Encoding Semiconstrained Systems

Ohad Elishco
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
ohadeli@bgu.ac.il

Tom Meyerovitch
Department of Mathematics
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
mtom@math.bgu.ac.il

Moshe Schwartz
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
schwartz@ee.bgu.ac.il

***Abstract*—Semiconstrained systems were recently suggested as a generalization of constrained systems, commonly used in communication and data-storage applications that require certain offending subsequences be avoided. In an attempt to apply techniques from constrained systems, we study sequences of constrained systems that are contained in, or contain, a given semiconstrained system, while approaching its capacity. In the case of contained systems we describe to such sequences resulting in constant-to-constant bit-rate block encoders and sliding-block encoders. Surprisingly, in the case of containing systems we show that a "generic" semiconstrained system is never contained in a proper fully-constrained system.**

I. Introduction

Many communication and data-storage systems employ constrained coding. In such a scheme, information is encoded in sequences that avoid the occurrence of certain subsequences. Perhaps the most common example is that of (d,k)-RLL which comprises of binary sequences that avoid subsequences of $k+1$ 1's, or two 1's that are separated by less than d 0's. For various other examples the reader is referred to [5] and the many references therein.

The reason for avoiding such subsequences is mainly due to the fact that their appearance contributes to noise in the system. However, by altogether forbidding their occurrence, the possible rate at which information may be transmitted is severely reduced. By relaxing the constraints and allowing some appearances of the offending subsequences, the rate penalty may be reduced. So rather than imposing combinatorial constraints on all substrings of the output, we impose statistical constraints on substrings that are sampled from the output at a uniform random offset. Such an approach was studied, for example, in the case of channels with cost constraints [6], [8].

A general approach was suggested in [2], [4], in which a *semiconstrained system (SCS)* was defined by a list of offending subsequences, and an upper bound (called a *semiconstraint*) on the frequency of each subsequence appearing. Note that fully-constrained coding is a special case of semiconstrained coding.

A careful choice of semiconstraints also allows the study of systems that, up to now, were studied in an ad-hoc manner only. As examples we mention DC-free RLL coding [10], constant-weight ICI coding for flash memories [7], [12], and coding to mitigate the appearance of ghost pulses in optical communication [13], [14].

One of the most important questions, given a SCS, is how to encode any unconstrained input sequence into a sequence that satisfies all the given semiconstraints. The various encoding schemes suggested in [7], [10], [12]–[14] are ad-hoc and do not apply to general SCS. The encoding scheme for channels with cost constraints given in [8] (which overlap somewhat with SCS) is indeed general, however it is not capacity achieving. Later, within the scope of channels with cost constraints, and motivated by partial-response channels, [9], [16] briefly report on capacity-achieving schemes.

Under the assumption that the input stream consists of i.i.d. uniformly-random bits, a general capacity-achieving encoding scheme for SCS was described in [2], [4]. The scheme involved a maxentropic Markov chain over a modified De-Bruijn graph. Input symbols were converted via an arithmetic decoder to a biased stream of symbols which were used to generate a path in the graph, which in turn generated symbols to be transmitted. A reverse operation was employed at the receiving side. Additionally to the assumption on the distribution of the input, to enforce a constant-to-constant bit rate, the encoder has a probability of failure (albeit, asymptotically vanishing). Thus, not all input streams may be converted to semiconstrained sequences.

Compared with SCS, for "conventional" fully-constrained systems there is a general method for constructing encoders working arbitrarily close to capacity: The celebrated state-splitting algorithm. However, as we explain in the following sections, this method fails even on very simple SCS, due to the fact that in most cases they do not form regular languages.

In this work we consider the problem of encoding an arbitrary input string into a sequence that satisfies all the given semiconstraints. We do not make statistical or combinatorial assumptions on the input, only that it is sufficiently long. Specifically, we show the following: For every given SCS that satisfies certain mild assumptions and every $\epsilon > 0$ we present a fully-constrained system that is "eventually-contained" in the given SCS, with capacity decrease of at most ϵ. This allows us to construct either block encoders or sliding-block encoders, trading encoder anticipation for number of states. In the other direction, we show that no proper constrained system with forbidden words can contain a given SCS (under certain mild assumptions). We also observe that any encoding scheme for a SCS that works for arbitrary input and has both finite memory and finite anticipation must produce sequences that satisfy some fully constrained system.

The paper is organized as follows. In Section II we present the definition and notation used throughout the paper. In Section III we study sequences of constrained systems that

978-1-5090-1805-5/16/$31.00 ©2016 IEEE

are contained in a given SCS and approach its capacity from below. In Section IV we do the reverse, and study constrained systems containing a given SCS. We present conclusions and other results in Section V. Due to page limitation proofs are sketched or omitted. For a more detailed version with full proofs, the reader is referred to [3].

