Proof: Let us first give a definition. A word $u \in A^*$ is *unbordered* if $u \neq vwv$ for any words $v, w \in A^*$ with v nonempty. Let us now consider a regular MSD code X. If X is complete, the theorem is trivially true. If X is not complete, there exists, by the definition of completeness, a word $v \notin F(X^*)$. Let $a \in A$ be a letter that is different from the first letter of v (we assume that card $(A) \geq 2$). Consider the word $u = va^{|v|}$. Clearly, u is unbordered and $u \notin F(X^*)$ because $v \notin F(X^*)$. Let $Y = \{uw_1uw_2 \cdots uw_nu | n \geq 1$, for all i from 1 to n, $w_i \notin X^* \cup A^*uA^*\}$, and let

$$Z = X \cup Y \cup \{u\}.$$

We first prove that Z is an MSD code. It suffices to prove that, given a word $z \in Z^*$, any possible parsing of z into elements of Z yields the same multiset of words. Since u is unbordered, z has a unique representation of the form $z = z_1uz_2u \cdots uz_n$ (that is, we can uniquely distinguish all occurrences of u in z). This representation provides the basis for the division of z into Z-blocks which is obtained as follows.

1) A factor $uz_juz_{j+1} \cdots uz_{j+k}u$ of z such that, $2 \leq j$, $j+k \leq n-1$, $z_j, \cdots, z_{j+k} \notin X^*$ and $z_{j-1}, z_{j+k+1} \in X^*$, constitutes a Z-block.
2) All occurrences of u not involved in blocks of type 1 are also Z-blocks.
3) All z_i which are not involved in blocks of type 1 must be in X^* and constitute Z-blocks.

The definition of Y and the fact that $u \notin F(X^*)$ and u is unbordered guarantee that z admits a unique parsing into Z-blocks. Z-blocks corresponding to cases 1 and 2 are elements of Z. Z-blocks corresponding to case 3 are elements of X^*. Since X is an MSD code, any parsing of these Z-blocks into elements of X yields the same multiset of words. Thus Z is an MSD code.

We now prove that Z is complete. Given an arbitrary word $w \in A^*$, one easily derives, by using the same argument as above, that the word uwu has a factorization into elements of Z, i.e., w is factor of an element of Z^*. Finally, the proof that Z is a regular set can be easily derived from its definition. This concludes the proof of the theorem.

We can now state the main results of this correspondence. The following theorem gives a positive answer to [3, conjecture 2].

Theorem 3: No finite MSD code contains a maximal UD code as a proper subcode.

Proof: We will prove the result by contradiction. Let X be a finite MSD code over the alphabet A; assume that a maximal UD code Y exists such that Y is a proper subset of X. Then there exists a word $s \in X - Y$. We will prove that there exist a word $m \in Y^*$ and an integer $p \geq 1$ such that $(msm)^p \in Y^*$; this contradicts the hypothesis that X is an MSD code. In fact in the equality

$$(msm)^p = y_1y_2 \cdots y_k$$

with $m \in Y^*$, $s \in X - Y$, $y_1, y_2, \cdots, y_k \in Y$, the left side contains a codeword, say s, which does not appear on the right side.

In the proof we use some techniques from automata theory. As a consequence of Theorem 1, if Y is a finite maximal UD code, then it is also complete. Since Y is finite, Y^* is a regular set; i.e., it is recognizable by a finite (deterministic) automaton $\mathcal{A} = (A, Q, \partial, i, F)$. For any set of states $S \subseteq Q$ and for any word $u \in A^*$, denote by Su the set $\{\partial(q, u) | q \in S\}$ of states reached by paths having label u and starting at any state of S. Let $n = Inf$ card (Qu) with u ranging over A^*, and choose u such that $n =$ card (Qu). Since Y is complete, we have $vuw = m \in Y^*$ for some $v, w \in A^*$. Since $Qvu \subseteq Qu$, it follows that card $(Qm) \leq$ card (Qu). Thus, by minimality, card $(Qm) = n$. Let $Q' = Qm$. Since $Q'm = Qmm \subseteq Qm = Q'$, it follows from the minimality of n that $Q'm = Q'$ and thus m defines a permutation of Q'. Thus replacing m by a suitable power of m, we may assume that $q'm = q'$ for all $q' \in Q'$. Let us now consider a word $s \in X - Y$, and let $t = msm$. Again we have $Qt \subseteq Qm$, and thus $Qt = Q' = Q't$. Thus again, for some power t^p, $p \geq 1$, we have $q't^p = q'$ for all $q' \in Q'$. To prove that

$$t^p = (msm)^p \in Y^*,$$

it suffices to show that $qt^p = qm$ for all $q \in Q$. Since $qmm = qm$, it follows that $qt = qmsm = qmmsm = qmt$ and therefore that $qt^p = qmt^p$. Since $Qm = Q'$, we have $qmt^p = qm$. Thus $qt^p = qm$, as required. This completes the proof.

The next theorem gives a negative answer to [3, conjecture 3].

Theorem 4: There exists a finite MSD code X such that $MS(X) > 1$.

Proof: A MSD code is *proper* if it is not a UD code. In [3] it is shown that there exist finite proper MSD codes. By Theorem 2, there exist regular MSD codes that are proper and complete. Let Z be such a code. Since Z is complete, by Theorem 1, $MS(Z) \geq 1$. If $MS(Z) = 1$, the fact that Z is complete implies, by using again Theorem 1, that Z is a UD code, which is a contradiction. Thus $MS(Z) > 1$. There exists, then, a finite subset $X \subseteq Z$ such that $MS(X) > 1$. This concludes the proof.

Remark: The completion procedure in the proof of Theorem 2 also gives an explicit construction for MSD codes whose McMillan sum exceeds unity. Consider indeed a finite and proper MSD code X. Take the set $Y \cup \{u\}$, as in Theorem 2, in some given order (for instance, the lexicographic order):

$$Y \cup \{u\} = \{u_1, u_2, u_3, \cdots\}.$$

The proof of Theorem 4 guarantees that a positive integer k exists such that the finite MSD code

$$Z = X \cup \{u_1, u_2, \cdots, u_k\}$$

satisfies the inequality $MS(Z) > 1$.

