Bounds on the Number of States in Encoder Graphs for Input-Constrained Channels

Brian Marcus* Ron M. Roth*†

October, 1990

Abstract

We exhibit lower bounds on the number of states in a fixed rate finite-state encoder that maps unconstrained n-ary sequences into a given set of constrained sequences, defined by a finite labelled graph G. In particular, one simple lower bound is given by $\min_{\mathbf{x}} \max_v x_v$ where $\mathbf{x} = [x_v]$ ranges over certain (nonnegative integer) approximate eigenvectors of the adjacency matrix for G. In some sense, our bounds are close to what can be realized by the state splitting algorithm, and in some cases, they are shown to be tight. In particular, these bounds are used to show that the smallest (in number of states) known encoders for the $(1, 7)$ and $(2, 7)$ run-length-limited systems are indeed the smallest possible. For any given constrained set S of sequences, we apply these bounds to study the growth of the number of states in families of encoders whose rates approach the capacity of S.

Key words: Input-constrained discrete channel; Run-length-limited system; Sofic system.

*IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120.

†On leave from the Computer Science Department, Technion — Israel Institute of Technology, Haifa 32000, Israel.

1. Introduction

Input-constrained discrete noiseless channels have been studied extensively in recent years in [1][6][13], among many others. These channels are modelled as constrained systems i.e., sets of sequences obtained by reading the labels of finite directed labelled graphs. There exist algorithms such as the state splitting algorithm [1] and improvements thereof (see [2][15][19][21]), which yield finite-state encoders for these systems. Obviously, the number of states in any encoder affects the complexity of its hardware implementation (e.g., the number of memory-units required to represent those states).

The state splitting algorithm yields encoders whose number of states is at most the *sum* of the components of some so-called approximate eigenvector, which depends on the constrained system and the designed rate. In this paper we obtain general lower bounds on the number of states in any encoder for a given constrained system and rate. In particular, one lower bound is given by the *maximum* component of some approximate eigenvector (Section 4). Thus, our lower bounds are close to what can be realized by the state splitting algorithm. Later in this paper (Section 8) we extend the techniques of Section 4 to obtain even tighter bounds. In particular, these bounds are used to show that the smallest (in number of states) known encoders for the well-known $(1,7)$ and $(2,7)$ run-length-limited constrained systems are indeed the smallest possible (Section 8 and Appendix C). In Section 5 we discuss other classes of constrained systems where the lower bounds are tight. Then, in Section 6, we apply these bounds to study the growth of the number of states in families of encoders whose rates approach the capacity of a given constrained system. We also discuss special classes of encoders e.g., those that have sliding block decoders (Section 7). Finally, some computational remarks are given in Section 9.

2. Definitions

A *labelled graph* $G = (V, E, L)$ is a finite directed graph with states V, edges E, and a labelling $L : E \to \Sigma$ for some finite alphabet Σ. We will often write $V(G)$ for V and $E(G)$ for E. We usually refer to a labelled graph as simply a graph.

A *constrained system* is the set of all words (i.e., finite sequences) obtained from reading the labels of paths in a graph G (unless otherwise specified, *path* means finite path). We call the set of sequences $S(G)$, and we say that G *presents* S. If a path is labelled by a word $\mathbf{w}$, we say that the path *generates* $\mathbf{w}$. The length of a word $\mathbf{w}$ is denoted $\ell(\mathbf{w})$. Constrained systems go by various names: in automata theory, this is essentially a regular language; in symbolic dynamics, this is a sofic system (although usually a sofic system means the set of *bi-infinite* sequences obtained from a graph).

A graph G is *deterministic* if for each state $v \in V$ the outgoing edges from v are distinctly labelled. Every constrained system has a deterministic presentation; this is proved by an easy modification of the well known subset construction in automata theory (this construction is reviewed in the proof of part (a) of Theorem 3).

A graph G is *lossless* if any two distinct paths with the same initial state and terminal state have different labellings. Note that any deterministic graph is lossless.

The notion of losslessness is due to Huffman [17]; see also Even [9]. These terms have other names in other fields; deterministic is sometimes called right resolving (in symbolic dynamics) or unifilar (in source coding), and lossless is sometimes called unambiguous (in automata theory) or finite-to-one (in symbolic dynamics).

Let S be a constrained system and n be a positive integer. An (S, n)-*encoder* is a labelled graph $\mathcal{E}$ such that

(a) $S(\mathcal{E}) \subseteq S$;

(b) each state of $\mathcal{E}$ has out-degree n;

(c) $\mathcal{E}$ is lossless.

With a choice of an initial state, such an encoder can be used to encode arbitrary n-ary sequences into the words of S by generating at each encoder state one output symbol of the labelling alphabet Σ for each input n-ary symbol (rate $1 : 1$). The losslessness condition (c) above is the minimal possible condition that one would want to impose on an encoder so that encoded sequences can be decoded: knowledge of the initial state, terminal state and codeword uniquely determines the encoder path and therefore the sequence that was encoded; conversely if $\mathcal{E}$ is not lossless, then (with some choice of initial state) there is some initial n-ary block $\mathbf{b}$ such that any n-ary block that has $\mathbf{b}$ as a prefix will be encoded to a sequence that cannot possibly be decoded.

Any finite-state machine $\mathcal{M}$ which encodes arbitrary binary sequences into the words of S at constant rate $p : q$ can be viewed as an encoder in our sense: namely, let S^q denote the constrained system obtained by dividing the sequences of S into non-overlapping q-blocks — thus regarding the sequences of S over the alphabet Σ^q; then the machine $\mathcal{M}$ can be viewed as an $(S^q, 2^p)$-encoder, and conversely. Note that S^q is indeed a constrained system: let G^q be the graph which has the same states as G, has edges from u to v for each path of length q from u to v in G, and labels inherited from the labels of these corresponding paths; if S is presented by G, then S^q is presented by G^q.

Given a graph G and a state $u \in V(G)$, the *follower set* $\mathcal{F}_G(u)$ is the set of words generated by paths in G which start at u. The follower set $\mathcal{F}_G(U)$ of a subset $U \subseteq V(G)$ is the union of the follower sets of the elements of U. The notion of follower set was exploited in [21] to reduce the number of states in encoders produced by the state splitting algorithm.