II. Preliminaries

A. Semiconstrained Systems

Let Σ be a finite alphabet and let Σ^* denote the set of all the finite sequences over Σ. The elements of Σ^* are called *words* (or *strings*). The *length* of a word $\omega \in \Sigma^*$ is denoted by $|\omega|$. Given two words, $\omega, \omega' \in \Sigma^*$, their concatenation is denoted by $\omega\omega'$. Repeated concatenation is denoted using a superscript, i.e., for any natural $m \in \mathbb{N}$, ω^m denotes $\omega^m = \omega\omega\dots\omega$, where m copies of ω are concatenated. By convention, $\omega^0 = \varepsilon$, where ε the unique empty word of length 0. By extension, if $S \subseteq \Sigma^*$ is a set of words, then S^m denotes the set

$$S^m = \{\omega_1\omega_2\dots\omega_m : \forall i, \omega_i \in S\},$$

with $S^0 = \{\varepsilon\}$, $S^* = \bigcup_{i\geqslant 0} S^i$, and $S^+ = \bigcup_{i\geqslant 1} S^i$.

The set of k-length subwords of ω is defined by

$$\mathrm{sub}_k(\omega) = \left\{\beta \in \Sigma^k : \omega = \alpha\beta\gamma \text{ for some } \alpha, \gamma \in \Sigma^*\right\}.$$

For $\omega \in \Sigma^*$ and $k \leqslant |\omega|$, $\mathrm{fr}^k_\omega \in \mathcal{P}(\Sigma^k)$ is defined as the uniform measure on $\mathrm{sub}_k(\omega)$ (taken in the multiset sense), where $\mathcal{P}(\Sigma^k)$ denotes the set of all probability measures on Σ^k. We can naturally identify

$$\mathcal{P}(\Sigma^k) = \left\{\eta \in [0,1]^{\Sigma^k} : \sum_{\phi\in\Sigma^k} \eta(\phi) = 1\right\}.$$

It follows that for all $\beta \in \Sigma^k$,

$$\mathrm{fr}^k_\omega(\beta) = \frac{1}{|\omega| - k + 1} \left|\{(\alpha,\gamma) : \alpha,\gamma \in \Sigma^*, \alpha\beta\gamma = \omega\}\right|$$

Definition 1. *Let $\mathcal{F} \subseteq \Sigma^*$ be a finite set of words, and let $P \in [0,1]^{\mathcal{F}}$ be a function from $\mathcal{F}$ to the real interval $[0,1]$. A* semiconstrained system (SCS) *is the pair $(\mathcal{F}, P)$. The set of* admissible words *for $(\mathcal{F}, P)$ is defined by,*

$$\mathcal{B}(\mathcal{F},P) = \left\{\omega \in \Sigma^* : \forall\phi \in \mathcal{F}, \mathrm{fr}^{|\phi|}_\omega(\phi) \leqslant P(\phi)\right\}.$$

For convenience we also define the set of admissible words of length exactly n as

$$\mathcal{B}_n(\mathcal{F},P) = \mathcal{B}(\mathcal{F},P) \cap \Sigma^n.$$

An important figure of merit associate with any set of words $S \subseteq \Sigma^*$ is its capacity.

Definition 2. *Let Σ by a finite alphabet and $S \subseteq \Sigma^*$. The* capacity *of S, denoted $\mathsf{cap}(S)$, is defined as*

$$\mathsf{cap}(S) = \limsup_{n\to\infty} \frac{1}{n} \log_2 |S \cap \Sigma^n|.$$

Thus, in the case of a SCS $(\mathcal{F}, P)$, the capacity $\mathsf{cap}(\mathcal{B}(\mathcal{F},P))$ intuitively measures the exponential growth rate of the number of words that satisfy the constraints $\mathcal{F}$ and P as a function of the word length.

A relaxation of semiconstrained systems was also suggested in [2], [4].

Definition 3. *Let $\mathcal{F} \subseteq \Sigma^*$ be a finite set of words, and let $P \in [0,1]^{\mathcal{F}}$. The set of* weakly-admissible words *for $(\mathcal{F}, P)$ is defined by*

$$\overline{\mathcal{B}}(\mathcal{F},P) = \left\{\omega \in \Sigma^* : \forall\phi \in \mathcal{F}, \mathrm{fr}^{|\phi|}_\omega(\phi) \leqslant P(\phi) + \xi(|\omega|)\right\},$$

where $\xi : \mathbb{N} \to \mathbb{R}^+$ is a function satisfying both $\xi(n) = o(1)$ and $\xi(n) = \Omega(1/n)$. Also $\overline{\mathcal{B}}_n(\mathcal{F},P) = \overline{\mathcal{B}}(\mathcal{F},P) \cap \Sigma^n$.

