Sofic Systems and Encoding Data

BRIAN MARCUS

Abstract—**Techniques of symbolic dynamics are applied to prove the existence of codes suitable for certain input-restricted channels. This generalizes the earlier work of Adler, Coppersmith, and Hassner on the same problem.**

INTRODUCTION

RECENTLY, Adler, Coppersmith, and Hassner [1] addressed the problem of encoding digital data from a free (unconstrained) n-ary source to a constrained set of available sequences. Their approach leads to codes of roughly the same complexity as codes previously constructed and, in fact, shares some ideas with P. Franaszek's approach [2]–[5]. However, what is new in [1] is a *proof* of the existence of codes (and an explicit algorithm for generating them), which guarantees several desirable properties (in particular, state independence and limited error propagation in decoding). The main assumption of [1] is that the constrained set of available sequences is generated by some discrete, noiseless, input-restricted channel with finite memory ("subshifts of finite type"). We treat here the case of possibly infinite memory ("sofic systems"). These systems are described by labeling the edges of a directed graph. It is important to remember that any one of these systems can be described by several different graphs. We exploit this fact.

All of these codes are synchronous. That is, the asymptotic ratio of input (unconstrained) bits to output (constrained) symbols is a constant, p/q, independent of the input sequence. The number

$$\boldsymbol{R} = \frac{p}{q}\log_2(n)$$

is called the rate of the code. Of course, given the constrained set of sequences, one desires to have codes of high rate. But Shannon's classical theorem [6] gives an upper bound on the rate, namely $\boldsymbol{R} \leq \boldsymbol{C}$, where $\boldsymbol{C}$ is the capacity of the channel that generates the constrained set of sequences. The point of [1] was to prove that, subject only to the condition $\boldsymbol{R} \leq \boldsymbol{C}$, there exist codes that satisfy the following.

1) They are synchronous.
2) They have limited look-ahead in encoding.
3) They have limited look-ahead and look-back in decoding.
4) They are stationary (i.e., if $\boldsymbol{R} = (p/q)\log_2(n)$, then the coding rule is invariant under shifting the input by p positions and shifting the output by q positions).

It is impossible to guarantee, in general, limited look-back in encoding, as well as the above. One can use eigenvectors to make estimates on the amount of look-back, look-ahead required, but we do not deal with that here.

Reference [1] was motivated by the problems of encoding computer data on a magnetic medium. Channels of finite memory arise naturally in attempting to control intersymbol interference and clock drift. The coding properties listed above are all important in this context; in particular, 3) guarantees that the very few hardware errors made will not be propagated into many decoding errors.

In some products, it is also desirable to shape the spectrum of read/write signals in some way. This generally leads to channels with infinite memory. An example of this type, described in Section I-D was the motivation for our study.

After providing some background information in Section I, we prove in Section II that the results of [1] generalize to the infinite memory case under the assumption $\boldsymbol{R} < \boldsymbol{C}$. This is done by approximating any discrete, noiseless input-restricted channel with infinite memory by such channels of finite memory, (i.e., throw away some bad blocks in the channel so that what is left has finite memory but large capacity), and then applying [1].

In the case $\boldsymbol{R} = \boldsymbol{C}$, there is no room to spare, so no blocks can be thrown out. Nevertheless, under an additional restriction, we do in principle get codes with properties 1), 2), 3), and a weak version of the stationary condition 4) mentioned above. Namely, the coding rule is invariant under shifting the input by kp positions and shifting the output by kq positions for some k. (See Section IV.)

While our proofs are constructive, a blind application of the algorithms contained in them leads, in general, to codes of unacceptable block length. Our main purpose is to *prove* the existence of codes and to give a skeleton scheme for finding reasonable codes.

The purely mathematical content of this work can be summarized as follows (see Section I for background).

Theorems: Let $\boldsymbol{S}$ be a sofic system, and let $h(\boldsymbol{S})$ denote its entropy. Let n be a positive integer.

a) If $h(\boldsymbol{S}) > \log(n)$, then $\boldsymbol{S}$ factors continuously onto the full n-shift (Corollary 1, presented in Section II).

Manuscript received September 7, 1982; revised November 1, 1984. This work was supported in part by the National Science Foundation under Grants MCS-8001796 and MCS-8301246. The material in this paper was presented at the AMS National Meeting, Cincinnati, OH, January 1982.

The author was with the Mathematics Department, University of North Carolina, Chapel Hill, NC 27514. He is now with the IBM Research Laboratory, San Jose, CA 95193, USA.

0018-9448/85/0500-0366$01.00

b) If $h(\boldsymbol{S}) < \log(n)$, then $\boldsymbol{S}$ factors continuously, finite-to-one into the full n-shift (Theorem 1 (see Section III)).

c) If $h(\boldsymbol{S}) = \log(n)$ and $\boldsymbol{S}$ is almost of finite type (see Definition 4), then some power of $\boldsymbol{S}$ factors continuously, finite-to-one onto the same power of the full n-shift (Theorem 2 (see Section IV)).

Note: We do not know if part c) can be improved.

The main idea for applications is that the set of allowable sequences to be recorded on a magnetic medium is represented by a sofic system and the computer data is represented by a full shift. The factor maps in the aforementioned Theorems (a) and c)) provide the codes; b) is used as a starting point for the proof of c).

The procedures here and in [1] were developed from techniques used to study the classification problem for smooth dynamical systems. The relationship is as follows. The phase space of the dynamical system is partitioned into a finite number of pieces, each piece labeled by a symbol; then, by observing the itinerary of an orbit relative to the pieces of the partition, the orbits are coded into sequences of symbols. The motion of the dynamical system is then reflected by the shift map on the space of sequences. The classification problem is, thus, turned into a shift-invariant coding problem on these sequence spaces. The connection between magnetic recording and dynamical systems was discovered by Hassner [7].

I. Background and Examples

We briefly summarize the necessary background. For more details, we refer the reader to [1] and the references therein. We now describe the types of sources of sequences (subshifts) that we use and codes to be constructed (factor maps).

A. Subshifts

Let A be a finite set, with n elements (thought of as a set of symbols, or states, or as an alphabet). The *full n-shift* Σ_n is the set of bi-infinite sequences

$$\{\cdots x_{-2}x_{-1}x_0x_1x_2\cdots : \text{each } x_i \in A\}$$

with a distinguished 0th coordinate. This is also known as the free source.

The shift map is defined as

$$\sigma\colon \Sigma_n \to \Sigma_n, \qquad \sigma(x) = y \text{ where each } y_i = x_{i+1}.$$

σ is continuous with respect to the natural metric. This map will be important for coding purposes because it is a convenient way of expressing the stationarity of the coding schemes.

By a *subshift*, we mean the restriction of σ to a closed σ-invariant subset, Λ, of Σ_n. This means that there is a collection (possibly infinite) C of finite words in the alphabet A such that (x belongs to Λ) $\Leftrightarrow$ (each finite block of x belongs to C). Thus the set Λ will really represent a collection of available messages.

Example 1: Λ_1 is the space of sequences with alphabet $A = \{1, 2\}$, where 2's are required to be isolated (called the *golden mean system*).

Example 2: Λ_2 is the space of sequences with alphabet $\{a, b\}$, where b's are required to appear in blocks of even length between two a's (called the *even system* [8]).

By a Λ-block, we mean a block that appears in some point of Λ. By a k-block, we mean a Λ-block of length k (Λ being understood by context).

Since a subshift Λ is defined to be shift invariant, the shift map σ naturally restricts to a map $\sigma\colon \Lambda \to \Lambda$.

Sometimes we use Λ to refer to either Λ or Λ together with σ/Λ, the restriction of the map σ to the set Λ.

B. Factor Maps

Let Λ_1 and Λ_2 be two subshifts with possibly different alphabets. Let $g_1\colon \Lambda_1 \to \Lambda_1$ and $g_2\colon \Lambda_2 \to \Lambda_2$ be two continuous maps. A *factor map* from (Λ_1, g_1) to (Λ_2, g_2) is a continuous map $\pi\colon \Lambda_1 \to \Lambda_2$ such that $\pi g_1 = g_2\pi$. Usually we shall be interested in the case $g_1 = \sigma^i$ and $g_2 = \sigma^j$ for some i and j. When we refer to a factor map $\pi\colon \Lambda_1 \to \Lambda_2$, we will assume that the maps involved are, in fact, $g_1 = \sigma$ and $g_2 = \sigma$ unless otherwise specified.

While the definition of a factor map has an abstract form, it is really a very concrete idea: it is simply a sliding block code (see [9], [1, p. 8]).