We will also need a slightly different notion of follower set: for $u \in V(G)$, the *semi-infinite follower set* $\mathcal{F}_G^\infty(u)$ is the set of semi-infinite sequences generated by semi-infinite paths that begin at u. Note that the follower set completely determines the semi-infinite follower set. Conversely, for *decent* graphs, i.e., graphs, each of whose states have at least one outgoing edge, the semi-infinite follower set completely determines the follower set. So, for decent graphs the two notions determine one another. Note also that for decent graphs inclusion of follower sets is the same as inclusion of the corresponding semi-infinite follower sets, but disjointness of follower sets can be a strictly stronger condition than disjointness of the corresponding semi-infinite follower sets. For example, if all words of length ≥ 2 generated from state u begin with 00 and all words of length ≥ 2 generated from state v begin with 01, then the semi-infinite follower sets are disjoint, whereas the follower sets are not.

A graph G is *irreducible* (or *strongly connected*) if for every two states $u, v \in V(G)$ there exists a path in G from u to v. In particular, irreducible graphs are decent. Given a graph G, we say that two states $u, v \in V(G)$ are *bi-connected* if there exists a path in G from u to v and vice versa. By definition, every state is bi-connected with itself (say, by virtue of the empty path) and therefore, bi-connection is an equivalence relation. An *irreducible* *component* G' of G is a subgraph of G whose set of states $V(G')$ forms an equivalence class of the bi-connection relation, and whose edges are all those in G that begin and terminate at elements of $V(G')$. Clearly, every irreducible component is an irreducible subgraph of G.

An *irreducible sink* of a graph G is an irreducible component G' of G such that the outgoing edges from $V(G')$ in G all terminate in $V(G')$. It is easy to verify that every graph has an irreducible sink.

Given a graph G, the adjacency matrix $A_G = \left[(A_G)_{u,v}\right]$ is the $|V(G)| \times |V(G)|$ matrix with $(A_G)_{u,v}$ being the number of edges from state u to state v (note that the adjacency matrix A_G depends only on the vertices and edges of G and not on the labelling L). Observe that $(A_G)^k = A_{G^k}$.

A nonnegative square matrix $A = [A_{u,v}]$ is called *irreducible* if for every u and v there exists an integer $m(u,v)$ such that $(A^{m(u,v)})_{u,v} > 0$. Clearly, a graph G is irreducible if and only if A_G is irreducible.

Given a square matrix A and an integer n, an (A,n)-*approximate eigenvector* is a nonnegative integer vector $\mathbf{x} \not\equiv \mathbf{0}$ such that $A\mathbf{x} \geq n\mathbf{x}$. The set of all such vectors is denoted by $\mathcal{X}(A,n)$.

The *capacity* $c(S)$ of a constrained system S is the asymptotic growth rate of the number of words in S i.e.,

$$c(S) = \lim_{n\to\infty} \frac{1}{n} \log |\,\text{words of length } n \text{ in } S\,| \ .$$

Hereafter $\log(\cdot)$ is taken to base 2.

The basic properties of the capacity of a constrained system $S = S(G)$, in terms of eigenvalues, eigenvectors and approximate eigenvectors of the corresponding adjacency matrix A_G, are summarized in the proposition following the well-known theorems below.

Hereafter, $\lambda(A)$ denotes the spectral radius (i.e., largest of the absolute values of the eigenvalues) of a matrix A.

Theorem 1. (Perron-Frobenius Theorem [22, p. 14]). *Let A be a nonnegative matrix. Then $\lambda(A)$ is an eigenvalue of A. Moreover, there is a nonnegative eigenvector associated with $\lambda(A)$.*

Furthermore,

Theorem 2. (Perron-Frobenius Theorem for irreducible matrices [22, Ch. 1][23, Ch. 1][25, Ch. 2]). *Let A be an irreducible matrix.*

(i) $\lambda(A) > 0$ and A has a positive (i.e., all components are positive) eigenvector associated with the eigenvalue $\lambda(A)$;

(ii) the algebraic (and, therefore, geometric) multiplicity of the eigenvalue $\lambda(A)$ is 1;

and —

(iii) if $A\mathbf{x} \geq \lambda(A)\mathbf{x}$ for a nonnegative vector $\mathbf{x} \not\equiv \mathbf{0}$, then $\mathbf{x}$ is an eigenvector for A associated with $\lambda(A)$.

Remark 1. If A is an irreducible matrix and $\lambda(A)$ is a positive integer, more can be said. In this case, by Theorem 2 (part (i)) and Gaussian elimination, there is a positive *integer* eigenvector associated with $\lambda(A)$; then by Theorem 2 (part (ii)), there is a smallest (componentwise) positive integer eigenvector $\mathbf{x}$ associated with $\lambda(A)$; finally, by Theorem 2 (parts (ii) and (iii)), $\mathcal{X}(A, \lambda(A))$ consists of all positive integer multiples of $\mathbf{x}$.

Proposition 1. *Let G be a lossless (in particular, deterministic) graph, $S = S(G)$, and let n be a positive integer.*

(i) $c(S) = \log \lambda(A_G)$;

(ii) $c(S) \geq \log n \Longleftrightarrow \lambda(A_G) \geq n \Longleftrightarrow \mathcal{X}(A_G, n) \neq \emptyset$.

We remark that when G is irreducible, a stronger version of part (i) holds i.e., $c(S) = \log \lambda(A_G)$ if and only if G is lossless.

Proof. As these are all fairly well-known, we give just some rough ideas and pointers to references.

Let $\lambda = \lambda(A_G)$. For (i), first show that $\log \lambda$ is the growth rate of the number of paths in G; this can be done by analyzing the solution of the linear vector recurrence $\mathbf{x}_{m+1} = A_G \mathbf{x}_m$, where the u-th component of $\mathbf{x}_m$ stands for the number of paths of length m emanating from u in G. Then use losslessness to show that the ratio of the number of paths of length m in G to the number of words of length m in S is uniformly (in m) bounded above and below.

For (ii), the first "$\Longleftrightarrow$" is immediate from (i); for the second "$\Longleftrightarrow$", if $\lambda > n$, one can perturb a nonnegative λ-eigenvector to get a nonnegative integer vector $\mathbf{x} \not\equiv \mathbf{0}$ such that $A\mathbf{x} \geq n\mathbf{x}$ i.e., $\mathbf{x} \in \mathcal{X}(A_G, n)$; if $\lambda = n$, then some eigenvector itself will be an element of $\mathcal{X}(A_G, n)$; the converse follows from the Perron-Frobenius Theory [25, Lemma 2.2]. $\square$

(To show the converse of part (i) for the irreducible case, note that, by definition, a violation of losslessness produces two paths with the same initial state, terminal state and labelling. Use this to show that the growth rate of paths in G is strictly larger than the growth rate of sequences in S.)

3. Irreducible systems

A constrained system S is *irreducible* if for every pair of words $\mathbf{w}, \mathbf{z}$ in S, there is a word $\mathbf{t}$ such that $\mathbf{wtz}$ is in S. In this section, we develop some basic properties of irreducible systems. We will see, in the next section, that if a lower bound on the number of states of an encoder can be established for irreducible constrained systems, then the bound naturally extends to a lower bound for all constrained systems.