We note that $\overline{\mathcal{B}}(\mathcal{F},P)$ was called a *weak semiconstrained system (WSCS)* in [2], [4], though we shall prefer to use the term weakly-admissible words for $(\mathcal{F},P)$. It was also shown there that it suffices to consider only sets $\mathcal{F} \in \Sigma^k$, i.e., all the offending patterns are of the same length k. We shall follow suit, and assume from now on, without loss of generality, that $\mathcal{F} \subseteq \Sigma^k$.

A SCS $(\mathcal{F},P)$ can naturally be identified with a subset of $\mathcal{P}(\Sigma^k)$,

$$\Gamma(\mathcal{F},P) = \left\{\eta \in \mathcal{P}(\Sigma^k) : \forall\phi \in \Sigma^k, \eta(\phi) \leqslant P(\phi)\right\}.$$

If $\mathcal{F}$ and P are understood from the context we shall simply write Γ. The admissible words for the SCS $(\mathcal{F},P)$ are therefore

$$\mathcal{B}(\mathcal{F},P) = \left\{\omega \in \Sigma^* : \mathrm{fr}^k_\omega \in \Gamma(\mathcal{F},P)\right\}.$$

A particular set of probability measures of interest to us is the set of *shift-invariant probability measures*. We say $\eta \in \mathcal{P}(\Sigma^k)$ is *shift-invariant* if for all $\phi \in \Sigma^{k-1}$, $\sum_{a\in\Sigma} \eta(a\phi) = \sum_{a\in\Sigma} \eta(\phi a)$. We denote the set of shift-invariant probability measures by $\mathcal{P}_{\mathrm{si}}(\Sigma^k)$, which is a closed subset of $\mathcal{P}(\Sigma^k)$. These are precisely the probability measures that arise as marginals of shift-invariant measures on $\Sigma^{\mathbb{N}}$ or $\Sigma^{\mathbb{Z}}$. For a discussion see [1]. In particular, we have the following lemma.

Lemma 4. *Fix a finite alphabet Σ, and $k \geqslant 2$. If $\Gamma \subseteq \mathcal{P}(\Sigma^k) \setminus \mathcal{P}_{\mathrm{si}}(\Sigma^k)$ is closed, and $\mathcal{B}(\Gamma) = \{\omega \in \Sigma^* : \mathrm{fr}^k_\omega \in \Gamma\}$, then $\mathsf{cap}(\mathcal{B}(\Gamma)) = -\infty$, i.e., $\mathcal{B}(\Gamma)$ is a finite set.*

Lemma 4 motivates us to study probability measures that are shift invariant. For a set $\Gamma \subset \mathcal{P}_{\mathrm{si}}(\Sigma^k)$ we denote by $\mathrm{int}(\Gamma)$ the interior of Γ, and by $\mathrm{cl}(\Gamma)$ the closure of Γ, both relatively to $\mathcal{P}_{\mathrm{si}}(\Sigma^k)$. We say $\Gamma \subset \mathcal{P}(\Sigma^k)$ is *fat* if $\mathrm{cl}(\mathrm{int}(\Gamma \cap \mathcal{P}_{\mathrm{si}}(\Sigma^k))) = \mathrm{cl}(\Gamma \cap \mathcal{P}_{\mathrm{si}}(\Sigma^k))$.

We recall the following result [2], [4]:

Theorem 5. *Let $\mathcal{F} \subseteq \Sigma^k$ and $P \in [0,1]^{\mathcal{F}}$. If $\Gamma(\mathcal{F},P)$ is fat,*

$$\mathsf{cap}(\mathcal{B}(\mathcal{F},P)) = \mathsf{cap}(\overline{\mathcal{B}}(\mathcal{F},P)) = 1 - \inf_{\eta\in\Gamma(\mathcal{F},P)} H(\eta|\eta'),$$

where $H(\cdot|\cdot)$ is the relative entropy function, and $\eta'(\phi a) = \sum_{a'\in\Sigma} \eta(\phi a') / |\Sigma|$, for all $\phi \in \Sigma^{k-1}$ and $a \in \Sigma$. Additionally, $\mathsf{cap}(\mathcal{B}(\mathcal{F},P))$ and $\mathsf{cap}(\overline{\mathcal{B}}(\mathcal{F},P))$ are continuous and convex in P, and the limits in their definitions exist.

B. Fully-Constrained Systems

As noted in the introduction, "conventional" constrained systems are a special case of semiconstrained systems. They can be viewed as SCS of the form $(\mathcal{F}, P)$ where $P(\phi) = 0$ for all $\phi \in \mathcal{F}$. We will refer to those as *fully-constrained systems*.