REFERENCES

[1] J. Berstel and D. Perrin, *The Theory of Codes*. New York: Academic, 1985.
[2] A. Ehrenfeucht and G. Rozenberg, "Each regular code is included in a maximal regular code," *RAIRO Inform. Théor. Appl.*, vol. 20, pp. 89–96, 1986.
[3] A. Lempel, "On multiset decipherable codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 714–716, 1986.

On Write-Unidirectional Memory Codes

GÁBOR SIMONYI

***Abstract*—Write-unidirectional memories generalize write-once memories storing binary sequences of some fixed length in a reusable manner. At every new usage the content of the memory can be rewritten by either changing some of the zeroes to ones, or by changing some of the ones to**

Manuscript received June 12, 1987; revised July 18, 1988. This work was supported in part by the Hungarian National Foundation for Scientific Research Grant 1806 (OTKA). This correspondence was presented at the Third Soviet–Swedish Workshop on Information Theory, Sochi, USSR, May 24–30, 1987, and presented in part at the IEEE Information Theory Workshop, Bellagio, Italy, June 21–25, 1987.

The author is with the Mathematical Institute of the Hungarian Academy of Sciences, Budapest P.O.B. 127, H-1364, Hungary.

IEEE Log Number 8927897.

zeroes, but not both. Improving on the results of Willems and Vinck, we construct codes of rate 0.5325. We discuss the four cases that arise according to whether or not the encoder and/or the decoder is informed of the previous state of the memory. Borden's converse bound is rederived using Fibonacci sequences.

I. Introduction

Write-unidirectional memories (WUM's) were recently introduced by Borden [1] and by Willems and Vinck [2]. They are closely related to write-once memories (WOM's) defined by Rivest and Shamir in 1982 [3] (cf. [4] and [5]). WUM's are binary storage media for multiple uses in which n binary symbols can be stored at every single usage. The memory can be reused (updated) with the restriction that the encoder is allowed either to write 1's in some chosen positions and skip all others or to write 0's in some positions and skip all others. This means that 0's and 1's cannot be written at the same updating.

The motivation for studying such a device comes from the updating process for rewritable optical disks. Here the choice of 0 or 1 as the binary digit stored depends on the orientation of the magnetic field generated by an electromagnet. The change of orientation of the magnetic field is a slow procedure. This is why the WUM is a preferable model for such a device.

Unlike in a WOM the role of 0's and 1's is symmetric, and a WUM can be used an arbitrary number of times. Accordingly, the effectiveness of a WUM will be measured by the quantity of stored information per updating. (In what follows an updating process will also be called a generation.) The main problem is how to construct good codes in this sense.

In Section II we give the basic definitions and discuss previous work. In Section III we generalize the construction of Willems and Vinck to a family of WUM codes having some better members than the Willems–Vinck code.

In Section IV, we consider the four cases that arise according to whether or not the encoder and/or the decoder is informed of the previous state of the memory. This parallels the four cases investigated by Wolf *et al.* for WOM's [4]. One of these cases is closely related to an unsolved conjecture of Erdös and Katona [8]. In Section V we present a new proof of Theorem 1 of Borden [1] based on Fibonacci sequences.

II. Preliminaries

Borden [1] (and also Willems and Vinck [2]) introduced WUM's under the assumption that the encoder knows the previous state of the memory before writing new information, while the decoder has no information about the previous state. We shall initially restrict attention to this case.

Definition 1: Let $x=\{x_1,\cdots,x_n\}$ and $y=\{y_1,\cdots,y_n\}$ be binary vectors of length n. We say that x and y are comparable ($x\sim y$) if

$$\text{for all } i \quad x_i=1\Rightarrow y_i=1 \qquad \text{or for all } i \quad y_i=1\Rightarrow x_i=1.$$

The technical condition that the encoder should not change the orientation of the magnet during a rewriting means formally that a state x can be rewritten into a y only if they are comparable.

For an error-free decoding we have to be able to partition the possible states of the WUM (these are binary n-vectors) into disjoint sets corresponding to the different messages, i.e., each element of a set S_i will be decoded as message m_i. Given a state of the WUM, the condition for being able to update it is the existence for every i of at least one element in S_i comparable to the given state. More formally, we have the following.

Definition 2: A family $C=\{S_1,\cdots,S_M\}$ of subsets of $\{0,1\}^n$ is a WUM code if

1) $S_i\cap S_j=\varnothing$, if $i\neq j$

2) for all $i\neq j$ for all $x\in S_i$ $\exists y\in S_j$, such that $x\sim y$.

Every set S_i is associated with a different message, and is called its codeset. The rate of a WUM-code is

$$R\triangleq\frac{\log M}{n}$$

where, herein, all the logarithms are to the base 2. We are interested in constructing WUM codes with high rates for large n. The larger the rate the more efficient the WUM code is. For each n the largest possible R is denoted by $R(n)$.

In what follows we define a special class of WUM codes. This class is important because its elements of any length can be expanded with the same rate for arbitrarily large n.

Definition 3: The weight of a binary n-vector x, denoted by $w(x)$, is the number of 1's among its coordinates. If $u\sim v$ and $w(u)\geqq w(v)$, we write $u\geqq v$.

Definition 4: A family $C=\{S_1,\cdots,S_M\}$ of subsets of $\{0,1\}^n$ is an alternating WUM code if

1) $S_i\cap S_j=\varnothing$, if $i\neq j$;
2) for every i there exist two sets T_{i0} and T_{i1} with the following properties:

$S_i=T_{i0}\cup T_{i1}$, and for all i, all $x\in T_{i0}$, and all j, there exists $y\in T_{j1}$ such that $y\geqq x$. For all i, all $x\in T_{i1}$ and all j, there exists $y\in T_{j0}$ such that $y\leqq x$.

