states of M which make predictions are assigned the response 1. For notational convenience, let C represent $C_{M,X}$, and let s_1 be the initial state. Since C is $\langle k\rangle$-distributed, there exists a string W for which

$$\overline{\Pr}(C_{n+k}=0 \mid C_n \cdots C_{n+k-1}=W)>\tfrac{1}{2} \quad \text{or} \quad \overline{\Pr}(C_{n+k}=1 \mid C_n \cdots C_{n+k-1}=W)>\tfrac{1}{2}.$$

Based on such a W and machine M, the prediction scheme of T is as follows: whenever the preceding k terms that M has tried to predict equal W, T will predict a 1 of the bit that M next tries to predict.

We now construct machine T. Machine T has $n\times(k+1)$ states which are indexed by a double subscript. Let $z\in\{0,1\}$, and $L(j,z)$ be as defined in proof of Theorem 1. Furthermore let f_M and f_T be the transition functions of M and T, respectively, and g_M and g_T be their respective response functions.

Construction of T:

- Set of states: $\{s_{i,j} \mid 1\leqslant i\leqslant n \text{ and } 1\leqslant j\leqslant k+1\}$ with initial state: $s_{1,1}$.
- Response function

$$g_T(s_{i,j})=\begin{cases} \# & \text{if } \ j\neq k+1 \\ g_M(s_i) & \text{if } \ j=k+1. \end{cases}$$

- Transition function: $f_T(s_{i,j},z)=s_{a,b}$, where $a=f_M(s_i,z)$ and

$$b=\begin{cases} j & \text{if } \ g_M(s_i)=\ \# \\ L(j,z) & \text{otherwise.} \quad \blacksquare \end{cases}$$

This theorem extends a previously known collection of subsequences of an

∞-distributed sequence which are also ∞-distributed. Let X be an ∞-distributed sequence sectioned into blocks of size k, for any k. Ivan Niven and H. S. Zuckerman [26] proved that the subsequence of X formed by the first m terms in each block is ∞-distributed. Moreover, John Maxwell [24] showed that given integers $0 \leqslant c_1 < c_2 < \cdots < c_j < m$, the subsequence of X obtained upon deletion of all blocks except those in positions congruent to c_1 or c_2 or $\cdots$ or c_j $(\operatorname{mod} m)$ is ∞-distributed.

It should be evident that there are finite automata to construct subsequences like those specified in [24 or 26], but note that the indices of the terms selected to form such a subsequence are the same regardless of the input sequence. Hence, our result is more general since the infinite subsequences constructible from finite automata also include those in which the indices of the terms selected to form a subsequence depend on the input sequence.

10. Conclusion

Using an unpredictability approach for defining randomness, we established the equivalence between the ∞-distributed sequences and the sequences which appear random to finite automata. We also provided an upper bound on the minimal number of state for which a Predictor exists for any $\langle k \rangle$-distributed sequence. Another important result showed that for any given ∞-distributed sequence, all infinite subsequences which are constructible from FPMs are also ∞-distributed.

To underscore the premise that the computational resources used to measure the randomness of binary sequences influence which sequences appear random, we conclude by pointing out that when using other computational models to construct prediction schemes, the *random* sequences are not necessarily the ∞-distributed sequences. One reason is that it is possible to replace an infinite number of terms of an ∞-distributed sequence with the value 1 in a particularly prescribed manner and still maintain an ∞-distributed sequence. For example, let X be ∞-distributed, and for all integers n, replace each (n^2)th term of X with a 1 to obtain X''. As shown in Knuth [20], X'' is ∞-distributed. Consider now a computational model which predicts a 1 of every (n^2)th term of an input sequence. This prediction scheme applied to X'' yields all correct predictions, and hence X'' would not be considered random within the given computational environment using an unpredictability approach. Of course, a finite automaton cannot carry out this prediction scheme, but models such as Turing machines can.

Acknowledgments

I thank Michael Sipser for his guidance and many valuable discussions. Partial support for this work was provided by NSF Grant MCS 83-04769.