Let $G = (V, E)$ be a finite directed graph, where we allow parallel edges. A labeling function $\mathcal{L} : E \to \Sigma^q$ assigns a length-q label over the alphabet to each edge. By simple extension, the label of a directed (non-empty) path in the graph $\gamma = e_1 \to e_2 \to \cdots \to e_n$ is defined as $\mathcal{L}(\gamma) = \mathcal{L}(e_1)\mathcal{L}(e_2)\ldots\mathcal{L}(e_n)$. Finally, we define the language presented by the graph G, denoted $\mathcal{L}(G)$, to be the labels of all finite directed paths in G.

Constrained systems have been widely studied [5], [11]. In particular, it is well known that $\mathcal{B}(\mathcal{F}, 0) = \mathcal{L}(G)$ for some finite directed labeled graph G in the manner described above. An immediate consequence is the fact that $\mathcal{B}(\mathcal{F}, 0)$ is a regular language in the Chomsky hierarchy of formal languages [15]. We do note, however, that not all regular languages (which correspond to languages of sofic subshifts) are constrained systems (which are defined by a finite number of forbidden words, and correspond to subshifts of finite type).

A wide variety of tools exist for manipulating constrained systems, including the state-splitting algorithm (see [11]). In essence, under mild assumptions, given a constrained system $\mathcal{B}(\mathcal{F}, 0) = \mathcal{L}(G)$, and two positive integers p and q that satisfy $p/q < \mathsf{cap}(\mathcal{B}(\mathcal{F}, 0))$, we can find another constrained system $\mathcal{B}(\mathcal{F}', 0) = \mathcal{L}(G')$ with the following properties:

- $\mathcal{L}(G') \subseteq \mathcal{L}(G)$.
- $\mathsf{cap}(\mathcal{B}(\mathcal{F}', 0)) = p/q$, also called the *rate* of the encoder.
- G' is a $p : q$ encoder for $\mathcal{L}(G)$ with finite anticipation $a \in \mathbb{N} \cup \{0\}$, i.e., the out-degree of each vertex is 2^p, the edges labels in G' are from Σ^q, and paths of length $a + 1$ that start from the same vertex and generate the same word agree on the first edge.

Unfortunately, even for very simple semiconstraints, $\mathcal{B}(\mathcal{F}, P)$ is not a regular language in general. As an example, for $\Sigma = \{0, 1\}$, $\mathcal{F} = \{1\}$, and $P(1) = p$, it is easily seen that for any rational $0 < p < 1$, the semiconstrained system $\mathcal{B}(\mathcal{F}, P)$ is a non-regular context-free language, whereas for any irrational p the system is not even context free [15, §4.9, Exercise 25]. Thus, the wonderful machinery of the state-splitting algorithm cannot be applied directly for general SCS.

Another important property of languages associated to fully-constrained systems is that these languages are *factorial*. This means that a subsword of an admissible word is also an admissible word. Factoriality implies for instance that the sequence $\frac{1}{n} \log |\mathcal{B}(\mathcal{F}, 0)|$ is subadditive, so the lim sup in the definition of the capacity is actually a limit by Fekete's lemma. The factoriality property is not shared by SCS in general.

III. Approaching Capacity from Below

In this section we study the problem of finding a sequence of fully-constrained systems that are contained in a given semiconstrained (or weakly semiconstrained) system, with the additional requirement that the capacity of the former approaches that of latter in the limit. We present two such sequences which induce (perhaps after state splitting) two possible encoders for the SCS or WSCS.

It will be easier for us to describe fully-constrained systems that are only *eventually* contained in the desired SCS. Formally, given two infinite subsets, $S_1, S_2 \in \Sigma^*$, we say S_1 is eventually contained in S_2, denoted $S_1 \subseteq^e S_2$, if $|S_1 \setminus S_2| < \infty$. A fully-constrained system that is eventually contained in a given SCS may easily be transformed into another fully-constrained system that is contained (in the usual sense) in the given SCS by removing the words that are inadmissible in the SCS.

A. Block Encoders for SCS

The first sequence of fully-constrained systems we construct are each presented by a graph with a single state. Such graphs are called block encoders.

Let $(\mathcal{F}, P)$ be a SCS with a fat $\Gamma(\mathcal{F}, P)$. The condition on $\Gamma(\mathcal{F}, P)$ guarantees that it can be slightly shrunk while remaining not empty. More formally, for any $\epsilon \in [0, 1)^{\mathcal{F}}$ we define the function $P - \epsilon$ by $(P - \epsilon)(\phi) = P(\phi) - \epsilon(\phi)$ for every $\phi \in \mathcal{F}$. If $\Gamma(\mathcal{F}, P)$ is fat then there exists $\epsilon \in [0, 1)^{\mathcal{F}}$ such that $\Gamma(\mathcal{F}, P - \epsilon) \neq \emptyset$ and $\Gamma(\mathcal{F}, P - \epsilon)$ is fat. It is also obvious that $\Gamma(\mathcal{F}, P - \epsilon) \subseteq \Gamma(\mathcal{F}, P)$. We say such an ϵ is $(\mathcal{F}, P)$-*admissible*. If $\epsilon \in (0, 1)^{\mathcal{F}}$, i.e., $\epsilon(\phi) > 0$ for all $\phi \in \mathcal{F}$, we denote it as $\epsilon > 0$. Otherwise, we write $\epsilon \geqslant 0$.