Example 3: Let Λ_1 be the golden mean system (isolated 2's), and let Λ_2 be the even system (even b's). Let π^* be the map

$$\pi^*\colon \{2 - \text{blocks of } \Lambda_1\} \to \{\text{symbols of } \Lambda_2\}$$

defined by

$$\pi^*(11) = a$$
$$\pi^*(21) = b$$
$$\pi^*(12) = b.$$

Then define the factor map $\pi\colon \Lambda_1 \to \Lambda_2$ by

$$\pi(\cdots x_{-2}x_{-1}x_0x_1x_2\cdots)$$
$$= \cdots \pi^*(x_{-2}x_{-1})\pi^*(x_{-1}x_0)\pi^*(x_0x_1)\pi^*(x_1x_2)\cdots.$$

So, for example,

$$\pi(\cdots 211211121211\cdots) = \cdots babbaabbbba\cdots.$$

This is a factor map from Λ_1 onto Λ_2 (more properly, (Λ_1, σ) onto (Λ_2, σ)).

In this example, π is a 2-block map. In general, a k-block factor map is a sliding block code generated by a map

$$\pi^*\colon \{k - \text{blocks of } \Lambda_1\} \to \{\text{symbols of } \Lambda_2\}$$

So, for $l \geq k$, the expression $\pi(x_1 \cdots x_l)$ makes sense:

$$\pi(x_1 \cdots x_l) = \pi^*(x_1 \cdots x_k)$$
$$\cdot\pi^*(x_2 \cdots x_{k+1}) \cdots \pi^*(x_{l-k+1} \cdots x_l).$$

Factor maps which are 1-1 and onto are called *conjugacies*. They play a very special role. If a conjugacy from Λ_1

to Λ_2 exists, we say that Λ_1 and Λ_2 are conjugate. The idea is that two conjugate subshifts are essentially the same even if they produce literally different sequences. Notice that the generating map π^* of a conjugacy may not be 1-1, although the conjugacy itself must be 1-1 (see Fig. 4 following).

Let Λ be a subshift, and let k be a positive integer. Let C_k denote the set of all k-blocks of Λ. In $(C_k)^Z$, there are two subshifts intimately related to Λ.

Example 4: Define

$$\phi_1\colon \Lambda \to (C_k)^Z$$

$$\phi_1(\cdots x_{-1}x_0x_1\cdots) = \cdots(x_{-1}\cdots x_{k-2})(x_0\cdots x_{k-1})(x_1\cdots x_k)\cdots.$$

Note that the blocks here overlap. The image of ϕ_1 is a subshift, conjugate (via ϕ_1) to Λ (more properly, (Λ, σ)). This subshift is called the *higher block system* (see [1, p. 7]) for Λ and is one of many different and convenient ways that we can represent a subshift.

Example 5: Define

$$\phi_2\colon \Lambda \to (C_k)^Z$$

$$\phi_2(\cdots x_{-1}x_0x_1\cdots) = \cdots(x_{-k}\cdots x_{-1})(x_0\cdots x_{k-1})(x_k\cdots x_{2k-1})\cdots.$$

Note that the blocks here do not overlap. The image of ϕ_2 is a subshift, $(\phi_2(\Lambda), \sigma)$, called the kth power, which is conjugate (via ϕ_2) to (Λ, σ^k). This is the standard way of representing a power of a subshift map as a subshift map in its own right.

C. Special Subshifts: SSFT and Sofic Systems

We are mostly interested in subshifts of finite type and sofic systems. A subshift, Λ, is of *finite* type (SSFT) if there is a positive integer k and a collection of k-blocks C such that

$$\Lambda = \{x \in A^Z\colon \text{for all } i,\ x_{i+1}x_{i+2}\cdots x_{i+k} \in C\}.$$

In other words, Λ is the set of points all of whose k-blocks are prescribed by C.

This really means that the Λ-blocks are determined by *finite memory* in the following sense: Given a symbol s and Λ-block w, in order to know whether the concatenated block ws is a Λ-block, one need only know the last k symbols of w.

If $k = 2$, then one constructs an $n \times n$ matrix

$$A_{ij} = \begin{cases} 1, & \text{if } ij \in C \\ 0, & \text{if } ij \notin C \end{cases}.$$

(Here we are thinking of the state set A as $\{1, 2, 3, \cdots, n\}$.) In this case ($k = 2$), the SSFT is denoted $\{A\}$. By a simple recoding (via the higher k-block system), every SSFT may be described as an $\{A\}$ (with perhaps a much larger set of states).

As is standard, one may represent an SSFT $\{A\}$ as the set of all bi-infinite walks on a directed graph as follows. The states are the elements of A; one draws an *edge* from i to j if and only if $A_{ij} = 1$. Thus, the point $x = (\cdots x_{-1}x_0x_1x_2\cdots)$ corresponds to a walk that at time i is at state x_i. The $\{A\}$-blocks correspond to the paths of this graph. For example, if $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, then the graph is as shown in Fig. 1. This SSFT $\{A\}$ is simply the golden mean system described before, because the only restriction that one encounters while walking on the graph is that two's are isolated.

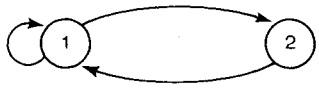

Fig. 1. Golden mean system.

The higher block systems of SSFT's are also represented by graphs in a very simple way. Namely, the 2-block system of $\{A\}$ is represented by the *edge graph* of the original graph of $\{A\}$; the 3-block system is represented by the edge graph of the edge graph, etc. For example, the 2-block system of the golden mean system is generated by the graph in Fig. 2, whose vertices represent the edges of the original graph.

The $n \times n$ matrix of all ones generates the full n-shift Σ_n, which is, of course, an SSFT.

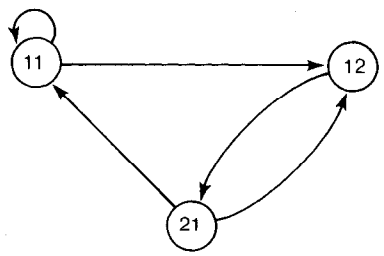

Fig. 2. Edge graph of golden mean system.

One typically assumes that all SSFT's are given by irreducible matrices (i.e., for all i, j there exists an $n = n(i, j)$ such that $A_{ij}^n > 0$) or, perhaps even stronger, that A is *aperiodic* (i.e., there exists an n such that for all $i, j, A_{ij}^n > 0$). Any SSFT that is not aperiodic can be studied in terms of its components [1, p. 10].

A subshift Λ is said to be *sofic* if there is an SSFT $\{A\}$ and a factor map π from $\{A\}$ onto Λ. Of course, every SSFT is sofic (let π = identity), but sofic systems are much more general.

In the definition of sofic system, by replacing $\{A\}$ by a higher block system, one may assume that π is a 1-block map or, equally well, a 2-block map. From this point of view, a sofic system is a subshift obtained by labeling the vertices if 1-block (or the edges if 2-block) of a directed graph. For example, the edge labeling shown in Fig. 3 presents a sofic system (the even system) as a 2-block factor of the golden mean system. The even system is not an SSFT: in order to know whether an a can follow a string of b's, one has to know when an a previously occurred; this, however, requires infinite memory. This

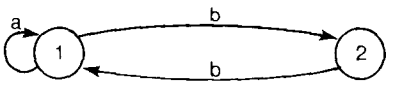

Fig. 3. Even system.

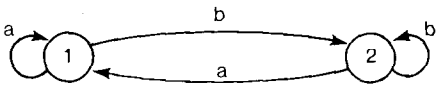

Fig. 4. Full 2-shift.

means that it can never be described by a discrete noiseless channel of finite memory.

On the other hand, the edge labeling shown in Fig. 4 presents the full 2-shift that is an SSFT.

A given sofic system or SSFT may be presented as labelings of a graph (or different graphs) in many different ways (e.g., as in Fig. 4). This is good; we exploit this flexibility.

A test for deciding whether a sofic system is SSFT can be found in [10].

D. Motivating Example

Let c be a positive integer. Let C be the set of all blocks $w_1 \cdots w_n$ with alphabet $\{+1, -1\}$ such that

$$\left| \sum_{i=1}^{n} w_i \right| \leq c.$$

The subshift generated by these blocks is called a charge-constrained system and is denoted by Λ_c. This is simply the set of sequences whose running sums are bounded by c.

The graph shown in Fig. 5 presents Λ_4 as a sofic system. A general graph of the type above presents any Λ_c as a sofic system. These systems are not SSFT's since, for example, if $c = 4$ and w is the block

$$111 - 11 - 11 - 11 \ldots - 11$$

(i.e., 111 followed by arbitrarily many concatenations of -11), then $-1w1$ is in Λ_4 but $1w1$ is not. (So that the concatenability of the symbol 1 depends on infinite memory.)

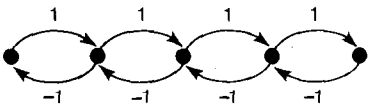

Fig. 5. Charge-constrained system (charge ≤ 4).

If one thinks of sequences in Λ_c as generating square waveforms, then the sequences all have a sharp null at dc. This is frequently desired in applications.