The following shows that irreducibility of a constrained system can be reformulated in terms of irreducible graphs.

Lemma 1. *Let S be a constrained system. The following are equivalent:*

(i) S is irreducible;

(ii) S is presented by some irreducible graph.

Proof. For (ii) $\Rightarrow$ (i), simply connect the terminal state of a path that generates $\mathbf{w}$ to the initial state of a path that generates $\mathbf{z}$. And (i) $\Rightarrow$ (ii) is obtained by replacing inclusion with equality in the following stronger statement (which is needed later on in Section 4).

Lemma 2. *Let S be an irreducible constrained system and let G be a graph such that $S \subseteq S(G)$. Then for some irreducible component G' of G, $S \subseteq S(G')$.*

Proof. Let $G_1, G_2, \ldots, G_k$ denote the irreducible components of G. We prove the lemma by contradiction. Suppose that for each $i = 1, 2, \ldots, k$, there is a word $\mathbf{w}_i$ in S but not in $S(G_i)$. Since S is irreducible, there is a word $\mathbf{w}$ that contains a copy of each $\mathbf{w}_i$; moreover, for any positive integer n, there is a word $\mathbf{z}$ that contains n non-overlapping copies of $\mathbf{w}$. Let $n = |V(G)| + 1$, and for this n let γ be a path in G that generates $\mathbf{z}$. Then γ can be written as $\gamma = \gamma_1\gamma_2 \ldots \gamma_n$ where each γ_j has a sub-path which generates $\mathbf{w}$. For some $j < k$, the initial states of γ_j, γ_k coincide. Thus, $\gamma_j\gamma_{j+1} \ldots \gamma_{k-1}$ is a cycle (i.e., begins and terminates at the same state) and has a sub-path that generates $\mathbf{w}$. Now, by definition, any cycle in a graph must belong to some irreducible component, say G_i, and thus $\mathbf{w}_i$ is in $S(G_i)$, contrary to the definition of $\mathbf{w}_i$. $\square$

The *Shannon cover* G_S of an irreducible constrained system S is a deterministic graph, which presents S, with the minimum number of states. The Shannon cover has the following well-known properties (in particular, it is unique), mostly due to Fischer [10][11].

Proposition 2. *Let S be an irreducible constrained system.*

(i) The Shannon cover G_S is the unique (up to graph isomorphism) deterministic irreducible graph which presents S and satisfies the condition that the follower sets of distinct states are distinct. (Here, two labelled graphs are isomorphic if they are the same, up to renaming of states.)

(ii) For any irreducible deterministic presentation G of S, the follower sets of G coincide with the follower sets of G_S.

(iii) For any two irreducible deterministic presentations G and H of S, the follower sets of G and H coincide.

Proof. The proof of (i) is contained in [10][11]. For the proof of (ii), apply the well-known state reduction algorithm to G; i.e., iteratively apply the procedure which merges states with the same follower sets until the follower sets of distinct states are distinct. This yields, by (i), the Shannon cover. At each step along the way, we pass from a graph G to a graph H that have the same follower sets, thus obtaining (ii). Finally, (iii) follows immediately from (ii). $\square$

The states of the Shannon cover can therefore be regarded as equivalence classes induced on the states of an irreducible deterministic presentation G of S by the equivalence relation $u \equiv u' \iff \mathcal{F}_G(u) = \mathcal{F}_G(u')$, $u, u' \in V(G)$. For any irreducible deterministic graph G, let Φ_G denote the map which sends each vertex in G to its equivalence class, viewed as a vertex in G_S; then for all $u \in V(G)$, u and $\Phi(u)$ have the same "outgoing picture" i.e., the same number of outgoing edges with the same distinct labels.

From this, we see that one cannot in general hope to get encoders that are deterministic. For example, if S is an irreducible constrained system with $c(S) = \log n$, and the Shannon cover is not an (S, n)-encoder (i.e., there are some states that do not have out-degree n), then no deterministic presentation of S can possibly be an (S, n)-encoder.

Now, we generalize part (iii) of Proposition 2 to the situation where H presents a subsystem of $S(G)$. This is a key element in the proof that establishes our lower bounds.

Lemma 3. *Let G and H be two irreducible deterministic graphs such that $S(H) \subseteq S(G)$. Then, for every $v \in V(H)$ there exists $u \in V(G)$ such that $\mathcal{F}_H(v) \subseteq \mathcal{F}_G(u)$.*

Proof. We form the so-called *fiber product* graph $G * H$ of G and H as follows: The states of $G * H$ are defined by $V(G * H) = \{\langle u, u' \rangle \,|\, u \in V(G), u' \in V(H)\}$. Now, for every edge e from u to v in $V(G)$, and every edge e' from u' to v' in $V(H)$ with the same label a, we draw an edge labelled a from $\langle u, u' \rangle$ to $\langle v, v' \rangle$ in $G * H$.

It is easy to verify that $G * H$ is deterministic. Furthermore, a word $\mathbf{w}$ is generated by $G*H$ if and only if it is generated by both G and H. Hence, $S(G*H) = S(G) \cap S(H) = S(H)$. By Lemma 1, $S(H)$ is irreducible, and thus by Lemma 2 there is an irreducible component G' of $G * H$ such that $S(G') = S(G * H) = S(H)$. By Proposition 2 (part (iii)), for every state $v \in V(H)$ there exists a state $\langle u, u' \rangle \in V(G') \subseteq V(G * H)$ such that

$$\mathcal{F}_H(v) = \mathcal{F}_{G'}\Big(\langle u, u' \rangle\Big) \subseteq \mathcal{F}_{G*H}\Big(\langle u, u' \rangle\Big) \subseteq \mathcal{F}_G(u). \qquad \square$$

4. Lower bounds

We now give some lower bounds on the number of states in an (S,n)-encoder. Given any deterministic graph G which presents S, these bounds are largely determined by the (A_G,n)-approximate eigenvectors and by the intersections of follower sets (really, the semi-infinite follower sets) of G. A tighter bound (Theorem 6) will be given later on in Section 8. The proof of Theorem 6 will make use of the simpler (in the statement, proof and computation of the bound) result stated in Theorem 3 below.

The following proposition shows that for the problem of finding lower bounds on the number of states in an (S,n)-encoder, we may assume that S is irreducible.

Denote by $M(S,n)$ the minimum number of states of an (S,n)-encoder.