References

1. C. Anderson and B. Shubert, Testing a simple symmetric hypothesis by a finite-memory deterministic algorithm, *IEEE Trans. Inform. Theory* **IT-19**, No. 5 (1973), 644–647.
2. M. Blum and S. Micali, How to generate cryptographically strong sequence of pseudo-random bits, *SIAM J. Comput.* **13**, No. 4 (1984), 850–864.
3. T. Booth, "Sequential Machines and Automata Theory." Wiley, New York, 1967.
4. K. Brownlee, "Statistical Theory and Methodology in Science and Engineering," Wiley, New York, 1960.
5. J. W. S. Cassels, On a paper of Niven and Zuckerman, *Pacific. Math.* **3** (1953), 555–557.
6. G. Chaitin, On the length of programs for computing finite binary sequence, *J. Assoc. Comput. Mach.* **13** (1966), 547–569.
7. D. Champernowne, The construction of decimels normal in the scale of ten, *London Math. Soc. J.* **8** (1933), 250–264.
8. B. Chandrasekaran and C. Lam, A fintie-memory deterministic algorithm for the symmetric hypothesis testing problem, *IEEE Trans. Inform. Theory* **IT–21**, No. 1 (1975), 40–44.
9. A. Church, On the concept of a random sequence, *Bull. Amer. Math. Soc.* February (1940), 130–135.
10. A. Copeland and P. Erdos, Note on normal numbers, *Bull. Amer. Math. Soc.* **52** (1956), 857–860.
11. T. Cover and M. Hellman, The two-armed bandit problem with time-invariant finite memory, *IEEE Trans. Inform. Theory* **IT-16**, No. 2 (1970), 185–195.
12. T. Cover and M. Hellman, Learning with finite memory, *Ann. Math. Statist.* **41** (1970), 341–352.
13. R. Daley, An example of information and computations resource trade-off, *J. Assoc. Comput. Mach.* **20** (1973). 687–695.
14. W. Gerwirz, "Investigations in the theory of Descriptive Complexity," Report No. NSO–5, Courant Institute of Mathematical scienes.
15. O. Goldreich, S. Goldwasser, and S. Micali, How to construct random functions in "Proceedings, of the 25th Found. of Comput. Sci., 1984," pp. 464–479.
16. H. Gordon, Complete degrees of finite-state transformability, *Inform. and Control* **32**, No. 2 (1976), 169–187.
17. P. Hirschler, Finite memory algorithms for testing Bernoulli random variables, *Inform. and Control* **24**, No. 1 (1974), 11–19.
18. J. Hopcroft and J. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison–Weslay, Reading, MA, 1979.
19. A. Kolmogorov, Three approaches to the quantitative definition of information, *Problemy Peradachi Informatsii* **1** (1965), 3–11.
20. D. Knuth, "The Art of Computer Programming," Vol. 2, Addison–Wesley, 2nd Ed., Reading, MA, 1981.
21. F. Leighton and R. Rivest, "Estimating a Probability Using Finite Memory," MIT/LCS/Technical Memo 248, November 1983.
22. D. Loveland, A variant of the Kolmogorov concept of complexity, *Inform. and Control* **15** (1969), 510–526.
23. P. Martin-Lof, The definition of random sequences, *Inform. and Control* **9** (1966), 602–619.
24. J. Maxfield, Normal k-tuples, *Pacific J. Math.* **3** (1953), 189–196.
25. N. Metropolis and G.-C. Rota, Combinatorial structure of the faces of the n-cube, *SIAM J. Appl. Math.* **35**, No. 4 (1978), 689–694.
26. I. Niven and H. S. Zuckerman, On the definition of normal numbers, *Pacific J. Math.* **1** (1951), 103–110.
27. G. Rayna, Degrees of finite-state transformability, *Inform, and Control* **24**, No. 2 (1974), 144–154.
28. H. Rogers, "Theory of Recursive Functions and Effective Computability," McGraw–Hill, New York, 1967.
29. G. Rose and J. Ullian, Approximations of functions on the integers, *Pacific J. Math.* **13** (1964), 693–701.

30. A. Salomaa, "Theory of Automata," Pergamon, Oxford, 1969.
31. F. Samaniego, On testing simple hypothesis in finite time with Hellman–Cover automata, *IEEE Trans. Inform. Theory* **IT-21**, No. 2 (1975), 157–162.
32. C. Schnorr, A unified approach to the definition of random sequences, *Math. Systems Theory* **5** (1970), 246–258.
33. B. Shubert, Finite-memory classification of Bernoulli sequences using reference samples, *IEEE Trans. Inform. Theory* **IT-20** (1974), 384–387.
34. M. Sipser, A complexity theoretic approach to randomness, *in* "Proceedings, 15th ACM Symposium on the Theory of Computing, 1983," pp. 330–335.
35. M. Sipser, "Three approaches to the definition of finite-state randomness," unpublished.
36. R. Wilber, Randomness and the density of hard problems, *in* "Proceedings 24th IEEE Symposium on Foundations of Computer Science, 1983," pp. 335–342.