One can also add run-length limits to the charge constraints by requiring that the run lengths of both $+1$'s and -1's are all bounded below by some positive integer D and above by some positive integer K. These systems are called charge-constrained run-length limited systems. They are denoted as $\Lambda_{d,k,c}$, where $d = D - 1$, $k = K - 1$, and c is the charge constraint above. These systems are important in magnetic recording [11], [12]. While the run-length limits are SSFT in nature, the systems $\Lambda_{d,k,c}$ are sofic and not SSFT (the latter because of the charge constraint).

E. Entropy

The entropy $h(\Lambda)$ of a subshift Λ is simply the asymptotic growth rate of the number of k-blocks of Λ (as $k \to \infty$). For an irreducible SSFT $\{A\}$,

$$h(\{A\}) = \log(\lambda),$$

where λ is the largest eigenvalue of A and the log is to base 2. From this, it follows that the entropy of the golden mean system is the log of the largest eigenvalue (of $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$), which happens to be the golden mean itself.

The following proposition is well known in symbolic dynamics.

Proposition 1 [13, p. 9]: If Λ_1 and Λ_2 are subshifts and π: $\Lambda_1 \to \Lambda_2$ is an onto factor map that is either finite-to-one or 1-1 almost everywhere, then $h(\Lambda_1) = h(\Lambda_2)$. (In particular, entropy is conjugacy invariant.)

Thus, one can compute the entropy of a sofic system by realizing it as a finite-to-one image of an SSFT. For example, the factor map in Example 3 is at most 2 to 1 (in fact, all points have exactly one preimage except $\cdots bbb \cdots$), and so the entropy of the even system is the log of the golden mean as well.

The entropy we use here was called capacity by Shannon [6] and is called topological entropy in dynamical systems.

F. Special Factor Maps: Right Resolving Maps

For a subshift Λ with alphabet A and $a \in A$, we denote

$$F_\Lambda(a) \equiv \{a' \in A: aa' \text{ is a 2-block of } \Lambda\}.$$

This is the follower set of a.

Definition 1: Let Λ_1 and Λ_2 be subshifts. A factor map π: $\Lambda_1 \to \Lambda_2$ is called *right resolving* if $\pi(a_1a_2) = \pi(a_1a_2')$ implies $a_2 = a_2'$ (i.e., knowledge of a_1 and $\pi(a_1a_2)$ determine a_2). Of course, this makes sense only if π is a 1-block or 2-block factor map. (This is essentially [1, def. 3.2] with parameters 1, 0, 1.)

Now suppose that $\{A\}$ is an SSFT, S is a sofic system, and π: $\{A\} \to S$ is a 2-block map. Then, as before, π is simply a labeling of the edges of the graph of A. To say that π is right resolving means that for each vertex, the outgoing edges are all labeled differently, i.e., the labeling is a Shannon graph [14]. Every sofic system can be realized in this way [10], [14], [15]. This will be used in the next section.

An important use of right resolving maps is the construction of codes. Consider an SSFT $\{A\}$, where A has row sum n for some positive integer n; this means that coming out of each vertex of the graph of A there are exactly n edges. For each vertex, one labels the n edges $1, \cdots, n$: this defines a right resolving map π: $\{A\} \to \Sigma_n$. Now one codes the free n-ary source into $\{A\}$ by starting at some arbitrary state in the graph of A and following the labels.

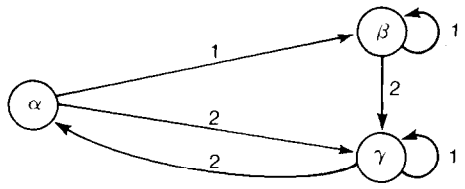

Fig. 6. Simple Code.

Fig. 7. Example of encoding (Fig. 6).

In Fig. 6 we have exactly this situation (with $n = 2$). The encoder (starting at state α) derived from this factor map would, for example, encode as shown in Fig. 7. The decoder is given by the factor map. In this type of coding, the encoder has no look-ahead (but possibly infinite look-back) and the decoder looks ahead one position and does not look back at all.

The idea of [1, Theorem 6.1] was to code the free n-ary source into any SSFT $\{\boldsymbol{A}\}$ with $h(\{\boldsymbol{A}\}) \geq \log(n)$ in much the same way. Namely, first, they construct an SSFT $\{\boldsymbol{B}\}$, conjugate to $\{\boldsymbol{A}\}$, with all rows sums of $\boldsymbol{B}$ at least n (see Section III). So, in the graph of $\boldsymbol{B}$, coming out of each vertex there are at least n edges, and one labels n of them by the distinct numbers $1, \cdots, n$; this yields a right resolving factor map from an SSFT sitting in $\{\boldsymbol{B}\}$ onto Σ_n. One codes Σ_n into this SSFT as described above—namely, starting at an arbitrary vertex, one follows the labels. Then use the conjugacy from $\{\boldsymbol{B}\}$ to $\{\boldsymbol{A}\}$ to code into $\{\boldsymbol{A}\}$. Such a code has all of the desired properties and has rate $\boldsymbol{R} = (1/1)\log_2(n)$.

We will make use of the following.

Proposition 2: Right resolving maps are finite-to-one.

II. Approximation

Definition 2: Let Λ_1 and Λ_2 be subshifts, and let π: $\Lambda_1 \to \Lambda_2$ be a 1-block factor map. A *resolving block* is a Λ_2-block $s = s_1 \cdots s_t$ for which there exists an $i \in [1, t]$ such that if $u \equiv u_1 \cdots u_t$ and $v = v_1 \cdots v_t$ are Λ_1-words with $\pi(u) = s = \pi(v)$, then $u_i = v_i$.

If π is right resolving and has a resolving block, then the i above can be chosen to be t. Also, a finite-to-one factor map is 1-1 almost everywhere if and only if it has a resolving block (see [13, Theorem 3.33]).

Proposition 3: Let $\boldsymbol{S}$ be a sofic system. Then there are SSFT's $\{\boldsymbol{B}_1\}, \{\boldsymbol{B}_2\}, \{\boldsymbol{B}_3\}, \cdots$ such that

1) each $\{\boldsymbol{B}_l\} \subseteq \boldsymbol{S}$;
2) $\sup_l h(\{\boldsymbol{B}_l\}) = h(\boldsymbol{S})$.

Remark: Here we are approximating a sofic system from the *inside* in terms of entropy. This cannot be done in general for arbitrary subshifts.

Proof: By definition, there is an SSFT $\{\boldsymbol{A}\}$ and a factor map π from $\{\boldsymbol{A}\}$ onto $\boldsymbol{S}$. By [15], [10] we may assume that $\{\boldsymbol{A}\}$ is irreducible, that π is a right resolving 1-block factor map, and that π has a resolving block $s = s_1 \cdots s_t$. Thus, if

$$u = u_1 \cdots u_t \quad \text{and} \quad v = v_1 \cdots v_t \tag{2.1}$$

are $\{\boldsymbol{A}\}$-words with $\pi(u) = s = \pi(v)$ then $u_t = v_t$.

Claim: Let $\{\boldsymbol{A}_l\}$ denote the SSFT determined by all $\{\boldsymbol{A}\}$-words $u = u_1 \cdots u_l$ of length l such that $\pi(u)$ is a resolving block. We claim that $\pi|_{\{\boldsymbol{A}_l\}}$ is 1-1.

Proof of Claim: Let $x, y \in \{\boldsymbol{A}_l\}$ with $\pi(x) = \pi(y)$. So, for each i, $\pi(x_{i+1} \cdots x_{i+l}) = \pi(y_{i+1} \cdots y_{i+l})$ is a resolving block. Thus, by (2.1) for each i, $x_{i+l} = y_{i+l}$. So $x = y$ and thus $\pi|_{\{\boldsymbol{A}_l\}}$ is 1-1.

Let $\{\boldsymbol{B}_l\} = \pi(\{\boldsymbol{A}_l\})$. By the claim, $\{\boldsymbol{B}_l\}$ is conjugate to $\{\boldsymbol{A}_l\}$ (via π) and is therefore an SSFT inside $\boldsymbol{S}$.

Next we show that $\sup_l h(\{\boldsymbol{B}_l\}) = h(\boldsymbol{S})$. This rests on the observation that any block with a resolving subblock in it is itself a resolving block. Thus, intuitively, most long $\boldsymbol{S}$-blocks will be resolving. We make this precise.