Proposition 3. *Let S be a constrained system and n be a positive integer. Let G be any presentation of S with irreducible components $G_1, G_2, \ldots, G_k$, and let $S_i = S(G_i)$. Then, whenever $\mathcal{E}$ is an (S,n)-encoder, there is a subgraph $\mathcal{E}'$ of $\mathcal{E}$ which is an (S_i,n)-encoder for some i. In particular,*

$$M(S,n) = \min_i M(S_i,n) \ .$$

Proof. Let $\mathcal{E}'$ be an irreducible sink of $\mathcal{E}$, and let $S' = S(\mathcal{E}')$. Then S' is irreducible, and we can apply Lemma 2 to find i such that $S' \subseteq S_i$. Hence, we have constructed an (S_i,n)-encoder $\mathcal{E}'$ which is a subgraph of $\mathcal{E}$. $\square$

Theorem 3. *Let S be an irreducible constrained system, presented by an irreducible deterministic graph G, and let n be a positive integer. Then, for any (S,n)-encoder $\mathcal{E}$,*

$$|V(\mathcal{E})| \geq \min_{[y_u]_u \in \mathcal{X}(A_G,n)} \max_U \sum_{u \in U} y_u \ , \qquad (1)$$

where the maximum is taken over all subsets $U \subseteq V(G)$ such that $u, u' \in U$, $u \neq u' \Rightarrow \mathcal{F}_G^\infty(u) \cap \mathcal{F}_G^\infty(u') = \emptyset$.

Recall that the condition $\mathcal{F}_G^\infty(u) \cap \mathcal{F}_G^\infty(u') = \emptyset$ is typically weaker than the condition $\mathcal{F}_G(u) \cap \mathcal{F}_G(u') = \emptyset$. Also, the minimum over an empty set is defined as infinity, so, by

Proposition 1, the bound is meaningful only if the adjacency matrix A_G has largest eigenvalue $\geq n$, equivalently, that $c(S) \geq \log n$.

We first present the idea of the proof. For any (S, n)-encoder $\mathcal{E}$ (which, by definition, presents a subsystem of S) we will associate an (A_G, n)-approximate eigenvector $\mathbf{x} = [x_u]_{u \in V(G)}$ whose entries are sizes of subsets of $V(\mathcal{E})$. The subsets which correspond to the entries x_u, $u \in U$, are then shown to be disjoint for every $U \subseteq V(G)$ which satisfies the disjointness condition of the theorem. This, in turn, implies the bound $|V(\mathcal{E})| \geq \sum_{u \in U} x_u$. Then, we maximize the sum $\sum_{u \in U} x_u$ over the sets U and, finally, we eliminate the dependency of $\mathbf{x}$ on the specific encoder $\mathcal{E}$ by minimizing over all (A_G, n)-approximate eigenvectors.

Proof of Theorem 3. Let $\mathcal{E}$ be an (S, n)-encoder. The following construction effectively provides an approximate eigenvector $\mathbf{x}$ which satisfies

$$|V(\mathcal{E})| \geq \sum_{u \in U} x_u \ , \tag{2}$$

for each subset U that satisfies the disjointness condition of the theorem. We break the construction into three pieces.

(a) Construct a so-called determinizing graph H which presents $S' = S(\mathcal{E})$ (this is a slight modification of the well-known subset construction of automata theory). For any word $\mathbf{w}$ and state $v \in V(\mathcal{E})$, let $T_{\mathcal{E}}(\mathbf{w}, v)$ denote the subset of states in $\mathcal{E}$ which are accessible from v by paths in $\mathcal{E}$ which generate $\mathbf{w}$. When $\mathbf{w}$ is the empty word (denoted $\boldsymbol{\epsilon}$), define $T_{\mathcal{E}}(\boldsymbol{\epsilon}, v) = \{v\}$. The states of H are defined as the distinct nonempty subsets $\{T_{\mathcal{E}}(\mathbf{w}, v)\}_{\mathbf{w}, v}$ of $V(\mathcal{E})$. As for the edges of H, for any two states $Z, Z' \in V(H)$ we draw an edge labelled a from Z to Z' if and only if there exists a state $v \in V(\mathcal{E})$ and a word $\mathbf{w}$ such that $Z = T_{\mathcal{E}}(\mathbf{w}, v)$ and $Z' = T_{\mathcal{E}}(\mathbf{w}a, v)$. In other words, an edge in H labelled a from Z leads to the subset Z' of $\mathcal{E}$-states, each of which is accessible in $\mathcal{E}$ from some $\mathcal{E}$-state in Z by an edge labelled a. Note that hereafter we regard Z and Z' as either states of H, or subsets of $V(\mathcal{E})$.

By definition, H is deterministic. We now claim that $S(H) = S(\mathcal{E}) = S'$. Indeed, if a word $\mathbf{w} = w_1 w_2 \ldots w_\ell$ is generated by paths in $\mathcal{E}$ starting at state v, then $\mathbf{w}$ is also generated by the path

$$\{v\} = T_{\mathcal{E}}(\boldsymbol{\epsilon}, v) \to T_{\mathcal{E}}(w_1, v) \to T_{\mathcal{E}}(w_1 w_2, v) \to \ \ldots \ \to T_{\mathcal{E}}(w_1 w_2 \ \ldots \ w_\ell, v)$$

in H. Conversely, if $\mathbf{w}$ is generated by H starting at a state $Z = T_{\mathcal{E}}(\mathbf{w}', v)$, then, by the definition of H, $\mathbf{w}'\mathbf{w}$ is generated by $\mathcal{E}$, starting at state v.

(b) For any irreducible sink H' of H, define a vector $\mathbf{c} \not\equiv \mathbf{0}$ such that $A_{H'}\mathbf{c} = n\,\mathbf{c}$. Let H' be an irreducible sink of H. Recalling that each state $Z \in V(H')$ $\big(\subseteq V(H)\big)$ is a subset of $V(\mathcal{E})$, let $c_Z = |Z|$ denote the number of $\mathcal{E}$-states in Z and let $\mathbf{c}$ be the positive integer vector defined by $\mathbf{c} \triangleq [c_Z]_{Z\in V(H')}$. We now claim that