The WUM codes introduced in [2] are alternating WUM codes. It is easy to see that if we construct an alternating WUM code of length n_0, then we can construct an alternating WUM code with the same rate and length ln_0 (l is some positive integer) by simple concatenation. This is not true for general WUM codes.

Let $\bar{x}$ denote the componentwise complement of the binary n-vector x. For $l\in\{0,1\}$, we set $\bar{l}=1-l$.

Definition 5: An alternating WUM code is symmetric if

$$\text{for all } i\colon T_{i1}=\overline{T_{i0}}, \qquad \text{where } \overline{T}=\{\bar{x}|x\in T\}.$$

Theorem 1 (Borden): It holds that $R(n)<\gamma\triangleq\log((1+\sqrt{5})/2)\approx 0.6942$ if $n\geqq 5$.

Theorem 2 (Borden): We have $\lim_{n\to\infty}R(n)=\gamma$.

These are proved in [1]: the latter is proved by a random coding argument; i.e., it is nonconstructive. The best explicit construction in [1] for arbitrarily large n yields $R=1/2$ and is the following.

Assume that n is even, and let

$$S_i=\{(\mathbf{1},x_i);(\bar{x}_i,\mathbf{0})\}, \qquad \text{where } \mathbf{0},\mathbf{1},x_i\in\{0,1\}^{n/2}$$

$\mathbf{1}$ is the all 1 vector, $\mathbf{0}$ is the all 0 vector, and x_i is an arbitrary $n/2$ vector. Less formally, the memory is divided into two equal parts and an arbitrary $n/2$ vector is written on the left (right) half while the all 0 (all 1) $n/2$ vector is written on the right (left) half. It is easy to see that this is really a WUM code with $R=1/2$.

A more refined construction was given by Willems and Vinck in [2] (They also state without proof a result implying Theorem 1.) The code of Willems and Vinck achieves a rate of $(\log 6)/5\approx 0.517$. In Section III we generalize this construction describing a family of WUM codes. The best member of this family yields the rate $(\log 58)/11\approx 0.5325$.

III. A Family of WUM-codes Yielding $R=0.5325$

In this section we construct a family of WUM codes all of which are alternating and symmetric. Because of the alternating property, our construction can be used for arbitrarily large mem-

ories. As usual we will identify a binary n vector with that subset of $\{1,2,\cdots,n\}$, of which it is the characteristic vector.

Baranyai's Theorem [6]: If k divides m, then the k sets of an m set can be partitioned into $\binom{m-1}{k-1}$ classes so that for any class its elements give a disjoint covering of the m set.

Now write $m=k(k+1)$. The foregoing theorem claims that there exist $\binom{m-1}{k-1}$ different families $U_i=\{A_{ij}\}_{j=1}^{k+1}$ of k-element subsets of $\{1,\cdots,m\}$ with the following properties:

1) for every i: $j\neq l \Rightarrow A_{ij}\cap A_{il}=\emptyset$;
2) $U_i\cap U_j=\emptyset$, if $i\neq j$.

Let x_{ij} denote the characteristic vector of A_{ij}. Then x_{ij} is an m vector with weight k. Let $T_{i0}^*=\{x_{ij},\ j=1,\cdots,k+1\}$, and so $T_{i1}^*=\{\bar{x}_{ij},\ j=1,\cdots,k+1\}$. We claim that in this way we obtain a WUM code.

Lemma 1: The family of sets

$$C^*=\left\{S_i^*=T_{i0}^*\cup T_{i1}^*,\ i=1,2,\cdots,\binom{m-1}{k-1}\right\}$$

is an alternating WUM-code.

Proof: By the definition, $S_i^*\cap S_j^*=\emptyset$ if $i\neq j$ is obvious. Consider an arbitrary $u\in T_{il}^*(l\in\{0,1\})$ and $j\neq i$. We have to prove that there exists $v\in T_{jl}^*$, $v\sim u$. Since the construction is symmetric in 0 and 1, we can assume $u\in T_{i0}^*$. Then $w(u)=k$.

Consider the elements $v\in T_{j1}^*$. Note that

$$u(u_1,\cdots,u_m)\not\leqq v(v_1,\cdots,v_m)\Rightarrow \exists r\quad u_r=1,\ v_r=0.$$

However, it follows immediately from the construction that if v, $v'\in T_{j1}^*$, then the sets

$$z(v)=\{r|v_r=0\}\quad\text{and}\quad z(v')=\{r|v_r'=0\}$$

are disjoint.

Suppose indirectly that $u\not\leqq v$ for each $v\in T_{j1}^*$. Then it follows that for each $v\in T_{j1}^*\ \exists r\quad u_r=1,\ v_r=0$ and because of the previous consequence of the construction this r must be different for each $v\in T_{j1}^*$. However, $|T_{j1}^*|=k+1$ implies $w(u)\geqq k+1$, a contradiction. This completes the proof.

Remark: The idea of Willems and Vinck was to construct a symmetric WUM code where $\max_{x\in T_{i0}}w(x)<|T_{j1}|$ and some structural property will guarantee that the code is a WUM code. The novelty here is the recognition of the relevance of Baranyai's theorem which gives us the flexibility to construct such codes for many different values of the parameters. In the next lemma we slightly modify the previous code, thereby improving the rate.

Lemma 2: By dropping the last bit of each codeword in the WUM code of Lemma 1, we get a new code C with length $n=m-1=k(k+1)-1$. C is again an alternating WUM code.

Proof: Let x' denote the codeword obtained from x by dropping its last bit. It is trivial to show that

$$u\leqq v\Rightarrow u'\leqq v'.$$

Denote by S_i the set obtained from S_i^* of C^*. It follows from Lemma 1 and the foregoing fact that for any $i\neq j$ and $u'\in T_{i0}$, there exists $v'\in T_{j1}$ such that $v'\geqq u'$.

It only remains to prove that $S_i\cap S_j=\emptyset$ for each $i\neq j$. Note that all the codewords in C^* had a weight of k or $m-k$. Then it is clear that the last bit is determined by the previous $m-1$ ones. Hence $S_i^*\cap S_j^*=\emptyset$ implies $S_i\cap S_j=\emptyset$. This completes the proof.