Construction A. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0, 1]^{\mathcal{F}}$. For every $m \in \mathbb{N}$ we construct $R_m(\mathcal{F}, P) \subseteq \Sigma^*$ by defining $R_m(\mathcal{F}, P) = \mathcal{B}_m(\mathcal{F}, P)^*$.* □

By definition, $R_m(\mathcal{F}, P)$ from Construction A is a regular language. It may be presented as the language of the following graph G: the graph contains a single vertex, all the edges are self loops and are labeled by the words of $\mathcal{B}_m(\mathcal{F}, P)$, i.e., the length-$m$ words in $\mathcal{B}(\mathcal{F}, P)$.

Theorem 6. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0, 1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Then for any $(\mathcal{F}, P)$-admissible $\epsilon > 0$, there exists $M_\epsilon \in \mathbb{N}$ such that for all $m > M_\epsilon$*

$$R_m(\mathcal{F}, P - \epsilon) \subseteq \mathcal{B}(\mathcal{F}, P).$$

Proof sketch: We upper bound the frequency of offending sequences, and connect m with ϵ. This is also used later when constructing encoders. ■

The following theorem shows that the sequence of systems $R_m(\mathcal{F}, P - \epsilon)$ has a capacity that approaches $\mathsf{cap}(\mathcal{B}(\mathcal{F}, P - \epsilon))$ as m grows.

Theorem 7. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0, 1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Then for every $(\mathcal{F}, P)$-admissible $\epsilon \geqslant 0$ the following limit exists and*

$$\lim_{m \to \infty} \mathsf{cap}(R_m(\mathcal{F}, P - \epsilon)) = \mathsf{cap}(\mathcal{B}(\mathcal{F}, P - \epsilon)).$$

Proof sketch: We find $\mathsf{cap}(R_m(\mathcal{F}, P - \epsilon))$ explicitly, and use the fat $\Gamma(\mathcal{F}, P)$ to show that the limit exists and is $\mathsf{cap}(\mathcal{B}(\mathcal{F}, P - \epsilon))$. ■

If $\epsilon_1, \epsilon_2 \in (0,1)^{\mathcal{F}}$, we say $\epsilon_1 \leqslant \epsilon_2$ if $\epsilon_1(\phi) \leqslant \epsilon_2(\phi)$ for all $\phi \in \mathcal{F}$. We note that if ϵ_2 is $(\mathcal{F}, P)$-admissible and $\epsilon_1 \leqslant \epsilon_2$, then ϵ_1 is also $(\mathcal{F}, P)$-admissible. Additionally, if $\epsilon_1, \epsilon_2, \dots$ is a sequence of $(\mathcal{F}, P)$-admissible functions, we say $\lim_{i\to\infty} \epsilon_i = 0$ if $\lim_{i\to\infty} \epsilon_i(\phi) = 0$ for all $\phi \in \mathcal{F}$.

Corollary 8. *For any SCS $(\mathcal{F}, P)$ with a fat $\Gamma(\mathcal{F}, P)$ there exist block encoders with rate arbitrarily close to $\mathsf{cap}(\mathcal{B}(\mathcal{F}, P))$.*

While the block encoders we constructed are quite simple, and have rate p/q arbitrarily close to $\mathsf{cap}(\mathcal{B}(\mathcal{F}, P))$, we do however point a major drawback. The edges are labeled by words from Σ^m. Thus, the encoder is not $p : q$ but $mp/q : m$. For a fair comparison with the next construction, if we transform this to an encoder with labels from Σ (e.g., via a standard tree argument), the anticipation becomes $\Omega(m)$, which is undesirable.

B. Sliding-Block Encoders

Unlike Construction A, in which a sequence was a concatenation of independent blocks, the construction we now present has a sliding-window restriction.

Construction B. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \in \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$. For every $m \in \mathbb{N}$ we construct $N_m(\mathcal{F}, P) \subseteq \Sigma^*$ by defining $N_m(\mathcal{F}, P) = \{\omega \in \Sigma^* : \mathrm{sub}_m(\omega) \subseteq \mathcal{B}(\mathcal{F}, P)\}$.* □

We observe that $N_m(\mathcal{F}, P)$ from Construction B is a fully-constrained system. Indeed, it is defined by a finite set of forbidden words, $\Sigma^k \setminus \mathcal{B}_m(\mathcal{F}, P)$.