$$A_{H'}\mathbf{c} \;=\; n\,\mathbf{c}\;. \tag{3}$$

Consider a state $Z \in V(H')$. Since $\mathcal{E}$ has out-degree n, the number of edges in $\mathcal{E}$ emanating from the set of states $Z \subseteq V(\mathcal{E})$ is $n\,|Z|$. Now, let E_a denote the $\mathcal{E}$-edges labelled a which emanate from the $\mathcal{E}$-states in Z, and let Z_a denote the set of terminal $\mathcal{E}$-states of these edges. Note that the E_a, $a \in \Sigma$, induce a partition on the $\mathcal{E}$-edges emanating from Z. Clearly, if $Z_a \neq \emptyset$, there is an edge labelled a from Z to Z_a in H and, since H' is a sink, this edge is also contained in H'. We now claim that any $\mathcal{E}$-state $u \in Z_a$ is accessible in $\mathcal{E}$ by exactly one edge labelled a that begins in Z; otherwise if $Z = T_{\mathcal{E}}(\mathbf{w}, v)$, the word $\mathbf{w}a$ could be generated in $\mathcal{E}$ by two distinct paths which start at v and terminate at u, contradicting the losslessness of $\mathcal{E}$. Hence, $|E_a| = |Z_a|$ and, so, the entry of $A_{H'}\mathbf{c}$ corresponding to the H-state Z satisfies

$$\begin{aligned}
(A_{H'}\mathbf{c})_Z \;&=\; \sum_{Y\in V(H')} (A_{H'})_{Z,Y}\,c_Y \;=\; \sum_{Y\in V(H')} (A_{H'})_{Z,Y}\,|Y| \\
&=\; \sum_{a\in\Sigma} |Z_a| \;=\; \sum_{a\in\Sigma} |E_a| \;=\; n\,|Z| \;=\; n\,c_Z\;,
\end{aligned}$$

thus proving (3).

(c) Construct an (A_G, n)-approximate eigenvector from $\mathbf{c}$. As G and H' comply with the conditions of Lemma 3, each follower set of a state in H' is contained in a follower set of some state in G. Let $\mathbf{x} = [x_u]_{u\in V(G)}$ be the nonnegative integer vector defined by

$$x_u \;=\; \max\left\{\, c_Z \,|\, Z \in V(H') \text{ and } \mathcal{F}_{H'}(Z) \subseteq \mathcal{F}_G(u) \,\right\}\,, \quad u \in V(G)\,, \tag{4}$$

and denote by $Z(u)$ some particular H'-state Z for which the maximum is attained in (4). In case there is no state $Z \in V(H')$ such that $\mathcal{F}_{H'}(Z) \subseteq \mathcal{F}_G(u)$, define $x_u = 0$ and $Z(u) = \emptyset$. We claim that $\mathbf{x}$ is an (A_G, n)-approximate eigenvector. First, since $V(H')$ is nonempty, we have $\mathbf{x} \not\equiv \mathbf{0}$. Now, let u be a state in G; if $x_u = 0$ then, trivially, $(A_G\mathbf{x})_u \geq n\,x_u$ and, so, we can assume that $x_u \neq 0$. Let $Z_a(u)$ be the terminal state in H' for an edge labelled a emanating from the H'-state $Z(u)$. Since $\mathcal{F}_{H'}\big(Z(u)\big) \subseteq \mathcal{F}_G(u)$, there exists an edge labelled a in G

from u which terminates at some G-state u_a; and, since G and H' are both deterministic, we have $\mathcal{F}_{H'}\big(Z_a(u)\big) \subseteq \mathcal{F}_G(u_a)$. Furthermore, by (4), $x_{u_a} \geq c_{Z_a(u)}$ and, so, letting $\Sigma(Z(u))$ denote the set of labels of edges in H' outgoing from $Z(u)$, we have

$$(A_G\mathbf{x})_u \;\geq \sum_{a\in\Sigma(Z(u))} x_{u_a} \;\geq \sum_{a\in\Sigma(Z(u))} c_{Z_a(u)} \;=\; (A_{H'}\mathbf{c})_{Z(u)} \;\stackrel{(3)}{=}\; n\, c_{Z(u)} \;=\; n\, x_u \;.$$

Hence, $A_G\mathbf{x} \geq n\mathbf{x}$.

Finally, let U be a subset of $V(G)$ such that for any two distinct states $u, u' \in U$, $\mathcal{F}_G^\infty(u) \cap \mathcal{F}_G^\infty(u') = \emptyset$. It remains to show that $|V(\mathcal{E})| \geq \sum_{u\in U} x_u$. Since $\mathcal{F}_{H'}\big(Z(u)\big) \subseteq \mathcal{F}_G(u)$ we have $\mathcal{F}_{H'}^\infty\big(Z(u)\big) \subseteq \mathcal{F}_G^\infty(u)$ and

$$\mathcal{F}_{H'}^\infty\big(Z(u)\big) \cap \mathcal{F}_{H'}^\infty\big(Z(u')\big) \;\subseteq\; \mathcal{F}_G^\infty(u) \cap \mathcal{F}_G^\infty(u') = \emptyset \;, \quad u \neq u' \;.$$

This implies that the subsets $Z(u)$ and $Z(u')$ of $V(\mathcal{E})$ are disjoint whenever $u \neq u'$ and, therefore

$$|V(\mathcal{E})| \;\geq\; |\cup_{u\in U} Z(u)| \;=\; \sum_{u\in U} |Z(u)| \;=\; \sum_{u\in U} c_{Z(u)} \;=\; \sum_{u\in U} x_u \;.$$

Thus, we have found an approximate eigenvector $\mathbf{x}$ which satisfies (2) whenever U satisfies the conditions of the theorem. Now maximize over all such subsets U and minimize over all (A_G, n)-approximate eigenvectors. This completes the proof of Theorem 3. □

Corollary 1. *Let S, G, n and $\mathcal{E}$ be as in Theorem 3. Then,*

$$|V(\mathcal{E})| \;\geq \min_{[y_u]_u\in\mathcal{X}(A_G,n)} \; \max_{u\in V(G)} y_u \;. \tag{5}$$

Proof. Simply take U to be singleton sets in Theorem 3. □

A graph G is *separable* if it is deterministic and the semi-infinite follower sets $\mathcal{F}_G^\infty(u), u \in V(G)$, are pairwise disjoint. In particular, if all the edges of G are labelled distinctly, then G is separable. A constrained system S is separable if there exists a (deterministic) separable presentation G of S; we remark that this is stronger than assuming that S has a (not necessarily deterministic) presentation in which the semi-infinite follower sets are pairwise disjoint. Note that if S is separable, then, by Proposition 3, the minimum number $M(S,n)$ of states in an (S,n)-encoder is the minimum of $M(S_i,n)$ over a finite set of irreducible separable systems S_i. By Proposition 2 (part (ii)), if S is irreducible and separable, then the Shannon cover G_S is a separable presentation of S. In fact, if S is irreducible and separable, then the only (deterministic) irreducible separable presentation of S is G_S.