By Lemmas 1 and 2 and the possibility of concatenation we have a family of WUM codes with the following parameters (k and l are positive integers):

$$n_k=[k(k+1)-1]\cdot l$$

$$M_k=\binom{n_k}{k-1}^l$$

$$R_k=\frac{\log\binom{n_k}{k-1}}{k(k+1)-1}.$$

The foregoing construction already achieves $R\approx 0.522$ for $k=4$ and $R\approx 0.525$ for $k=3$. We can, however, improve our codes by adding further code sets to them.

Until now the code sets of C include all the codewords of weight k and $k-1$, their complements, and nothing else. If we can construct some further T_{i0} with codewords of weight less than $k-1$ (T_{i1} will be $\bar{T}_{i0}$ as before) while keeping the important property of our construction that $\forall i\ |T_{i0}|=k+1$ and the codewords in a T_{i0} are characteristic vectors of disjoint subsets of $\{1,2,\cdots,n\}$, then the obtained code will remain an alternating (and symmetric) WUM code with an obviously better rate. The proof of the last statement is similar to that of Lemma 1. Furthermore, note that the all-0 and all-1 vectors are comparable with all others; thus these two vectors can form a separate code set of cardinality two. (In fact, we could use the all-0 and the all-1 vectors as two separate code sets of cardinality one. In this way, however, we could get a nonalternating code, and the possibility of concatenation would be lost.)

The maximum number of new code sets constructed by the above method is at most

$$\left\lfloor\frac{\sum_{i=1}^{k-2}\binom{n}{i}}{k+1}\right\rfloor+1.$$

This is always achievable as one can easily verify using the following.

General Form of Baranyai's Theorem [6]: Let $a_1,\cdots,a_t$ be natural numbers such that $\sum_{j=1}^t a_j=\binom{n}{h}$. Then the h-sets of an n-set can be partitioned into disjoint families $S_1,\cdots,S_t$ such that $|S_j|=a_j$ and each $i\in\{1,2,\cdots,n\}$ is included in exactly $\lfloor a_j\cdot h/n\rfloor$ or $\lfloor a_j\cdot h/n\rfloor$ elements of S_j.

For our purposes we use this theorem with h running over the set $\{1,2,\cdots,k-2\}$. For every fixed h we choose $a_2(h)=\cdots=a_{t-1}(h)=k+1$. For $h>1$ $a_1(h)=k+1-a_t(h-1)$ $a_1(1)=k+1$ while $a_t(h)=\binom{n}{h}-\sum_{j=1}^{t-1}a_j(h)$. In this way it is easy to get such families of subsets of $\{1,2,\cdots,n\}$ the characteristic vectors of which define codesets of the desired type.

This completes the description of our construction. The new code has the following parameters (k and l are positive integers):

$$n_k=(k(k+1)-1)l$$

$$M_k=\left[\binom{n_k}{k-1}+\left\lfloor\frac{\sum_{i=1}^{k-2}\binom{n_k}{i}}{k+1}\right\rfloor+1\right]^l=\left[\left\lfloor\frac{\sum_{i=1}^{k}\binom{n_k}{i}}{k+1}\right\rfloor+1\right]^l$$

$$R_k=\frac{\log\left(\left\lfloor\frac{\sum_{i=1}^{k}\binom{n_k}{i}}{k+1}\right\rfloor+1\right)}{k(k+1)-1}.$$

We are interested in the behavior of R_k for different k's. It is easy to check that for $k \to \infty$, $R_k \to 0$ and thus we can hope to achieve good rates for small k's. The parameters of the codes we obtain for different k's and $l=1$ given in Table I.

TABLE I
PARAMETERS OF THE CODES OBTAINED FOR DIFFERENT k'S AND $l=1$

k	n_k	M_k	R_k
2	5	6	$(\log 6)/5 \approx 0.517$ (Willems–Vinck code)
3	11	58	$(\log 58)/11 \approx 0.5315$ (the best code in this family)
4	19	1008	0.525
5	29	24433	0.503
≥ 6			$<1/2$

Our construction gives the code of Willems and Vinck as a special case for $k=2$. The best code we obtain has $k=3$. The T_{i0} parts of some of its code sets are as follows (for $i=1,2,\cdots,55$ T_{i0} can be obtained by permuting the columns in this matrix form of T_{10}; Baranyai's theorem ensures that this is possible while $T_{i0} \cap T_{j0} = \varnothing$ remains true for $i \neq j$):

$$T_{10} = \begin{Bmatrix} 11100000000 \\ 00011100000 \\ 00000011100 \\ 00000000011 \end{Bmatrix}$$

$$T_{56\ 0} = \begin{Bmatrix} 10000000000 \\ 01000000000 \\ 00100000000 \\ 00010000000 \end{Bmatrix} \quad T_{57\ 0} = \begin{Bmatrix} 00001000000 \\ 00000100000 \\ 00000010000 \\ 00000001000 \end{Bmatrix}$$

$$T_{58\ 0} = \{00000000000\}, \qquad T_{i1} = \overline{T_{i0}} \text{ for } i = 1,2,\cdots,58.$$

IV. General Cases

In this section we will consider the following four cases:

Case 1: both encoder and decoder are informed about the previous state of the memory;
Case 2: encoder informed, decoder uninformed about the previous state of the memory;
Case 3: encoder uninformed, decoder informed about the previous state of the memory;
Case 4: both encoder and decoder uninformed about the previous state of the memory.

Until now we have considered WUM's in Case 2. For the more general discussion we introduce a new formulation. Instead of set systems we will now define WUM codes as mappings.

Let $\mathcal{M} = \{m_1, \cdots, m_M\}$ be the set of all possible messages. Given a message, for every cell of the memory, the encoder has three choices: write a 1, write a 0, or leave the content of the cell unchanged. Hence given a message, the action of the encoder can be described by an n vector over the alphabet $\{0, 1, \square\}$ where $\square$ refers to the case when the content of the cell is left unchanged.