Fix an $\{\boldsymbol{A}\}$-word $u = u_1 \cdots u_k$ such that $\pi(u)$ is a resolving block. Let

$$p_i \equiv (\boldsymbol{A}^i)_{u_k, u_1}.$$

So p_i is the number of $\boldsymbol{A}$-admissible $(i+1)$-blocks beginning with u_k and ending with u_1. It is well known [13, Theorem 3.10] that since $\boldsymbol{A}$ is irreducible,

$$\overline{\lim_{i \to \infty}} \frac{\log(p_i)}{i} = h(\boldsymbol{A}). \tag{2.2}$$

Now let r and l be positive integers, and assume that l is even and

$$r > l > 2k.$$

Let $U_{r,l} = \{\{\boldsymbol{A}\}\text{-words } v = v_1 \cdots v_r \text{ that have } u \text{ appearing periodically with period } l/2\}$:

$$\underbrace{u_1 \cdots u_k}_{l/2} \quad \underbrace{u_1 \cdots u_k}_{l/2} \quad u_1 \cdots u_k.$$

Then

$$\#U_{r,l} \geq (p_{l/2-k})^{[2r/l]}. \tag{2.3}$$

Moreover, if $v \in U_{r,l}$, then every subblock z of $\pi(v)$ with length l contains $\pi(u)$ as a subblock; whence v is an $\{\boldsymbol{A}_l\}$-block. So the number of $\{\boldsymbol{A}_l\}$-blocks of length r is at least $\#U_{r,l}$.

Thus, by (2.3),

$$h(\{\boldsymbol{A}_l\}) \geq \overline{\lim_{r \to \infty}} \frac{\log(p_{(l/2)-k})^{[2r/l]}}{r} = \frac{\log p_{(l/2)-k}}{l/2}.$$

Thus, since $\{\boldsymbol{B}_l\}$ is conjugate to $\{\boldsymbol{A}_l\}$

$$\sup_l h(\{\boldsymbol{B}_l\}) \geq \sup_l \frac{\log p_{(l/2)-k}}{l/2} \geq h(\{\boldsymbol{A}\}) = h(\boldsymbol{S}),$$

the latter inequality because of (2.2), and the latter equality because π is a finite-to-one map from $\{\boldsymbol{A}\}$ onto $\boldsymbol{S}$ (see Propositions 1 and 2).

Example 6: We give a very simple example of Proposition 3. Let S be the sofic system given by Fig. 8. Intrin-

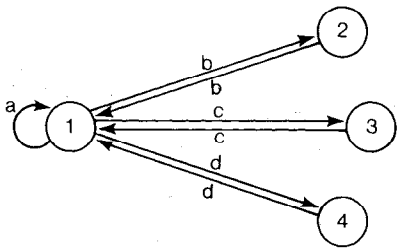

Fig. 8. Example of approximation.

sically, S is the set of sequences on symbols a, b, c, d such that b's, c's, and d's appear only in blocks of even length. It is easily checked that $h(S) > \log 2$. (To see this, let A be the matrix of the SSFT defined by Fig. 8, and observe that the vector $x_1 = 2$, $x_2 = x_3 = x_4 = 1$ satisfies $Ax \geq 2x$ and equality does not hold in the first component; this means that $h(S) = h(\{A\}) > \log 2$ (see [16]).) So Proposition 3 guarantees that there is an SSFT $\{B\}$ inside S with $h(\{B\}) > \log 2$ as well. One can then apply [1, Theorem 6.1], as roughly described in our Section I-F, to encode the free binary source into $\{B\}$—hence into S. In Section IV, we give another way of constructing such a code.

Now the SSFT $\{B\}$ cannot be obtained by simply eliminating some edges of the Fig. 8 graph (any SSFT obtained in that way will have zero entropy). However, one can get $\{B\}$ by eliminating some edges in the 2-block system. We indicate this as follows.

The 2-block system is represented by Table I. The states are the 2-blocks of the original SSFT $\{A\}$: 11, 21, 31, 41, 12, 13, and 14; the arrows indicate the edges, and the letters in parentheses indicate the labels of states that define the factor map onto S.

TABLE I

11 (a) → 11, 12, 13, 14
21 (b) → 11, 12, 13, 14
31 (c) → 11, 12, 13, 14
41 (d) → 11, 12, 13, 14
12 (b) → 21
13 (c) → 31
14 (d) → 41

Now, eliminating the edges (from Table I),

$$21 \to 12$$
$$31 \to 13$$
$$41 \to 14,$$

it can be seen that the SSFT $\{A'\}$ defined by the remaining table (not the labels) has entropy $> \log 2$ (the vector $x_{11} = x_{21} = x_{31} = x_{41} = 2$, $x_{12} = 1$, $x_{13} = 1$, $x_{14} = 1$ satisfies $A'x \geq 2x$, and equality does not hold in the 11 component) [16]. Moreover, the labelings restricted to $\{A'\}$ define a 1-1 map and therefore an SSFT $\{B\}$ inside the sofic system S. (Intrinsically, $\{B\}$ is the set of sequences such that b's, c's, and d's appear only in blocks of 2.) One actually encodes the free binary source (using [1]) into $\{A'\}$ and then composes with the conjugacy defined by the labeling.

Now let S be an arbitrary sofic system. Let n, p, and q be positive integers such that $h(S) > (p/q)\log(n)$, equivalently

$$qh(S) > \log(n^p). \tag{2.4}$$

The left side of (2.4) is the entropy of the subshift (S, σ^q), which is a sofic system in its own right called T. One can then apply Proposition 3 to T to get an SSFT $\{B\} \subset T$ with $h(\{B\}) > \log(n^p)$. Applying [1] to $\{B\}$ (as in Section I-F) one gets a code from the free n^p-ary source into the system defined by T. If one interprets T as (S, σ^q) and Σ_{n^p} as (Σ_n, σ^p) (via Example 5), one gets a code from the free n-ary source into the system S that is invariant under shifting by p positions in the free source and q positions in S.

We now use these ideas to prove a general factor theorem.

Corollary 1: Let S be a sofic system with $h(S) > \log(n)$. Then there is a factor map from S onto the full n-shift, Σ_n.

Proof: By Proposition 3 we get an SSFT $\{B\} \subset S$ with $h(\{B\}) > \log(n)$. Then [1, Theorem 6.1] provides a factor map π from an SSFT $\{C\} \subset \{B\}$ onto Σ_n. Since π is a sliding k-block code, one can simply extend π to all of S by making arbitrary assignments on those k-blocks of S, which are not k-blocks of $\{C\}$, to symbols of $\{1, \cdots, n\}$. (Here it is important that the range is a full (unconstrained) shift—so that the arbitrary assignments will stay within the range—see [17] for more on this.)

III. State Splitting

To construct codes from fairly arbitrary systems to full shifts, we need a good standard form. Let $\{A\}$ be an irreducible SSFT. In [18], we proved that if $h(\{A\}) = \log n$, $n \in Z^+$ there is a matrix B such that $\{B\}$ is conjugate to $\{A\}$ and each row sum of B is n. Adler *et al.* proved the analogous result for $h(\{A\}) \geq \log n$; this is presented in the following proposition.

Proposition 4 ([1, Theorem 6.1]): Let $\{A\}$ be an irreducible SSFT with $h(\{A\}) \geq \log n$, $n \in Z^+$. Then there is a matrix B such that $\{B\}$ is conjugate to $\{A\}$ and each row sum of B is at least n.

Now we prove the following.

Proposition 5: Let $\{A\}$ be an irreducible SSFT with $h(\{A\}) \leq \log(n)$. Then there is a matrix B such that $\{B\}$ is conjugate to $\{A\}$ and each row sum of B is at most n.

Remark: In all of these results, the set of column sums is not changed. So one can get conjugate representations with the correct row and column sums simultaneously. We conjecture that,[1] given A with $\log(n) < h(\{A\}) < \log(n + 1)$, there is a conjugate representative B with all row (and column) sums in the set $\{n, n + 1\}$ (i.e., that Propositions 4 and 5 can be done simultaneously!).

Before proving Proposition 5, we need to establish the following notion.

[1] This was proved by Joel Friedman in a paper to appear in the *Proceedings of the American Math Society* entitled "A Note on State Splitting."

State Splitting [18], [19], [20]: Let $\{A\}$ be an SSFT defined by states A and matrix A. Fix $v \in A$ and a partition of the follower set $F_A(v) = E_1 \cup E_2$ into two pieces. We construct a *new* directed graph by replacing the vertex v by two new vertices, v_1 and v_2. Each edge that terminated at v is replaced by two edges: one terminating at v_1 and the other at v_2. Each edge that emanated from v and terminated at a vertex $p \in E_i$ $(i = 1, 2)$ is replaced by an edge that emanates from v_i and terminates at p. If $p = v$ (and say $i = 1$), then there was a loop at v that is replaced in the new graph by a loop at v_1 and an edge from v_1 to v_2. The new vertex set is $A' = (A - \{v\}) \cup \{v_1, v_2\}$ and the new matrix denoted A'.

For example, if in the graph shown in Fig. 9 $E_1 = \{v, w\}$ and $E_2 = \{u\}$, then the new graph is as shown in Fig. 10.

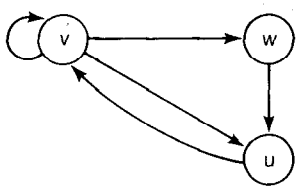

Fig. 9. Before splitting.

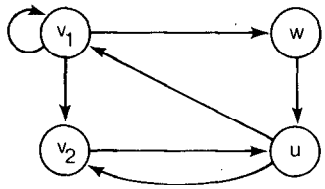

Fig. 10. After splitting.