Corollary 2. *Let S, G, n and $\mathcal{E}$ be as in Theorem 3. Assume also that G is separable (so, $G = G_S$). Then,*

$$|V(\mathcal{E})| \geq \min_{[y_u]_u \in \mathcal{X}(A_G,n)} \sum_{u \in V(G)} y_u \ . \tag{6}$$

Proof. In this case, $U = V(G)$ satisfies the condition of Theorem 3. □

As noted before, for a given constrained system S and a positive integer n, the preceding results are completely vacuous if $c(S) < \log n$ i.e., in this situation, there are no (S,n)-encoders and, equivalently, no (A_G,n)-approximate eigenvectors (Proposition 1, part (ii)). Conversely, if $c(S) \geq \log n$, then there always exists an (S,n)-encoder: one can be constructed using the state splitting algorithm on some presentation G of S [1] (see also [7, §8.6] and the tutorial [20]). The state splitting algorithm starts with an (A_G,n)-approximate eigenvector $\mathbf{x}$ and produces an encoder with $\sum_{u \in V(G)} x_u$ states. It does this by splitting states of G. In many situations, the states of the encoder can be merged to produce an encoder with a smaller number of states. However, the previous results provide lower bounds on the number of states in an encoder and therefore bounds on how much merging can be done; of course, these bounds hold not only for encoders constructed by the state splitting algorithm, but for *any* encoder at all.

Ashley, in [4][5, Lemma 7], showed that for a given irreducible constrained system S and given designed rate (not greater than the capacity of the constrained system), the largest component of some approximate eigenvector is at most exponential in the number of states of the Shannon cover G_S. Thus, for fixed designed rate, there is always an encoder (constructed by the state splitting algorithm) whose number of states is at most exponential in the number of states of G_S. In the following we show that, in the worst case, one cannot do much better.

Example 1. The following is a modification of an example of Ashley [4], presenting a sequence of irreducible constrained systems S_k with the property that every $(S_k,2)$-encoder must be exponentially large in the number of states of the Shannon cover of S_k. Each S_k

is an irreducible constrained system over $\Sigma = \{a, b, c\}$ presented by the $(2k)$-state graph G_k depicted in Figure 1.

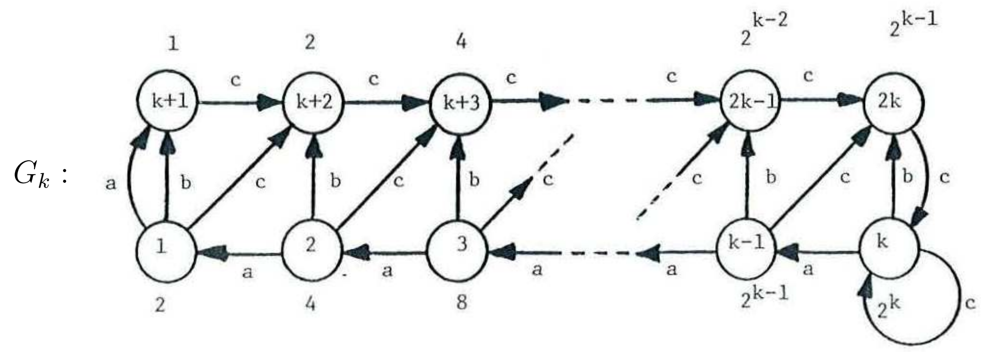


Figure 1: Constrained systems whose encoders have exponential number of states.

It can be readily verified that G_k is the Shannon cover of S_k and that $\lambda(A_{G_k}) = 2$; hence, $c(S_k) = 1$. Therefore, by Remark 1, the $(A_{G_k}, 2)$-approximate eigenvectors are the positive integer multiples of the smallest (componentwise) positive integer eigenvector $\mathbf{x}$ associated with the eigenvalue 2. The components of $\mathbf{x}$ are written next to each state in Figure 1 above. Note that the maximum component of $\mathbf{x}$ is 2^k; therefore, by Corollary 1, the number of states in any $(S_k, 2)$-encoder is at least 2^k i.e., exponential in the number, $2k$, of states in G_k. •

Corollary 2 applies to the Shannon cover; note, however, that in Theorem 3 and Corollary 1, we did not impose any restrictions on the choice of G, except that it be an irreducible deterministic presentation of S. We claim that in those two results, the bounds for any such G will give the same bounds as for the Shannon cover G_S. To see this, first recall (from the discussion after the proof of Proposition 2) the map $\Phi_G : V(G) \to V(G_S)$ which maps each state to its follower set. For any approximate eigenvector $\mathbf{x} \in \mathcal{X}(A_G, n)$, the vector $\mathbf{y}$, indexed by $V(G_S)$ and defined by

$$y_v = \max_{u \in V(G) \,:\, \Phi_G(u) = v} x_u \ , \tag{7}$$

is an (A_{G_S}, n)-approximate eigenvector; this follows from the fact that u and $\Phi_G(u)$ have the same number of outgoing edges with the same (distinct) labels. Also, if $U \subseteq V(G)$ satisfies the disjointness condition of Theorem 3, then the follower sets of states in U are distinct, and so Φ_G is 1-1 over U; also $\Phi_G(U)$ satisfies the disjointness condition of Theorem 3. Moreover, any $\mathbf{y} \in \mathcal{X}(A_{G_S}, n)$ is obtained from some $\mathbf{x} \in \mathcal{X}(A_G, n)$ via (7): simply "lift" $\mathbf{y}$ by "copying" its components; and any $U' \subseteq V(G_S)$ that satisfies the disjointness condition of Theorem 3 is the Φ_G-image of some $U \subseteq V(G)$ satisfying the disjointness condition. Putting this all together now, the reader can easily check that we get the same lower bound for G as for G_S. So, we may as well use the Shannon cover presentation, and typically this is how constrained systems are described. We recall that in case S is presented by a non-deterministic graph, we can find a deterministic presentation of S in the same way we constructed H out of $\mathcal{E}$ in part (a) of the proof of Theorem 3.

5. Realizing lower bounds

In some cases, the lower bounds above are tight i.e., they can be realized by an encoder. In particular, the state splitting algorithm shows that in the separable case, the lower bound of Corollary 2 can be achieved (see the discussion after Corollary 2). We formally record this as:

Proposition 4. *If S is an irreducible separable constrained system, then the bound (6) in Corollary 2 is tight i.e., there is an (S, n)-encoder $\mathcal{E}$ such that*

$$|V(\mathcal{E})| = \min_{[y_u]_u \in \mathcal{X}(A_G, n)} \sum_{u \in V(G)} y_u \ ,$$

where $G = G_S$.