Let $F_1 = \{1,\square\}^n$ $F_2 = \{0,\square\}^n$ $F = F_1 \cup F_2$. Then the elements of F represent the possible actions of the encoder, while the elements of $\{0,1\}^n$ are the possible states of the memory.

To formalize, we introduce a function φ: $\{0,1,\square\} \times \{0,1\} \to \{0,1\}$ as follows:

$$\varphi(\epsilon, x) = x, \qquad \text{if } \epsilon = \square$$
$$\varphi(\epsilon, x) = \epsilon, \qquad \text{else.}$$

With a slight abuse of notation we denote by φ: $\{0,1,\square\}^n \times \{0,1\}^n \to \{0,1\}^n$ the obvious extension of the function. Notice, however, that we shall be dealing with the restriction of the former to $F \times \{0,1\}^n$.

During the history of successive updatings of the WUM the actions of the encoder and the decoder are described by the functions f^i, g^i where i refers to the generation number. The problems connected with the generation number are discussed in [7]. Here we assume that both the encoder and the decoder know enough about the generation number to be able to function as described. Considering the different cases.

Case 1 (Encoder and Decoder Informed): For every t the encoding function f^t is a mapping

$$f^t: \mathcal{M} \times \{0,1\}^n \to F$$

while the decoding function is

$$g^t: \{0,1\}^n \times \{0,1\}^n \to \mathcal{M}.$$

An error-free decoding is guaranteed if and only if

$$g^t\big(\varphi\big(f^t(m,x),x\big),x\big) = m$$

holds for every $m \in \mathcal{M}$ and every x which is a possible state of the memory at generation $t-1$.

Case 2 (Encoder Informed, Decoder Uninformed): For every t the encoding function f^t is a mapping

$$f^t: \mathcal{M} \times \{0,1\}^n \to F$$

while the decoding function is

$$g^t: \{0,1\}^n \to \mathcal{M}.$$

An error-free decoding is guaranteed if and only if

$$g^t\big(\varphi\big(f^t(m,x),x\big)\big) = m$$

holds for every $m \in \mathcal{M}$ and every x which is a possible state of the memory at generation $t-1$.

Note that this formulation is slightly more general than the one used in Section I. If $f^t(m,x)$ and $g^t(x)$ depend on t only through x, then setting

$$S_i = \{x | g(x) = m_i\}$$

we obtain the previous formulation. For alternating codes we can set

$$T_{i0} = \{x | \exists y \text{ possible codeword with } y \geq x,\ x = \varphi(f(m_i, y), y)\}$$

while

$$T_{i1} = \{x | \exists y \text{ possible codeword with } y \leq x,\ x = \varphi(f(m_i, y), y)\}.$$

Cases 3 and 4 can be described in a similar way with an encoding function f^t: $\mathcal{M} \to F$ and decoding functions g^t: $\{0,1\}^n \times \{0,1\}^n \to \mathcal{M}$ and g^t: $\{0,1\}^n \to \mathcal{M}$, respectively. We now redefine the alternating property.

Definition 6: A WUM code (in all four cases) is alternating if the range of the encoding function f^t is a subset of F_1 or of F_2 depending on the parity of t.

Remark: It still remains true that by concatenating alternating WUM codes we obtain alternating WUM codes of the same rate and double, triple, etc., length. One can easily check that definition 6 is really a generalization of Definition 4.

Definition 7: An alternating WUM code is said to be symmetric (in all four cases) if there are only two different pairs (f^t, g^t) and

$$f^2 \equiv \overline{f^1}$$

where the complementation is understood componentwise as before and the complement of $\square$ is $\square$.

We now consider the four cases separately and state some basic facts for each one of them.

Case 1 (Encoder and Decoder Informed): It is easy to check that Borden's proof of Theorem 1 remains true in this case. Thus $R(n) < \gamma$ for $n \geq 5$ (from Borden's Theorem 1), and $\lim_{n\to\infty} R(n) = \gamma$ (from Borden's Theorem 2).

Using the decoder's knowledge we can use a nonrandomized version of Borden's random code (which helped prove his Theorem 2, see [1]). This achieves the asymptotic optimum. For a detailed description see [7].

Case 2 (Encoder Informed, Decoder Uninformed): This case was considered in Section III. Borden's Theorems 1 and 2 give the best possible asymptotic upper and lower bounds for $R(n)$. The best known construction is the one described in Section III, yielding a rate of $(\log 58)/11 \approx 0.5325$.

For Cases 3 and 4 we have the following.

Theorem 3: In Cases 3 and 4 for every WUM code of rate R_0 there exists an alternating WUM code having exactly the same rate. Our proof of this theorem actually yields a construction (cf. [7]).

Consider now Cases 3 and 4 separately.

Case 3 (Encoder Uninformed, Decoder Informed): The only upper bound we know for $R(n)$ in this case is that of Theorem 1. The best lower bound we have derives from the following construction. We construct an alternating and symmetric WUM code with length $n=3$, and $M=3$, yielding the rate of $(\log 3)/3 \approx 0.528$. Because of the alternating property this rate is achieved asymptotically. Set

$$f^1(m_1)=\square 11 \quad f^1(m_2)=1\square 1 \quad f^1(m_3)=11\square$$

$$f^2=\overline{f^1}.$$

Lemma 3: There exist decoding functions g^1 and g^2 such that, together with the above encoding functions, they yield an error-free WUM-code.

Proof: Set g^1 and g^2 as follows:

$$g^1(011,x)=m_1 \qquad g^1(111,100)=m_1$$

$$g^2(101,x)=m_2 \qquad g^1(111,010)=m_2$$

$$g^1(110,x)=m_3 \qquad g^1(111,001)=m_3$$

$$g^2(x,y)=g^1(\bar{x},\bar{y}).$$

It is easy to see that f^1, f^2, g^1, g^2 define an error-free WUM code.