Proposition 6: Let A' be a matrix obtained from A by splitting the state v. Then $\{A\}$ and $\{A'\}$ are conjugate.

Proof of Proposition 6: This is well known; just define

$$\pi^*(w) = \begin{cases} w, & \text{if } w \in A - \{v\} \\ v, & \text{if } w = v_1 \\ v, & \text{if } w = v_2. \end{cases}$$

Then π^* generates a conjugacy π.

Lemma 1 ([18, Lemma 4]: Let n be a positive integer, and for each $i = 1, \cdots, n$ let s_i be a positive integer. Then there exists an $E \subseteq [1, n]$ such that n divides $\sum_{i \in E} s_i$.

Proof of Lemma 1: Either $\{s_1, s_1 + s_2, \cdots, s_1 + s_2 \cdots + s_n\}$ are all distinct mod n or two of them are congruent mod n. In the former case, one of those sums must be divisible by n. In the latter case, the difference between two of the sums must be divisible by n.

We may now proceed to the proof of Proposition 5.

Proof of Proposition 5: Let ξ be a positive eigenvector of A (corresponding to the positive eigenvalue of largest modulus λ). By virtue of the entropy assumption, $\lambda \leq n$. Thus, by approximating ξ by a rational vector and clearing denominators, one gets a positive integral vector r such that

$$Ar \leq nr.$$

(Note: If $h(\{A\}) = \log n$, then ξ may already be assumed to be positive integral.) We call such an r a *positive integral approximate eigenvector.*

Fix $v \in A$ with $\#F_A(v) > n$. (If v does not exist, we are finished already.) Let U be any subset of $F_A(v)$ with exactly n elements. By Lemma 1, there is a subset $E \subseteq U$ such that

$$n \text{ divides } \sum_{j \in E} r_j.$$

Do state splitting, as described previously, with

$$E_1 = E$$

$$E_2 = F_A(v) - E.$$

Since $E \subseteq U \subsetneq F_A(v)$, E_2 must be nonempty. Define a vector r' as

$$r'_{v_1} = \frac{1}{n}\left(\sum_{j \in E} r_j\right)$$

$$r'_{v_2} = r_v - r'_{v_1}$$

and for $i \in A - \{v\}$

$$r'_i = r_i.$$

Then, one easily sees that r' is a positive integral approximate eigenvector for A'. Clearly r and r' satisfy

$$\sum_{i \in A} r_i = \sum_{i \in A'} r'_i.$$

Thus, since $\#A' > \#A$, the state splitting process can only be repeated a finite number of times, and so eventually we must obtain a matrix B with $\{B\}$ conjugate (by Proposition 6) to $\{A\}$ and for each state v of $\{B\}$, $\#F_B(v) \leq n$, as desired.

Remarks:

1) The end result of this gives a conjugacy between $\{A\}$ and $\{B\}$. It would be good to know the best possible estimate on the size of the block length of the conjugacy in general and also in various special cases that arise in practice. The important point in the above proof is that one can find a state v and a proper subset $E \subsetneq F_A(v)$ such that $\sum_{j \in E} r_j$ is divisible by n. One can split any vertex v with this property and thereby obtain shortcuts in the method.

2) The proof of Proposition 4 follows similar lines except that $r'_{v_2} > 0$ is not automatic unless one splits a vertex v with maximal r-component and such that $F_A(v)$ has an element whose r-component is not maximal. This will force the existence of the set E above and will also force $r'_{v_2} > 0$.

The problem with the notion of a right resolving map is that it is not invariant under conjugacy. The following notion is invariant.

Definition 3: Let π be a factor map from Λ_1 to Λ_2. We say that π is *right closing* if it never identifies a pair of negatively asymptotic points, as shown in Fig. 11. More

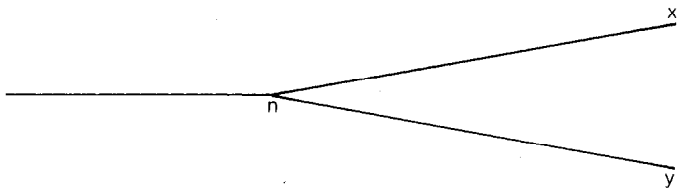

Fig. 11. Right closing.

precisely, if $x, y \in \Lambda_1$, $x \neq y$, and there exists n such that for all $i \leq n$ $x_i = y_i$, then $\pi(x) \neq \pi(y)$. Similarly, one has the notion of *left closing*.

The following facts are easy to verify.

Fact 1: A k-block map π: $\Lambda_1 \to \Lambda_2$ is right closing if and only if whenever $\pi(x) = \pi(y)$ and there exists an n such that $x_i = y_i$ for all $i \in [n - k + 1, n]$, then $x_i = y_i$ for all $i > n$ as well.

Fact 2: Any right closing map is finite-to-one.

Fact 3: Any right resolving map is right closing.

Fact 4: The composition of a right resolving map and a conjugacy is right closing.

While, strictly speaking, right closing is a (literally) more general notion than right resolving, in a certain sense it really is not. We need the following form of this statement.

Proposition 7 [21]: Let $\{\boldsymbol{A}\}$, $\{\boldsymbol{B}\}$, and $\{\boldsymbol{C}\}$ be SSFT's with $\{\boldsymbol{B}\} \subset \{\boldsymbol{A}\}$. Let π: $\{\boldsymbol{B}\} \to \{\boldsymbol{C}\}$ be a right closing factor map. Then there are SSFT's $\{\boldsymbol{B}'\} \subset \{\boldsymbol{A}'\}$ and a conjugacy ϕ: $\{\boldsymbol{A}'\} \to \{\boldsymbol{A}\}$ such that $\phi(\{\boldsymbol{B}'\}) = \{\boldsymbol{B}\}$ and the factor map $\pi \circ \phi$: $\{\boldsymbol{B}'\} \to \{\boldsymbol{C}\}$ is right resolving.

The point here is that we can think of factor maps π and $\pi \circ \phi$ as representing essentially the same map.

The following will be the starting point for the construction of codes where the rate is maximal (in Section IV).

Theorem 1: Let $\boldsymbol{S}$ be a sofic system with $h(\boldsymbol{S}) < \log(n)$. Then there is a right closing factor map from $\boldsymbol{S}$ into Σ_n.

Proof: It is well-known and easy to see that any subshift can be approximated by SSFT's from the *outside* in entropy, i.e., if $\boldsymbol{S}$ is a subshift and $\epsilon > 0$, then there is an SSFT $\{\boldsymbol{A}\}$ such that

$$\boldsymbol{S} \subset \{\boldsymbol{A}\}$$

and

$$h(\{\boldsymbol{A}\}) < h(\boldsymbol{S}) + \epsilon.$$

(Just look at the SSFT determined by blocks of large fixed length in $\boldsymbol{S}$.) Thus there is a SSFT $\{\boldsymbol{A}\}$ such that

$$\boldsymbol{S} \subset \{\boldsymbol{A}\}$$

and

$$h(\{\boldsymbol{A}\}) < \log(n).$$

By Proposition 5, $\{\boldsymbol{A}\}$ is conjugate to SSFT $\{\boldsymbol{B}\}$ and each row sum of $\boldsymbol{B}$ is at most n. Then one easily defines (by labeling edges) a 2-block right resolving factor map from $\{\boldsymbol{B}\}$ into Σ_n. Composing this with the conjugacy, one gets a right closing factor map from $\{\boldsymbol{A}\}$ into Σ_n (by Fact 4). Now one just restricts this factor map to $\boldsymbol{S}$.

IV. Entropy = $\log(n)$

Suppose that $\boldsymbol{S}$ is a sofic system with $h(\boldsymbol{S}) = (p/q)\log(n)$, where p, q, and n are integers. To code Σ_n into $\boldsymbol{S}$ at rate $(p/q)\log(n)$, we need to use all of $\boldsymbol{S}$, so we cannot use the approximation idea of Section II. So, instead of throwing out blocks, we must use blocks carefully. The idea, when $p/q = 1$, is that if $\boldsymbol{S}$ is presented as a factor of an SSFT $\{\boldsymbol{A}\}$, one finds a right resolving factor map $\{\boldsymbol{A}\} \to \Sigma_n$ such that any two points of $\{\boldsymbol{A}\}$ that present the same point of $\boldsymbol{S}$ are mapped to the same point of Σ_n. This defines a right closing map $\boldsymbol{S} \to \Sigma_n$ that can be used to construct codes. When $p/q \neq 1$, one applies the same sort of scheme replacing $\boldsymbol{S}$ by $(\boldsymbol{S}, \sigma^q)$ and Σ_n by Σ_{n^p}.

We first illustrate the rough idea with a very simple example. Let $\boldsymbol{S}$ be the sofic system shown in Fig. 12. $\boldsymbol{S}$ is a subset of Example 6. Also $h(\boldsymbol{S}) = \log(2)$.