For the purpose of building an encoder, we construct a labeled graph G that presents $N_m(\mathcal{F}, P)$. The vertex set is defined as $V = \bigcup_{i=0}^{m-1} \Sigma^i$. The edges, with labels from Σ, are given by

$$a_0 a_1 \dots a_i \xrightarrow{a_{i+1}} a_0 a_1 \dots a_i a_{i+1},$$

for all $0 \leqslant i \leqslant m-2$ and $a_j \in \Sigma$ for all j, as well as

$$a_0 a_1 \dots a_{m-2} \xrightarrow{a_{m-1}} a_1 \dots a_{m-2} a_{m-1},$$

for all $a_0 a_1 \dots a_{m-2} a_{m-1} \in \mathcal{B}(\mathcal{F}, P)$ and $a_j \in \Sigma$ for all j.

It is easily observed that every path of length $m-1$ labeled by $\omega \in \Sigma^{m-1}$ ends in the vertex labeled by ω. From then on, by simple induction, assuming $\omega'\omega$ is a label of a path with $\omega \in \Sigma^{m-1}$, then the path ends in the vertex ω and a letter $a \in \Sigma$ may be generated following that path if and only if $\omega a \in \mathcal{B}(\mathcal{F}, P)$.

Theorem 9. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Then for any $(\mathcal{F}, P)$-admissible $\epsilon > 0$, and for all $m \geqslant k$,*

$$N_m(\mathcal{F}, P - \epsilon) \subseteq^e \mathcal{B}(\mathcal{F}, P).$$

Proof sketch: We upper bound the frequency of offending words, showing it exceeds the semiconstraints at most as a vanishing function of the length of the word. Apart from proving the claim, it assists in the proof of the next theorem, and in constructing encoders later on. ■

A stronger statement than that of Theorem 9 may be made in the case WSCS, in which ϵ is removed.

Theorem 10. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$, and tolerance function $\xi(n) = \max_{\phi\in\mathcal{F}} (P(\phi)(k-1)(n-k+1))$. Then for all $m \geqslant k$,*

$$N_m(\mathcal{F}, P) \subseteq^e \overline{\mathcal{B}}(\mathcal{F}, P).$$

Theorem 11. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Then*

$$\limsup_{m\to\infty} \mathsf{cap}(N_m(\mathcal{F}, P)) = \mathsf{cap}(\mathcal{B}(\mathcal{F}, P)).$$

Proof sketch: We upper bound the capacity using weakly-admissible words with Theorem 10 and Theorem 5. For the lower bound we prove that for large enough m, $\mathsf{cap}\left(N_{m^2}(\mathcal{F}, P)\right) \geqslant \mathsf{cap}(R_m(\mathcal{F}, P - \epsilon))$, and carefully take appropriate limits. ■

The graph G that presents $N_m(\mathcal{F}, P)$, as described above, is $(m, 0)$-definite, i.e., all the paths that generate a given word of length $m+1$ symbols agree on the edge that generated the last symbol. The graph is not necessarily an encoder (due to an unequal out-degree), but by using the state-splitting algorithm on G we may generate a $p : q$ encoder. Compared with the block encoder from the previous section however, this encoder may have an exponential number of states (in m).

C. A Short Case Study

As a short case study we provide the following example. Consider the SCS over $\Sigma = \{0,1\}$, which is defined by $\mathcal{F} = \{11\}$, and $P(11) = 0.205$. This SCS was called the $(0,1,0.205)$-RLL SCS in [4], and its capacity is $\mathsf{cap}(\mathcal{B}(\mathcal{F}, P)) \approx 0.98$. We investigate the encoders presented thus far, with an intention of building an encoder with rate $\frac{3}{4}$.

We first focus on the block encoder associated with $R_m(\mathcal{F}, P)$. Choosing $\epsilon = 0.005$, a quick use of the proof of Theorem 6 shows that we need $m \geqslant 16$ in order to satisfy the semiconstraints, i.e., we need to take blocks from $\mathcal{B}_{16}(\mathcal{F}, 0.2)$. This is indeed tight, since for any $5 \leqslant m' \leqslant 15$ we have $\omega = 1^{\lfloor 0.2m\rfloor} 0^{m'-1-\lfloor 0.2m\rfloor} 1 \in \mathcal{B}_{m'}(\mathcal{F}, 0.2)$, and $\mathrm{fr}^2_{\omega^i}(11) > 0.205$ for large enough i. Similar arguments hold for $m' \leqslant 4$. Taking $m = 16$ is adequate, since $|\mathcal{B}_{16}(\mathcal{F}, 0.2)| = 32274$ giving a rate of ≈ 0.93. As expected, a graph with this amount of edges is unwieldy, requiring large look-up tables or enumerative-coding techniques.