Lemma 4: The following two questions are equivalent:

1) for Case 3, what is the maximum number $M(n)$ for which there exists an alternating and symmetric error-free WUM code of length n?
2) what is the maximum cardinality of a family $\mathcal{F}$ of subsets of an n set with the following property:

   $\mathcal{F}=\{A_i | A_i \subseteq \{1,\cdots,n\}$: there exists no triple A_i, A_j, A_k such that $A_i \neq A_j$ and $A_i \triangle A_j \subseteq A_k\}$, where $A_i \triangle A_j$ denotes the symmetric difference of A_i and A_j.

Proof: Consider a family $\mathcal{F}_0=\{A_i | A_i \subseteq \{1,\cdots,n\}\}$. Define the function f^1 as follows:

$$f^1(m_i)=\boldsymbol{a}_i=(a_{i1},\cdots,a_{in}), \qquad \text{where } \boldsymbol{a}_i \in F_1$$

and

$$a_{ij}=\begin{cases}\square, & \text{if } j \in A_i\\ 1, & \text{if } j \notin A_i.\end{cases}$$

Then f^2 is defined to be symmetric with f^1, i.e.,

$$f^2(m_i)=\boldsymbol{b}_i=(b_{i1},\cdots,b_{in}), \qquad \boldsymbol{b}_i \in F_2$$

$$b_{ij}=\begin{cases}\square, & \text{if } j \in A_i\\ 0, & \text{if } j \notin A_i.\end{cases}$$

Note that the correspondence between the families $\mathcal{F}_0$ and the (symmetric) pairs of functions f^1, f^2 is one to one.

Suppose that there exist functions g^1, g^2 which form a WUM code with f^1 and f^2. If there exist $A_i, A_j, A_k \in \mathcal{F}_0$ for which $A_i \triangle A_j \subseteq A_k$, then consider the two series of messages for an arbitrary state x^0 of the WUM:

$$m^1=m_i \qquad m^2=m_k \qquad m^3=m_j \qquad m^4=m_k \qquad m^5=m_i$$

and

$$\hat{m}^1=m_i \qquad \hat{m}^2=m_k \qquad \hat{m}^3=m_j \qquad \hat{m}^4=m_k \qquad \hat{m}^5=m_j.$$

Then it is easy to check that x^4 and also x^5 (i.e., the state of the memory in generations 4 and 5) will be the same for the two series. This implies immediately that it is impossible to find decoding functions that can decide if the fifth message was m_i or m_j, a contradiction. On the other hand, if the family $\mathcal{F}_0$ has the property described in 2) then the foregoing ambiguity can never arise, so we always find g^1 and g^2 to form an error-free WUM code with f^1 and f^2.

Now consider a family

$$\mathcal{F}'=\{A_i | A_i \subseteq \{1,\cdots,n\}, \quad \text{no pairwise different sets } A_i, A_j, A_k \text{ yield } A_i \triangle A_j \subseteq A_k\}.$$

It is conjectured by Erdös and Katona [8, p. 27] that the maximum cardinality of such $\mathcal{F}'$ is obtained by the following construction. Divide $\{1,\cdots,n\}$ into $\lceil n/3 \rceil$ classes of 3 and 2 elements, and let each $A_i \in \mathcal{F}'$ contain exactly one element from each class.

Note that this construction satisfies the slightly more restricted conditions for $\mathcal{F}$ in Lemma 4. This means that if the conjecture of Erdös and Katona is true, then the same is true for such a family $\mathcal{F}$ described in Lemma 4.

Finally, note that the conjectured optimal construction corresponds (in the sense of Lemma 4) to the WUM code construction we gave for Case 3. Then the Erdös–Katona conjecture implies the following (only slightly weaker).

Conjecture: For an alternating and symmetric WUM-code in Case 3, $R(n) \leqq (\log 3)/3 \approx 0.528$.

Case 4: We know that $R(n) < \gamma$ as in all other cases. We know $R(n) \geqq 1/2$ (for every even n, and $R(n) \geqq 1/2 - 1/2n$ for odd n) from the construction of Borden described in Section II. (It is easy to check that this construction is good even in Case 4.) We conjecture that in Case 4 this construction is the best possible, i.e., the best achievable rate for large n is $1/2$ in Case 4. A summary of our knowledge about $\lim_{n\to\infty} R(n)$ in the four cases is given in Table II.

TABLE II
OUR KNOWLEDGE ABOUT $R(n)$ IN THE FOUR CASES INVESTIGATED

Case	Encoder Informed	Decoder Informed	$\lim_{n\to\infty} R(n)$	Best Rate Achieved (For Arbitrarily Large n) By Construction
1	yes	yes	γ	$\gamma \approx 0.694$
2	yes	no	γ	$(\log 58)/11 \approx 0.5325$
3	no	yes	?	$(\log 3)/3 \approx 0.528$
4	no	no	0.5?	0.5

V. A New Proof of Borden's Result $R(n) \leq \gamma$

In conclusion, we present a new proof of the asymptotic upper bound of Borden's theorem. This proof, partially due to Tardos, explains the connection between WUM codes and the Fibonacci sequence.

Proof of Borden's theorem:

Case A: We claim that for alternating WUM codes of length n

$$R(n) < \gamma = \log\left(\frac{1+\sqrt{5}}{2}\right). \tag{1}$$

Consider the ith cell in the WUM. Without loss of generality, we can assume that $i=1$. Its state in the jth generation is x^j. We now shall count the number of all the possible "fates" of this cell during t generations, i.e., the number $|B_t|$ where

$$B_t = \left\{(x^0, x^1, \cdots, x^t) | x^0 \in \{0,1\},\ x^{2l} \in \{0\} \cup \{x^{2l-1}\},\ x^{2l+1} \in \{1\} \cup \{x^{2l}\}\right\}.$$

The conditions for x^{2l} and x^{2l+1} are the formal descriptions of the fact that our WUM code is alternating, i.e., the encoder cannot write 0's in odd or 1's in even generations.