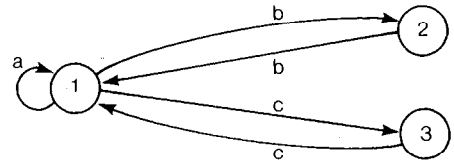

Fig. 12. Simple sofic system.

$\boldsymbol{S}$ is presented as the image of a 2-block factor map π from an SSFT $\{\boldsymbol{A}\}$. There are two points x and y in $\boldsymbol{S}$ that are bad in the sense that they have more than one π-inverse image (or, equivalently, they are each represented by more than one path on the graph). Namely,

$$x = \cdots bbb \cdots$$
$$y = \cdots ccc \cdots,$$

for if

$$u_1 = \cdots 1212 \cdots$$
$$u_2 = \cdots 2121 \cdots,$$

then

$$\pi(u_1) = \pi(u_2) = x,$$

and if

$$v_1 = \cdots 1313 \cdots$$
$$v_2 = \cdots 3131 \cdots,$$

then

$$\pi(v_1) = \pi(v_2) = y.$$

The points x and y are the only bad points, because once you see a, bc, or cb in a sequence, then you know where you are on the graph. A general procedure for finding the bad points is in [10]. Let $\boldsymbol{H} = \pi^{-1}$ (Bad set) $= \{u_1, u_2, v_1, v_2\}$.

Now, we want to construct an SSFT $\{\boldsymbol{B}\}$, which contains $\boldsymbol{H}$, and an (into) right resolving factor map: ϕ:$\{\boldsymbol{B}\} \to \Sigma_2$ such that $\phi(u_1) = \phi(u_2)$ and $\phi(v_1) = \phi(v_2)$. This is as shown in Fig. 13, ($\{\boldsymbol{B}\}$ is the SSFT generated by throwing out the loop at state 1; the zeros and ones in parentheses indicate the map ϕ.)

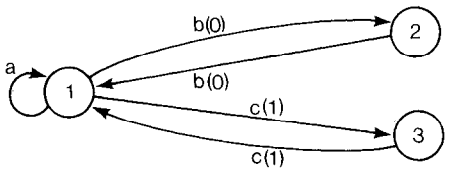

Fig. 13. Map on bad set.

Now we do state splitting on the graph (as in the proof of Proposition 5). The eigenvector is $(211)^{tr}$. So we split state 1 (see Fig. 14). Here we have partitioned the successors of state 1 into two groups $E_1 = \{1\}$ and $E_2 = \{2, 3\}$. (The idea is to partition the successors into groups such that, for each each group, the sum of the eigenvector components is divisible by n, in this case divisible by 2.)

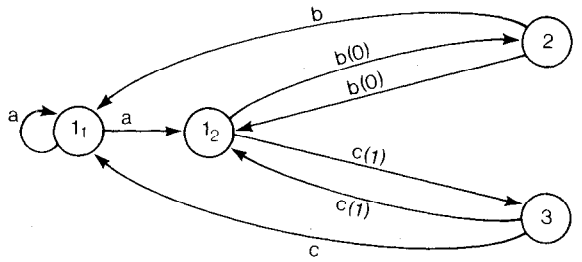

Fig. 14. State splitting.

Now, the SSFT $\{B\}$ is represented by the four edges connecting states 1_2, 2, and 3. On these four edges, the map ϕ is forced (since it was already defined). Now extend ϕ to all of $\{A\}$ by labeling the remaining edges (the arcs outside of $\{B\}$) with zeros and ones in a right resolving way. Any such extension will automatically define a factor map from $\{A\}$ onto Σ_2. It also defines a factor map from S onto Σ_2 since the only points of ambiguity were in the bad set (by definition) and these were already taken care of (see Fig. 15). This now defines a code from the free binary source into the sofic system S: one fixes an arbitrary state (say 3) and encodes 0-1 sequences by walking along the unique path defined by the sequence and then reading off

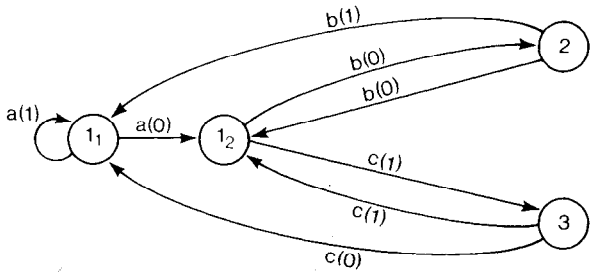

Fig. 15. Extension of factor map.

the corresponding a, b, c labels. For example, 0110101100 encodes to *caaaccaaab*. One decodes an a, b, c sequence by walking along *any* path corresponding to the sequence and then reading off the 0-1 labels. By construction, encoding has no look-ahead and decoding has limited look-ahead and no look-back.

Intuitively, what we did here was the following. We first made a (partial) right resolving 0-1 labeling (φ) on the original graph; this labeling was consistent with the original a-b-c labeling (π) and was defined only on the paths where consistency could possibly be a problem. Then, by state splitting, we represented the original sofic system in a new way, where the defining graph had two outgoing edges at each vertex; this new split graph naturally inherited an a-b-c labeling as well as 0-1 labeling (but again the 0-1 labeling was defined only on the paths where consistency could be a problem). Finally, the 0-1 labeling on the split graph was extended to a right resolving 0-1 labeling on the entire split graph.

The general sofic system presents many more difficulties. For instance, it is possible that in the newly created split graph, there is a state for which the 0-1 labels of both outgoing edges are forced to be identical. This would mean that the final 0-1 labeling on the split graph could not be right resolving. However, this will not happen if one can split states so that all of the bad outgoing edges (i.e., edges that are a-b-c labeled by symbols that are represented by more than one edge) can be put in one group of the state splitting partition, $\{E_1, E_2\}$. While this may not be possible for the original graph, it may (and in fact *will* for a large class of systems) work for some power of the original graph (the kth power of a labeled graph is the graph whose edges represent paths of length k in the original graph—this represents the kth power of the original system (see Example 5)).

Definition 4: A sofic system S is *almost of finite type* (AFT) if there is an irreducible SSFT $\{A\}$ and an onto factor map π: $\{A\} \to S$ that is 1-1 on an open set.

Remark: All sofic systems are 1-1 "almost-everywhere" images of SSFT's [10]. However, M. Boyle showed us an example of a transitive sofic system that is not AFT. A test for AFT is contained in [22].

The following gives a more concrete notion of the AFT idea.

Proposition 8: Let π: $\{A\} \to S$ be an onto factor map from an irreducible SSFT to a sofic system. The following are equivalent:

1) π is 1-1 on an open set;
2) π is 1-1 on an open dense set of full measure;
3) π is left closing, right closing, and has a resolving block.

Proof: 1) and 2) are equivalent by irreducibility. Given 2), then by the proof of [13, theorem 3.33], π has a resolving block; π must also be left and right closing since otherwise the non-1-1 set would be dense. Thus 2) implies 3). Given 3), one easily sees that since π is left and right closing, π must be 1-1 on the resolving block (an open set) (See Fact 1 and assume that π is a 1-block map.) Thus 1) holds.

All of the examples in this paper, as well as [5, Example 2] are AFT. The definition of AFT is motivated by the charge-constrained run-length limited sofic systems $\Lambda_{(d,k,c)}$.

Proposition 9: $\Lambda_{(d,k,c)}$ is AFT.

Proof: We first prove Proposition 9 for the charge-constrained systems Λ_c (without the run-length constraints). These sofic systems are presented by the labelings shown in Fig. 16. The factor maps represented by the labelings are right and left resolving, since at each vertex all outgoing edges are labeled distinctly and all incoming edges are labeled distinctly. Moreover, any block of 1's of sufficient length is a resolving block. Thus, by Proposition 8, the systems Λ_c are AFT.

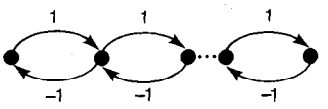

Fig. 16. General charge-constrained system.

For the general case, one can "jazz up" the preceding argument directly. Alternatively, one realizes that since the run-length constraints are SSFT, each $\Lambda_{(d,k,c)}$ is the intersection of an SSFT with Λ_c. But we just proved above that Λ_c is AFT. So Proposition 9 will follow from the following lemma.

Lemma 2: The intersection of an SSFT with an AFT sofic system is again an AFT sofic system.

Proof: Let $\{\boldsymbol{B}\}$ be the SSFT, and let $\boldsymbol{S}$ be the AFT sofic system with π: $\{\boldsymbol{A}\} \to \boldsymbol{S}$ 1-1 on an open set. The reader can easily verify that $\pi^{-1}(\boldsymbol{S} \cap \{\boldsymbol{B}\})$ is an SSFT. Moreover, the restriction of π to this SSFT,

$$\pi\colon \pi^{-1}(\boldsymbol{S} \cap \{\boldsymbol{B}\}) \to \boldsymbol{S} \cap \{\boldsymbol{B}\}$$

inherits the right and left closing properties that π has (Proposition 8). While this restriction does not necessarily inherit a resolving block from π, the construction in [10, 3.4] will present $\boldsymbol{S} \cap \{\boldsymbol{B}\}$ as the image of an SSFT by a map that is right closing, left closing, and has a resolving block.