Now, we turn to the situation which is the extreme opposite of the separable case. A graph G is called *linearly ordered* if the follower sets $\mathcal{F}_G(u)$, $u \in V(G)$, are linearly ordered by inclusion. Note that, for decent graphs, this is the same thing as requiring that the semi-infinite follower sets $\mathcal{F}_G^\infty(u)$, $u \in V(G)$, be linearly ordered by inclusion. A constrained system S is linearly ordered if it is presented by a linearly ordered graph G. Observe that, in contrast to the separable case, we did not require here that G be deterministic. However, by applying the determinizing construction, it can be readily verified that a constrained system is linearly ordered if and only if it is presented by a deterministic linearly ordered graph. By Proposition 2 (part (ii)), if an irreducible constrained system S is linearly ordered, then G_S is linearly ordered.

Proposition 5. *If S is an irreducible linearly ordered constrained system and $c(S) = \log n$, then the bound (5) in Corollary 1 is tight i.e., there is an (S, n)-encoder $\mathcal{E}$ such that*

$$|V(\mathcal{E})| = \min_{[y_u]_u \in \mathcal{X}(A,n)} \max_{u \in V(G)} y_u \ ,$$

where G is an irreducible deterministic linearly ordered presentation of S.

Note that since G is irreducible and $c(S) = \log n$ then, by Remark 1, there is a smallest (componentwise) element $\mathbf{x} = [x_u]_u$ of $\mathcal{X}(A_G, n)$, and so the proposition asserts that there is an (S, n)-encoder $\mathcal{E}$ such that

$$|V(\mathcal{E})| = \max_{u \in V(G)} x_u \ .$$

Proof of Proposition 5. By the remarks at the end of the previous section, we may assume that $G = G_S$. Write $V(G) = \{1, 2, \ldots, k\}$. We may reorder the states of G so that $\mathcal{F}_G(1) \subseteq \mathcal{F}_G(2) \subseteq \ldots \subseteq \mathcal{F}_G(k)$. Since $G = G_S$, the follower sets are all distinct and so $\mathcal{F}_G(i) \subseteq \mathcal{F}_G(j) \iff i \leq j$. We will find a graph H with the same states as G such that

(i) H presents S;

(ii) H is lossless;

and —

(iii) A_H has maximal eigenvalue (spectral radius) n and corresponding nonnegative eigenvector $\mathbf{d} = [d_i]_i$ defined by $d_i = x_i - x_{i-1}$ (set $x_0 = 0$).

Then we can apply the state splitting algorithm to the graph H with the eigenvector $\mathbf{d}$. This yields an (S, n)-encoder with $\sum_i d_i$ states. But $\sum_i d_i = x_k$, the maximal component of $\mathbf{x}$.

We define H as follows. Let $V(H) = V(G)$ (with the same ordering of states); endow H with an edge from i to j labelled a if and only if

(a) There is an edge in G from i to some $j' \geq j$ labelled a;

and —

(b) For all $j'' \geq j$, there is *no* edge in G labelled a from $i-1$ to j''.

To verify property (i), we first show that $S \subseteq S(H)$. Let $\mathbf{w}$ be a word in S, and let $u_{\mathbf{w}}$ be the smallest integer u such that $\mathbf{w} \in \mathcal{F}_G(u)$. We will show that $\mathbf{w} \in \mathcal{F}_H(u_{\mathbf{w}})$ and therefore $S \subseteq S(H)$. We prove this by induction on the length of $\mathbf{w}$. This certainly holds for the empty word. Now write $\mathbf{w} = a\mathbf{b}$ where a is a symbol in Σ ($\mathbf{b}$ may be the empty word). By the inductive hypothesis, $\mathbf{b} \in \mathcal{F}_H(u_{\mathbf{b}})$. Now, $\mathbf{w}$ is the labelling of some path in G that begins at $u_{\mathbf{w}}$. The first edge of this path is labelled a and goes from $u_{\mathbf{w}}$ to some state v. By the definition of $u_{\mathbf{b}}$, we get $v \geq u_{\mathbf{b}}$; therefore, by construction of H, there is an edge in H from $u_{\mathbf{w}}$ to $u_{\mathbf{b}}$ labelled a, unless there is an edge labelled a in G from $u_{\mathbf{w}} - 1$ to some $v'' \geq u_{\mathbf{b}}$; but if there were such an edge, then, since $\mathbf{b} \in \mathcal{F}_G(u_{\mathbf{b}}) \subseteq \mathcal{F}_G(v'')$, we would have $\mathbf{w} = a\mathbf{b} \in \mathcal{F}_G(u_{\mathbf{w}} - 1)$, contrary to the definition of $u_{\mathbf{w}}$. Thus, there is an edge in H from $u_{\mathbf{w}}$ to $u_{\mathbf{b}}$ labelled a and, therefore, there is a path in H labelled $\mathbf{w}$ which begins at $u_{\mathbf{w}}$ i.e., $\mathbf{w} \in \mathcal{F}_H(u_{\mathbf{w}})$ as desired.

Conversely, we show that for all states u we have $\mathcal{F}_H(u) \subseteq \mathcal{F}_G(u)$ and, hence, $S(H) \subseteq S$. We prove this inductively on the length of $\mathbf{w} \in \mathcal{F}_H(u)$; again, write $\mathbf{w} = a\mathbf{b}$ where a is a symbol in Σ. Thus, there is an edge in H from u to some state v labelled a, and $\mathbf{b} \in \mathcal{F}_H(v)$. By construction of H, there is an edge in G from u to some state $v' \geq v$ labelled a and, by the inductive hypothesis, $\mathbf{b} \in \mathcal{F}_G(v) \subseteq \mathcal{F}_G(v')$. Hence, there is a path in G labelled $\mathbf{w} = a\mathbf{b}$ that begins at u i.e., $\mathbf{w} \in \mathcal{F}_G(u)$. Thus, $S = S(H)$, proving property (i).

We now show property (ii) i.e., that H is lossless. In fact we will prove a stronger property, namely, that H is "backwards deterministic": for each state $u \in V(H)$, incoming edges to u are labelled distinctly. To see this, suppose that there are distinct edges e, e' in H, both with the same terminal state t and the same label a. Let i, i' be the initial states of e, e'. By construction of H we have $i \neq i'$, so we may assume $i < i'$. Also, by construction of H, there is an edge labelled a in G from i to some state j where $j \geq t$. Now since $i \leq i' - 1$, $\mathcal{F}_G(i) \subseteq \mathcal{F}_G(i' - 1)$. Hence, there is an edge in G from $i' - 1$ to some state j'' labelled a. Furthermore, since G is deterministic, we must have $\mathcal{F}_G(j) \subseteq \mathcal{F}_G(j'')$, implying that $j'' \geq j \geq t$. However, this contradicts the existence of the edge e' in H.