We claim that

$$|B_t| = a_{t+2} \tag{2}$$

where a_n is the Fibonacci sequence defined by

$$a_1 = 1,\ a_2 = 2$$
$$a_k = a_{k-1} + a_{k-2}, \qquad \text{for } k \geqq 3.$$

It is easy to check that $B_0 = \{0,1\}$ and $B_1 = \{(0,0),(0,1),(1,1)$ i.e., $|B_0| = 2$ and $|B_1| = 3$ is true.

We have to show

$$|B_k| = |B_{k-1}| + |B_{k-2}|, \qquad \text{for } k \geqq 2.$$

Let us define the sets B_{k0} and B_{k1} as follows:

$$B_{k0} = \left\{(x^0 = 0, x^1, \cdots, x^k) | x^{2l} \in \{0\} \cup \{x^{2l-1}\},\ x^{2l+1} \in \{1\} \cup \{x^{2l}\}\right\}$$

$$B_{k1} = \left\{(x^0 = 1, x^1, \cdots, x^k) | x^{2l} \in \{0\} \cup \{x^{2l-1}\},\ x^{2l} \in \{1\} \cup \{x^{2l}\}\right\}.$$

It is obvious that

$$B_k = B_{k0} \cup B_{k1}, \qquad \text{while } B_{k0} \cap B_{k1} = \varnothing,$$
$$\text{implying } |B_k| = |B_{k0}| + |B_{k1}|.$$

The condition $x^0 = 0$ does not restrict the value of x^1, whence

$$|B_{k0}| = \left|\left\{(x^1, \cdots, x^k) | x^1 \in \{0,1\},\ x^{2l} \in \{0\} \cup \{x^{2l-1}\},\ x^{2l+1} \in \{1\} \cup \{x^{2l}\}\right\}\right| = |B_{k-1}|.$$

If $x^0 = 1$, then $x^1 = 1$, but there is no restriction on the value of x^2. Thus

$$|B_{k1}| = \left|\left\{(x^2, \cdots, x^k) | x^2 \in \{0,1\},\ x^{2l} \in \{0\} \cup \{x^{2l-1}\},\ x^{2l+1} \in \{1\} \cup \{x^{2l}\}\right\}\right| = |B_{k-2}|.$$

Hence

$$|B_k| = |B_{k-1}| + |B_{k-2}|$$

which proves (2).

The explicit formula of a_k is well-known (see [9, p. 158]):

$$a_k = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right\}. \tag{3}$$

It is clear that the number of all possible fates of the whole memory during t generations, i.e., the number of all the possible vectors of n vectors $(x^0, x^1, \cdots, x^t)$ is exactly $|B_t|^n$.

On the other hand, there are M^t different series of messages during t generations, and each of these must belong to a different fate of the memory or else error-free decoding is impossible. This implies that for every t,

$$M^t \leqq |B_t|^n = (a_{t+2})^n. \tag{4}$$

Then using (3) we obtain for the rate

$$R = \frac{\log M}{n} = \frac{\log M^t}{tn} \leqq \frac{1}{t}\log\left[\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{t+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{t+2}\right)\right].$$

This is true for any t; therefore,

$$R(n) \leqq \lim_{t\to\infty} \frac{1}{t}\log\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{t+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{t+2}\right] = \log\left(\frac{1+\sqrt{5}}{2}\right).$$

The proof of Case A is complete because simple arithmetic shows that equality can never occur for any finite n.

Case B (Tardos [10]): Let $\boldsymbol{b}$ be a t-length sequence of 0's and 1's. We are given a WUM for which at the jth generation we are only allowed to write b_j, i.e., the possible fates of a certain cell of the memory during t generations can be described by the set

$$B_t^{\boldsymbol{b}} = \left\{(x^0, x^1, \cdots, x^t) | x^0 \in \{0,1\},\ x^j \in \{b_j\} \cup \{x^{j-1}\}\right\}.$$

It is easy to see (by induction and using the method of the proof of Case A) that $B_t^{\boldsymbol{b}} \leqq a_{t+2}$ for any fixed binary sequence $\boldsymbol{b}$. This implies that, once we have fixed $\boldsymbol{b}$, the number of all the possible fates of the whole memory is $|B_t^{\boldsymbol{b}}|^n \leqq (a_{t+2})^n$. In fact, we have 2^t possible choices for the sequence $\boldsymbol{b}$. Hence for the number d_t of all the possible fates of the memory during t generations (without predetermined writing directions), we have

$$d_t = \sum_{\boldsymbol{b} \in \{0,1\}^t} |B_t^{\boldsymbol{b}}|^n \leqq 2^t (a_{t+2})^n.$$

There are M^t different sequences of messages during t generations, each of which must belong to a different fate of the memory. Thus

$$M^t \leqq d_t \leqq 2^t (a_{t+2})^n$$

which is the same as (4) with $M/2$ instead of M. Hence writing

$$R = \frac{\log M}{n} = \frac{\log\frac{M}{2} + 1}{n},$$

we can repeat the previous computation (Case A) obtaining

$$R \leqq \log\left(\frac{1+\sqrt{5}}{2}\right) + \frac{1}{n}$$

which asymptotically gives

$$R \leqq \log\left(\frac{1+\sqrt{5}}{2}\right).$$

ACKNOWLEDGMENT

Thanks are due to Gérard Cohen, Philippe Godlewski, and Gábor Tardos for helpful conversations. I am especially grateful to Gérard Cohen for introducing me to the topic and to János Körner for encouragement and help.