Remark: D. Lind [24] in fact showed us that the intersection of two AFT sofic systems is again an AFT sofic system.

It can happen that $h(\Lambda_{(d,k,c)})$ is the log of a rational root of a positive integer, e.g., $h(\Lambda_{(1,3,3)}) = \log(\sqrt{2}) = 1/2$ (see [11]). Thus, the highest possible rate for a code of the free binary source into $\Lambda_{(1,3,3)}$ is $1/2$. The following theorem shows that, in principle, one can find such a code with a weaker version of the stationary property 4) (see Introduction); namely, we produce a code that is invariant by shifting by l in the domain and $2l$ in the range for some l. (Patel [11] found a nice simple stationary code but with rate slightly lower than $1/2$.)

The purpose of this section is to prove the following Theorem.

Theorem 2: Let $\boldsymbol{S}$ be an AFT sofic system with $h(\boldsymbol{S}) = \log(n)$. Then there exists a positive integer l and a right closing (thus finite-to-one) factor map from $(\boldsymbol{S}, \sigma^l)$ onto (Σ_n, σ^l).

Remarks:

1) We do not know if the theorem can be strengthened (i.e., can the AFT condition be dropped? Can l be reduced to 1?) In our proof, l depends on the entropy of the bad set (i.e., the set of points with more than 1 inverse image via a map π: $\{\boldsymbol{A}\} \to \boldsymbol{S}$ that is 1-1 on an open set). Also, if the bad set is finite, then l can be made to be 1.

2) If $\boldsymbol{S}$ is mixing, then the factor map of Theorem 2 can be chosen to have a resolving block and therefore 1-1 "almost everywhere." However, we remark without proof that it cannot, in general, be chosen to be 1-1 on an open set.

Proof of Theorem 2: Let π: $\{\boldsymbol{A}\} \to \boldsymbol{S}$ be an onto factor map that is 1-1 on an open set. Let $\boldsymbol{H} = \{x \in \{\boldsymbol{A}\}: \#\pi^{-1}(\pi x) > 1\}$. Then $\overline{\boldsymbol{H}} \subsetneq \{\boldsymbol{A}\}$ (in fact, $\boldsymbol{H}$ is closed, but this is irrelevant to the proof). Thus $\boldsymbol{H}$ is a proper subshift of $\{\boldsymbol{A}\}$. Since any subshift is an intersection of the SSFT's that contain it, there must be an SSFT $\{\boldsymbol{B}\}$ such that

$$\overline{\boldsymbol{H}} \subset \{\boldsymbol{B}\} \subsetneq \{\boldsymbol{A}\}.$$

Now, since $\{\boldsymbol{B}\}$ is proper, we have $h(\{\boldsymbol{B}\}) < h(\{\boldsymbol{A}\})$ [23, Theorem 3.3]. This, together with the facts that π is finite-to-one (Proposition 8 and Fact 2) and finite-to-one maps preserve entropy, (Proposition 1) yields

$$\begin{aligned} h(\pi(\{\boldsymbol{B}\})) &= h(\{\boldsymbol{B}\}) < h(\{\boldsymbol{A}\}) \\ &= h(\pi(\{\boldsymbol{A}\})) = h(\boldsymbol{S}). \end{aligned}$$

Thus, by Theorem 1 there is a right closing factor map ψ: $\pi(\{\boldsymbol{B}\}) \to \Sigma_n$. Since π is right closing (Proposition 8) it follows that $\phi \equiv \psi \circ \pi|_{\{\boldsymbol{B}\}}$ is a right closing factor map. By Proposition 7 we may assume (by possibly conjugating $\{\boldsymbol{B}\}$ and $\{\boldsymbol{A}\}$ to another form) that ϕ is right resolving. Now, since π is 1-1 off of $\{\boldsymbol{B}\}$, any factor map that is an extension of ϕ to all of $\{\boldsymbol{A}\}$ will automatically yield a well-defined factor map from $\boldsymbol{S}$ into Σ_n. If, moreover, the extension is right closing, it will be finite-to-one (Fact 2) whence the image of $\boldsymbol{S}$ will have full entropy ($\log(n)$) in Σ_n. But, then again, by [23, Theorem 3.3] this means that the map is onto. So it suffices to prove the following.

Theorem 3 (Extension Theorem): Let $\{\boldsymbol{B}\} \subset \{\boldsymbol{A}\}$ be two SSFT's with $h(\{\boldsymbol{A}\}) = \log(n)$. Let ϕ: $\{\boldsymbol{B}\} \to \Sigma_n$ be a right resolving factor map. Then there exists an integer l such that ϕ can be extended to a right resolving factor map from $(\{\boldsymbol{A}\}, \sigma^l)$ onto (Σ_n, σ^l).

Proof: Since $\{\boldsymbol{B}\}$ is a proper subshift, we may assume by going to a higher block system (Example 4) that the state set of $\{\boldsymbol{B}\}$ is a proper subset of the state set of $\{\boldsymbol{A}\}$. Now, if $\boldsymbol{A}$ had row sum n, then it would be easy to extend as a right resolving factor map. Of course, $\{\boldsymbol{A}\}$ is conjugate to an SSFT defined by a matrix with row sum n, but this conjugacy would in general represent $\{\boldsymbol{B}\}$ in a form that makes ϕ right closing, not right resolving. The idea is (as in Section III) to split states and reduce the components of an eigenvector while keeping the map ϕ true to its original definition.

We need the following two propositions.

Proposition 10: Let A^* be an irreducible 0-1 matrix with states A^* and $h(\{A^*\}) = \log(p)$, $p \in Z^+$. Let B^* be a 0-1 matrix with $B^* \le A^*$ (i.e., entry by entry) and $B^*_{ij} < A^*_{ij}$ for some i, j. Let x be a positive integral eigenvector for A^* and assume that not all of the components of x are the same. Then, for any $v \in A^*$ with x_v maximal, there exists l such that

$$p^l < \sum_{\alpha \in A^*} \left((A^*)^l_{v\alpha} - (B^*)^l_{v\alpha} \right).$$

Proof: Without loss of generality, we may assume that A^* is aperiodic (otherwise replace A^* by an appropriate power) [13, Theorem 3.6]. Thus, there exists a constant $C > 0$ such that for sufficiently large l and for all $i, j \in A^*$

$$Cp^l < (A^*)^l_{ij} \tag{4.1}$$

Also, $h(\{B\}) < h(\{A\}) = \log p$, and so for all $\epsilon > 0$ there exists k_0 such that for all $l \ge k_0$ and for all $i, j \in B$

$$(B^*)^l_{ij} < \epsilon p^l. \tag{4.2}$$

Now fix $v \in A^*$ with x_v maximal. Since x is an eigenvector, we have

$$x_v p^l = \sum_{\alpha \in A^*} (A^*)^l_{v\alpha} \cdot x_\alpha \le \left[\sum_{\substack{\alpha \in A^* \\ x_\alpha < x_v}} (A^*)^l_{v\alpha} \right](x_v - 1) + \left[\sum_{\substack{\alpha \in A^* \\ x_\alpha = x_v}} (A^*)^l_{v\alpha} \right] x_v.$$

Thus, dividing by x_v,

$$p^l + \frac{1}{x_v} \left[\sum_{\substack{\alpha \in A^* \\ x_\alpha < x_v}} (A^*)^l_{v\alpha} \right] \le \sum_{\alpha \in A^*} (A^*)^l_{v\alpha}.$$

This together with (4.1) and the assumption that not all the components of x are the same shows

$$p^l + \frac{C}{x_v} p^l \le \sum_{\alpha \in A^*} (A^*)^l_{v\alpha}. \tag{4.3}$$

Now, apply (4.2) with $\epsilon = C/(x_v(\#A^*))$ to get

$$p^l + \sum_{\alpha \in A^*} (B^*)^l_{v\alpha} < p^l + \frac{C}{x_v} p^l.$$

This, together with (4.3) yields the following proposition.