On the other hand, the encoder associated with $N_m(\mathcal{F}, P)$ is much simpler. We can choose $m = 5$ without exceeding the semiconstraints since the upper bound on the frequency of 11 approaches 0.2 as the length of the word increases (see proof of Theorem 9). We first construct the modified De-Bruijn graph of order $m - 1 = 4$ where we allow only one appearance of the pattern 11. Since we would like to build an encoder with rate $\frac{3}{4}$, we take the graph to its 4th power, and keep the appropriate irreducible subgraph. Combining vertices with the same follower sets we obtain the graph presented in Fig. 1. Using the state splitting algorithm we split the vertex 0000 and obtain the following graph which can be used as an encoder after removing some edges. Using the proof of Theorem 9, any word of length longer than 41 satisfies the semiconstraints.

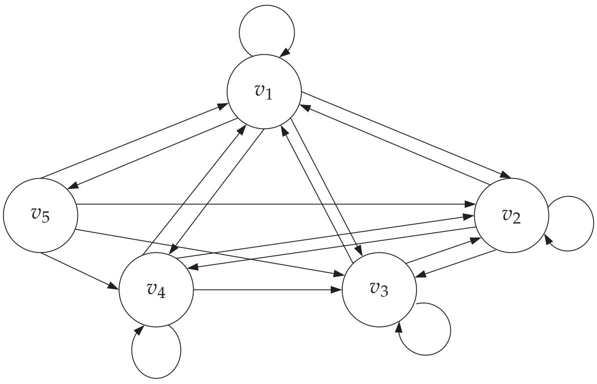

Figure 1. The appropriate irreducible subgraph of the 4th power of the modified De-Bruijn graph after combining states with the same follower sets. To reduce notation length, 4-tuples of bits are written in hexadecimal notation. The state combinations are given by $v_1 = \{0,2,4,8,A,C\}$, $v_2 = \{1,5,9\}$, $v_3 = \{3,B\}$, $v_4 = \{6\}$, and $v_5 = \{D\}$. The parallel edges' labels are $\mathcal{L}(v_1 \to v_1) = \{0,2,4,8,A,C\}$, $\mathcal{L}(v_1 \to v_2) = \mathcal{L}(v_2 \to v_2) = \mathcal{L}(v_4 \to v_2) = \{1,5,9\}$, $\mathcal{L}(v_1 \to v_3) = \mathcal{L}(v_4 \to v_3) = \{3,B\}$, $\mathcal{L}(v_1 \to v_4) = \mathcal{L}(v_2 \to v_4) = \mathcal{L}(v_4 \to v_4) = \mathcal{L}(v_5 \to v_4) = \{6\}$, $\mathcal{L}(v_1 \to v_5) = \{D\}$, $\mathcal{L}(v_2 \to v_1) = \mathcal{L}(v_4 \to v_1) = \{0,2,4,8,A\}$, $\mathcal{L}(v_2 \to v_3) = \mathcal{L}(v_3 \to v_3) = \mathcal{L}(v_5 \to v_3) = \{3\}$, $\mathcal{L}(v_3 \to v_1) = \mathcal{L}(v_5 \to v_1) = \{0,2,4\}$, $\mathcal{L}(v_3 \to v_2) = \mathcal{L}(v_5 \to v_2) = \{1,5\}$.

IV. Approaching Capacity from Above

In this section we consider the dual question to the one asked in Section III. We now ask which fully-constrained systems, presented as the language of a directed labeled graph, contain a given semiconstrained system. Additionally, we would like to know whether the capacity of a sequence of those fully-constrained system can approach the capacity of the semiconstrained system in the limit. As we shall soon see, the answer is quite pessimistic. We first give an auxiliary lemma, and then proceed to prove the main theorem.

Lemma 12. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Then for all $\alpha \in \Sigma^*$ there exists $\beta \in \Sigma^*$ such that $\alpha\beta \in \mathcal{B}(\mathcal{F}, P)$, i.e., any finite prefix may be completed to a word in the semiconstrained system. Additionally, there exists $m \in \mathbb{N}$ such that $\mathcal{B}(\mathcal{F}, P)$ contains a word of length m and a word of length $m+1$.*

Proof sketch: Since $\Gamma(\mathcal{F}, P)$ is fat and the rationals are dense, we find a rational shift-invariant probability measure in $\Gamma(\mathcal{F}, P)$. It is used to build a modified De-Bruijn graph over which Eulerian cycles correspond to strings with the desired statistics. Of these, we find a cycle with the correct prefix. ∎