REFERENCES

[1] J. M. Borden, "Coding for write-unidirectional memories," submitted to *IEEE Trans. Inform. Theory*.
[2] F. M. J. Willems and A. J. Vinck," Repeated recording for an optical disk," in *Proc. 7th Symp. Information Theory in the Benelux*, May 22–23, 1986, pp. 49–53.
[3] R. L. Rivest and A. Shamir, "How to reuse a "write-once" memory," *Inform. Contr.*, vol. 55, pp. 1–19, 1982.
[4] J. K. Wolf, A. D. Wyner, J. Ziv, and J. Körner, "Coding for a write-once memory," *Bell Syst. Tech. J.*, vol. 63, no. 6, pp. 1089–1112, July-Aug. 1984.
[5] G. D. Cohen, P. Godlewski, and F. Merkx, "Linear binary codes for write-once memories," *IEEE Trans. Inform. Theory*, vol. IT-32, no. 5, pp. 697–700, Sept. 1986.
[6] Z. Baranyai, "On the factorization of the complete uniform hypergraph," in *Infinite and Finite Sets*, A. Hajnal, R. Rado, and Vera T. Sós, Eds. Amsterdam, The Netherlands, and London: North-Holland, 1975.
[7] G. Simonyi, "On write-unidirectional memories," internal rep., Ecole Nationale Supérieure des Télécommunications, Paris, France, 1986.
[8] G. O. H. Katona, "Extremal problems for hypergraphs," *Math. Centre Tracts*, vol. 56, pp. 13–42, 1974.
[9] L. Lovász, *Combinatorial Problems and Exercises*. New York: North-Holland, 1979.
[10] G. Tardos, personal communication.

Estimating the Information Content of Symbol Sequences and Efficient Codes

PETER GRASSBERGER

***Abstract*—Several variants of an algorithm for estimating Shannon entropies of symbol sequences are presented. They are all related to the Lempel–Ziv algorithm and to recent algorithms for estimating Hausdorff dimensions. The average storage and running times increase as N and $N \log N$, respectively, with the sequence length N. These algorithms proceed basically by constructing efficient codes. They seem to be the optimal algorithms for sequences with strong long-range correlations, e.g., natural languages. An application to written English illustrates their use.**

I. INTRODUCTION

Since the development of information theory by Shannon [1], it has been recognized that few good algorithms exist for measuring the information of signals containing strong long-range correlations. To be more specific, let us assume a binary sequence $S = (s_1, s_2, \cdots)$ with $s_i \in \{0,1\}$. Our assumption represents no restriction since any other discrete sequence with a finite alphabet can be encoded in this way, and even the output of continuous dynamical systems can be encoded in such a way that the entropy of the sequence of code symbols ("Kolmogorov entropy" [2], [3]) is independent of the encoding. We assume that the sequence is statistically stationary. Thus, for any sequence S_N of N binary digits, well-defined probabilities $p_N\{S_N\}$ exist for finding S_N starting at any chosen site within S. The Shannon entropy is defined as

$$h = \lim_{N \to \infty} h_N \tag{1.1}$$

with

$$h_N = -\frac{1}{N} \sum_S p_N\{S_N\} \log_2 p_N\{S_N\}. \tag{1.2}$$

Here, we take the logarithm to base 2 to obtain h in bits/digit.

Manuscript received June 1, 1987; revised July 18, 1988.
The author is with the Physics Department, University of Wuppertal, D-5600 Wuppertal 1, Gauss-Strasse 20, West Germany.
IEEE Log Number 8928194.

The quantities h_N are called block entropies. The limit in (1.2) converges always from above, i.e., one always has $h \le h_N$. The latter is very useful in practical applications because it implies that estimating upper bounds on h is very easy.

These bounds are tight if the sequence has no strong long-range correlations. More precisely, $h_n = h$ for all $n \ge N$ if the sequence is an Nth-order Markov chain. However, in realistic cases such as natural languages, DNA sequences, or TV images coded in some sequential form, these correlations are very strong, and h_N converges so slowly that (1.1) is rendered virtually useless.

This fact was understood by Shannon who devised an ingeneously simple method for estimating the information of written English [4]: he randomly erased letters from some text (replacing them by blanks) and presented the mutilated text to students who had not seen it before. If the students were able to guess the meaning of the text in which a percentage p of letters were missing, then these letters were obviously redundant, giving a redundancy $\ge p$. This method and other similar subjective methods [5]–[10] (for a complete survey of the literature, see [10]) have several drawbacks. First, they can be applied only to natural languages since guessing at statistics and grammar would require perfect and comprehensive knowledge. Secondly, since the process is subjective, estimating errors is difficult. Finally, in view of the increasing availability of computers and of computer-readable written text, algorithms that can be used by computers would seem useful.

Up to now, the objective method with the best chances of taking long-range correlations into account has been the algorithm of Lempel and Ziv [11], [12]. Originally constructed to provide an information measure for individual finite sequences, the Lempel–Ziv algorithm is similar in spirit to the Kolmogorov–Solomonoff–Chaitin [13]–[15] algorithmic complexity. However, it was shown [16] that, in the case of statistically stationary strings, it converges to the Shannon entropy as $N \to \infty$. The Lempel–Ziv complexity will be defined in the next section.

As proposed in [11], [12] and implemented, e.g., in [17], the Lempel–Ziv algorithm needs a time of the order of N^2 for a sequence of N symbols. (Algorithms needing running time of the order of N have been proposed [18], [19], but they refer not to the Lempel–Ziv algorithm but the Ziv–Lempel algorithm of [22] which seems considerably less optimal for finite strings, its virtue being that it is much easier to apply. Furthermore, these algorithms seem to attain their claimed asymptotic behavior only when applied to Markov sequences. We are interested in the opposite case where the Markov property cannot be used or is to be tested.) With present-day computers, this restricts one to analyzing $\le 10^4$–10^5 symbols. This is not enough for the long-range correlations in written language. The algorithms presented in Section II need times of the order of $N \log N$, making more serious analyses feasible.

These algorithms can be understood in several ways. In one sense, they are modifications of the Lempel–Ziv algorithm. In another, they are adaptations of algorithms proposed by Badii and Politi and others [21]–[23] for estimating Hausdorff dimensions. In [21]–[23] the dimension of a measure μ supported by a metric space is estimated from the nearest neighbor distances $d(i, j)$ of N points chosen randomly with respect to μ. The formula

$$D = -\lim_{N \to \infty} \frac{\log N}{\langle \log d \rangle} \tag{1.3}$$

is conjectured (here and in what follows, angular brackets denote average values). As pointed out by Farmer [24], the Shannon entropy is just the dimension of the set of symbol sequences, provided one uses a suitable metric: the distance between two

0018-9448/89/0500-0669$01.00 ©1989 IEEE