Proposition 11: Let $\bar{B} \le \bar{A}$ be 0-1 matrices. Let $\bar{B}$ and $\bar{A}$ be the states of $\{\bar{B}\}$ and $\{\bar{A}\}$ (so naturally $\bar{B} \subset \bar{A}$). Assume that A is irreducible, has entropy $\log(m)$ ($m \in Z^+$), and assume that x is a positive integral eigenvector for $\bar{A}$ with not all of its components the same. Let M be the maximal component of x. If

1) $\bar{\phi}$: $\{\bar{B}\} \to \Sigma_m$ is a right resolving factor map, and
2) for all $\bar{a} \in \bar{B}$ with $x_{\bar{a}} = M$, $m < \#(F_{\bar{A}}(\bar{a}) - F_{\bar{B}}(\bar{a}))$,

then there exist 0-1 matrices $B' \le A'$ with states $B' \subset A'$, a positive integral eigenvector y for A', with maximal component not exceeding M, and a conjugacy (f) from $\{A'\}$ to $\{\bar{A}\}$, which carries $\{B'\}$ to $\{B\}$ such that

1) $\bar{\phi} \circ f$: $\{B'\} \to \Sigma_m$ is a right-resolving factor map;
2) for all $a' \in B'$ with $y_{a'} = M$, $m < \#(F_{A'}(a') - F_{B'}(a'))$,
3) $\#\{a' \in A'$: $y_{a'} = M\} < \#\{\bar{a} \in \bar{A}$: $x_{\bar{a}} = M\}$.

Proof: Let $\bar{v} \in \bar{A}$ with $x_{\bar{v}} = M$. We want to split $\bar{v}$, as in Section III. By Lemma 1 and Proposition 11, there exists a set

$$E \subsetneq F_{\bar{A}}(\bar{v}) - F_{\bar{B}}(\bar{v})$$

with $\sum_{\alpha \in E} x_\alpha$ a multiple of m. Do state splitting with

$$E_1 \equiv E$$
$$E_2 \equiv F_{\bar{A}}(\bar{v}) - E.$$

Let $A' = (\bar{A} - \{\bar{v}\}) \cup \{v_1, v_2\}$, and let A' be the transition matrix defined by the description of state splitting in Section III. Also

$$B' = \begin{cases} \bar{B}, & \text{if } \bar{v} \notin \bar{B} \\ (\bar{B} - \{\bar{v}\}) \cup \{v_2\}, & \text{if } \bar{v} \in \bar{B} \end{cases}$$

and B' is defined by transitions among B'. Also f is defined (see Proposition 6) as the factor map generated by f^*:

$$f^*|_{A' - \{v_1, v_2\}} = \text{identity}$$
$$f^*(v_1) = f^*(v_2) = \bar{v}.$$

To prove 1) above, let $a'\alpha', a'\alpha''$ be 2-blocks of $\{B'\}$ with

$$\bar{\phi} \circ f(a'\alpha') = \bar{\phi} \circ f(a'\alpha'').$$

Since $\bar{\phi}$ is right resolving, it follows that $f(\alpha') = f(\alpha'')$. But $f|_{B'}$ is 1-1 and so $\alpha' = \alpha''$.

Observe that the vector

$$y_{a'} = \begin{cases} x_{a'}, & \text{if } a' \ne v_1, v_2 \\ \dfrac{1}{m} \displaystyle\sum_{\alpha' \in E} x_{\alpha'}, & \text{if } a' = v_1 \\ \dfrac{1}{m} \displaystyle\sum_{a' \in F_{\bar{A}}(\bar{v}) - E} x_{\alpha'}, & \text{if } a' = v_2 \end{cases}$$

is an eigenvector for A'. Thus, since x is an eigenvector, $y_{v_1}, y_{v_2} < M$ and so 3) holds. To see 2), observe that for all $a' \in B'$ with $a' \ne v_2$,

$$\#\left(F_{\bar{A}}(a') - F_{\bar{B}}(a')\right) \le \#\left(F_{A'}(a') - F_{B'}(a')\right),$$

and if $a' = v_2$, then $y_{v_2} < M$.

Proof of Theorem 3: Let x be a positive integral eigenvector for A (with eigenvalue n). If all the components of x were the same, then A would have row sum n and the map ϕ would be easy to extend. Otherwise, apply Proposition 10 to $A^* \equiv A$, $B^* \equiv B$, and $p \equiv n$. This yields an integer l_1. Now apply Proposition 11 to $\bar{A} \equiv$ the matrix of σ^{l_1} relative to $\bar{A} \equiv$ the $\{A\}$-allowable l_1-blocks, $\bar{B} \equiv$ the matrix of σ^{l_1} relative to $\bar{B} \equiv$ the $\{B\}$-allowable l_1-blocks and $m = n^{l_1}$. (Here, we are identifying (Σ_m, σ) with (Σ_n, σ^{l_1}) as in

Example 5 and we choose an eigenvector $x_{a_1 \cdots a_k} = x_{a_k}$.) Now we apply Proposition 11 iteratively until we arrive at a matrix of A' with a positive integral eigenvector whose largest component is less than M. Now apply Proposition 10 to $A^* \equiv A'$, $B^* \equiv B'$ and $p \equiv m = n^{l_1}$. This produces a new integer l_2, and then one applies Proposition 11 again to the matrix of σ^{l_2} relative to the $\{A^*\}$-allowable l_2-blocks, etc. Repeating this application of Propositions 10 and 11 we eventually obtain matrices A', B' with states $B' \subset A'$ and an integer l such that

1) $(\{A'\}, \sigma)$ is conjugate to $(\{A\}, \sigma^l)$ via a conjugacy (f) that carries $(\{B'\}, \sigma)$ to $(\{B\}), \sigma^l)$,
2) $\phi \circ f$: $(\{B'\}, \sigma) \to (\Sigma_n, \sigma^l)$ is right resolving, and
3) A' has row sum n^l.

One easily extends $\phi \circ f$ to a right resolving factor map ϕ' from $(\{A'\}, \sigma)$ onto (Σ_n, σ^l). Then, $\phi' \circ f^{-1}$ is the desired factor map from $(\{A\}, \sigma^l)$ onto (Σ_n, σ^l).

ACKNOWLEDGMENT

We are indebted to many people for useful discussions: E. Coven, M. Hassner, N. Hunau, B. Kitchens, M. Paul, K. Petersen, P. Siegel, and especially R. Adler.

REFERENCES

[1] R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 5–22, 1983.
[2] P. A. Franaszek, "A general method for channel coding," *IBM J. Res. Dev.*, vol. 24, pp. 638–691, 1980.
[3] ——, "Construction of bounded delay codes for discrete channels," *IBM J. Res. Dev.*, vol. 26, pp. 506–514, 1982.
[4] ——, "On future-dependent block coding for input-restricted channels," *IBM J. Res. Dev.*, vol. 23, pp. 75–81, 1979.
[5] A. Lempel and M. Cohn, "Look ahead coding for input restricted channels," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 933–937, 1982.
[6] C. Shannon, "Mathematical theory of communication," *Bell Syst. Tech. J.* vol. 27, pp. 379–423, 623–656, 1948.
[7] M. Hassner, "A nonprobablistic source and channel coding theory," Ph.D. thesis, Univ. California, Los Angeles, 1980
[8] B. Weiss, "Subshifts of finite type and sofic systems," *Monats. Math.*, vol. 77, pp. 462–474, 1973.
[9] R. M. Gray, "Sliding block source coding," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 357–368, 1975.
[10] E. Coven and M. Paul, "Finite procedures for sofic systems," *Monats. Math.*, vol. 83, pp. 265–278, 1977.
[11] A. M. Patel,"Zero modulation in magnetic recording," *IBM J. Res. Dev.*, vol. 19, no. 4, pp. 366–378, 1975.
[12] K. Norris and D. S. Bloomberg, "Channel capacity of charge constrained run-length limited codes," *IEEE Trans. Magn.*, vol. MAG17, pp. 3452–3455, 1981.
[13] R. Adler and B. Marcus, "Topological entropy and equivalence of dynamical systems," *Mem. AMS*, vol. 219, 1979.
[14] R. Fischer, "Graphs and symbolic dynamics," *Colloq. Math. Soc. Janos Bolyai. Topics in Inform. Theory*, 1975.
[15] E. Coven and M. Paul, "Sofic systems," *Israel J. Math.*, vol. 20, pp. 165–177, 1975.
[16] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1962, Ch. 2.
[17] M. Boyle, "Factors of sofic systems," *Trans. AMS*, to appear.
[18] B. Marcus, "Factors and extensions of full shifts," *Monats. Math.*, vol. 88, pp. 239–247, 1979.
[19] R. F. Williams, "Classification of shifts of finite type," *Ann. Math.*, vol. 98, pp. 120–153, 1973; Errata, *Ann. Math.*, vol. 99, pp. 380–381, 1974.
[20] W. Parry and R. Williams, "Block coding and a Zeta function for finite Markov chains," *Bull. London Math. Soc.*, vol. 35, pp. 483–495, 1977.
[21] B. Kitchens, "Continuity properties of factor maps in ergodic theory," Ph.D. Thesis, Univ. North Carolina, Chapel Hill, 1981.
[22] M. Boyle, B. Kitchens, and B. Marcus, "A note on minimal covers for sofic systems," *Proc. AMS*, to appear.
[23] E. Coven and M. Paul, "Endomorphisms of irreducible SSFT," *Math. Syst. Theory*, vol. 8, pp. 167–175, 1974.
[24] D. Lind, personal communication.