Theorem 13. *Let $(\mathcal{F}, P)$ be a SCS, $\mathcal{F} \subseteq \Sigma^k$, $P \in [0,1]^{\mathcal{F}}$, with a fat $\Gamma(\mathcal{F}, P)$. Let $(\mathcal{F}', 0)$ be any fully constrained system such that $\mathcal{B}(\mathcal{F}, P) \subseteq \mathcal{B}(\mathcal{F}', 0)$. Then $\mathcal{B}(\mathcal{F}', 0) = \Sigma^*$.*

Proof sketch: Using Lemma 12 every graph presenting $\mathcal{B}(\mathcal{F}', 0)$ is shown to even single-letter labels, and must generate every prefix, i.e., Σ^*. ∎

V. Conclusions and Discussion

Other results, not reported in this paper, contain a treatment of a natural extension that enables the analysis of probability measures with 0 semiconstraints, i.e., a mix semiconstraints and full constraints. In that case, for example, fully-constrained systems containing the SCS are no longer the entire space Σ^*.

Additionally, the definition of semiconstraints is generalized even further, and we define SCS simply by a set of allowed probability measures, $\Gamma \subseteq \mathcal{P}(\Sigma^k)$. More importantly, we expand on a connection between SCS and certain spaces of infinite sequences. In the classical fully-constrained case, a constrained system corresponds directly to a subshift of finite type – a special type of compact invariant subset of $\Sigma^{\mathbb{Z}}$. Thus, coding theory of fully-constrained systems is closely related to symbolic dynamics. However, in the SCS case, any reasonable space of sequences associated to a fat SCS would be dense in $\Sigma^{\mathbb{Z}}$. This leads to problems in dynamics on certain non-compact spaces, and naturally brings in ergodic-theory considerations.

Other open questions remain. In particular, we mention the lack of bounds on encoder parameters, such as number of states and anticipation. We leave this problem for future works.

References

[1] J.-R. Chazottes, J.-M. Gambaudo, M. Hochman, and E. Ugalde, "On the finite-dimensional marginals of shift-invariant measures," *Ergodic Theory Dyn. Syst.*, vol. 32, no. 5, pp. 1485–1500, 2012.

[2] O. Elishco, T. Meyerovitch, and M. Schwartz, "Semiconstrained systems," in *Proceedings of the 2015 IEEE International Symposium on Information Theory (ISIT2015), Hong Kong, China SAR*, Jun. 2015, pp. 246–250.

[3] ——, "Encoding semiconstrained systems," *arXiv preprint arXiv:1601.05594*, 2016.

[4] ——, "Semiconstrained systems," *IEEE Trans. Inform. Theory*, 2016, accepted.

[5] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.

[6] R. Karabed, D. Neuhoff, and A. Khayrallah, "The capacity of costly noiseless channels," IBM Research Report, Tech. Rep. RJ 6040 (59639), Jan. 1988.

[7] S. Kayser and P. H. Siegel, "Constructions for constant-weight ICI-free codes," in *Proceedings of the 2014 IEEE International Symposium on Information Theory (ISIT2014), Honolulu, HI, USA*, Jul. 2014, pp. 1431–1435.

[8] A. S. Khayrallah and D. L. Neuhoff, "Coding for channels with cost constraints," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 854–867, 1996.

[9] V. Y. Krachkovsky, R. Karabed, S. Yang, and B. A. Wilson, "On modulation coding for channels with cost constraints," in *Proceedings of the 2014 IEEE International Symposium on Information Theory (ISIT2014), Honolulu, HI, USA*, Jun. 2014, pp. 421–425.

[10] O. F. Kurmaev, "Constant-weight and constant-charge binary run-length limited codes," *IEEE Trans. Inform. Theory*, vol. 57, no. 7, pp. 4497–4515, Jul. 2011.

[11] D. Lind and B. H. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1985.

[12] M. Qin, E. Yaakobi, and P. H. Siegel, "Constrained codes that mitigate inter-cell interference in read/write cycles for flash memories," *IEEE J. Select. Areas Commun.*, vol. 32, no. 5, pp. 836–846, May 2014.

[13] A. Shafarenko, A. Skidin, and S. K. Turitsyn, "Weakly-constrained codes for suppression of patterning effects in digital communications," *IEEE Trans. Communications*, vol. 58, no. 10, pp. 2845–2854, Oct. 2010.

[14] A. Shafarenko, K. S. Turitsyn, and S. K. Turitsyn, "Information-theory analysis of skewed coding for suppression of pattern-dependent errors in digital communications," *IEEE Trans. Communications*, vol. 55, no. 2, pp. 237–241, Feb. 2007.

[15] J. Shallit, *A Second Course in Formal Languages and Automata Theory*. Cambridge University Press, 2008.

[16] J. B. Soriaga and P. H. Siegel, "On the design of finite-state shaping encoders for partial-response channels," in *Proceedings of the 2006 Information Theory and Application Workshop (ITA2006), San Diego, CA, USA*, Feb. 2006.