As for property (iii), we first claim that

$$(A_H)_{i,j} = \sum_{j' \geq j} \Big((A_G)_{i,j'} - (A_G)_{i-1,j'} \Big) , \quad 1 \leq i, j \leq k , \tag{8}$$

where $(A_G)_{0,j'} \equiv 0$. To show this, it is sufficient to verify that for $2 \le i \le k$ and $1 \le j \le k$, if there is an edge labelled a from state $i-1$ to state j' for some $j' \ge j$ in G, then there must be an edge labelled a from state i to state j'' for some $j'' \ge j' \ge j$ in G. This follows from the fact that

$$\mathcal{F}_G(i-1) \subseteq \mathcal{F}_G(i) \Rightarrow \mathcal{F}_G(j') \subseteq \mathcal{F}_G(j'') .$$

Now, let Q be the $k \times k$ matrix whose nonzero entries are $Q_{i+1,i} = -1$ and $Q_{i,i} = 1$:

$$Q = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} .$$

By (8), one easily verifies that $A_H = QA_GQ^{-1}$. So, A_H, being similar to A_G, has the same maximal eigenvalue, n, as A_G, and $\mathbf{d} = Q\mathbf{x}$ is a corresponding eigenvector. Furthermore, since G is irreducible, n is a simple eigenvalue of A_G (Theorem 2, part (ii)), and therefore for A_H. Hence, by Theorem 1, $\mathbf{d}$ is either nonnegative or nonpositive. However, the first nonzero entry of $\mathbf{d} = Q\mathbf{x}$ is positive. $\square$

We remark that the encoder constructed in Proposition 5 is "backwards deterministic".

6. Coding at rates approaching capacity

So far, we have considered encoders which can encode arbitrary sequences at some prespecified rate. Now, let's consider families of encoders which encode arbitrary sequences into the words of a constrained system S at higher and higher rates which approach $c(S)$. Proposition 6 below presents the simple and well-known result, that it is possible to approach capacity so that the encoders have only one state i.e., the encoders are really block codes.

A graph G is called *primitive* if, for some m, $(A_G)^m$ has strictly positive entries (in particular, G is irreducible). Let $\mathbf{l}$ and $\mathbf{r}$ be left (row) and right (column) eigenvectors, scaled so that $\mathbf{l} \cdot \mathbf{r} = 1$, for A_G associated with $\lambda = \lambda(A_G)$. For any primitive graph we have [23, Theorem 1.2],

$$\lim_{m\to\infty} \frac{(A_G)^m}{\lambda^m} = P_G \ , \tag{9}$$

where $P_G = \mathbf{r}{\cdot}\mathbf{l}$. Note that P_G has spectral radius 1 with associated right and left eigenvectors $\mathbf{r}$ and $\mathbf{l}$. Furthermore, by Theorem 2 (part (ii)), these eigenvectors are unique, up to scaling, for both A_G and P_G.

Lemma 4. (see [23, Ch. 1]). *Let G be a graph.*

(i) There exists an integer d such that each irreducible component of G^d is primitive.

(ii) For the above d, let $G_1, G_2, \ldots, G_k$ denote the irreducible components of G^d. Then the graphs $G_1^s, G_2^s, \ldots, G_k^s$ are the irreducible components of G^{ds} for all positive integers s.

Proposition 6. *Let S be a constrained system with $c(S) = \log \lambda$. There exist sequences $\{p_m\}$, $\{q_m\}$ and corresponding one-state $(S^{q_m}, 2^{p_m})$-encoders such that*

$$\lim_{m\to\infty} \frac{p_m}{q_m} = \log\,\lambda \ . \tag{10}$$

Proof. Let G be a deterministic presentation of S and let d be as in Lemma 4. Let $S' \subseteq S^d$ be a constrained system presented by an irreducible component G' of G^d with $\lambda(A_{G'}) = \lambda(A_{G^d})$ i.e., $c(S') = c(S^d)$ (such a component always exists: it is easy to verify that, for any graph H, $\lambda(A_H) = \max_{H'} \lambda(A_{H'})$, where H' ranges over the irreducible components of H). Set $q_m = dm$, $1 \le m < \infty$, and let u_m be a state u in G' such that the number of cycles in G' of length m through u is maximal. This number is equal to $\alpha_m = \max_{u\in V(G')}\big((A_{G'})^m\big)_{u,u}$ and, by (9) and Theorem 2 (parts (i) and (ii)) we have $\lim_{m\to\infty}(\alpha_m/\lambda^{q_m}) > 0$. We now construct a block code for S^{q_m} by mapping binary words of length $p_m = \lfloor \log\,\alpha_m \rfloor$ into words in S of length q_m obtained by reading the labelling on the cycles of length m in G' starting at u_m. Clearly, the rate $p_m : q_m$ satisfies (10). $\square$

So, it is not necessary to increase the number of states in order to approach capacity. However, usually, the block lengths p_m and q_m will grow "quickly" in this construction. One way to slow the growth of the block lengths would be to require that p_m/q_m approach capacity very "fast" — in the sense that the ratio p_m/q_m be close to $c(S)$ while keeping p_m

and q_m as small as possible. If the rate approaches capacity very fast, can we still hope for encoders whose numbers of states do not grow? For the typical constrained system, the answer is no. We now elaborate on this.

Write $c(S) = \log \lambda$. We divide capacities into three cases.

The first case is that λ be a rational power of 2 i.e., $c(S)$ is a rational number, s/t. Then the state splitting algorithm provides an $(S^t, 2^s)$-encoder i.e., a rate $s : t$ encoder. So, in this case, we can encode at rate equal to capacity.

The second case is that λ equals $k^{s/t}$, for some positive integer k which is not a power of 2. Then it is impossible to encode at rate equal to capacity. However, we will show that in this case, there exists an integer $N < \infty$ such that whenever $p/q < c(S)$ and t divides q, then there is an $(S^q, 2^p)$-encoder with N states; thus, whenever p_m/q_m approaches capacity from below and t divides q_m, then there exist $(S^{q_m}, 2^{p_m})$-encoders all with the same number of states. So, in this case, in spite of how fast we approach capacity, we can guarantee encoders with a (uniformly) fixed number of states.

Finally, the third case is what remains: λ is not a rational power of a positive integer. In this case, the typical case, if we approach capacity fast enough, we will show that the number of encoder states must grow without bound. In particular, if we approach capacity with the continued fraction approximants (which are the best approximants) to $\log \lambda$, then the number of encoder states tends to infinity. Of course, as in Proposition 6, it is still possible to approach capacity with one-state encoders.

We now make the foregoing discussion precise.

Theorem 4. *Let S be a constrained system with $c(S) = \log \lambda$.*

(i) If $\lambda = 2^{s/t}$, then there is an $(S^t, 2^s)$-encoder.

(ii) If $\lambda = k^{s/t}$ for some positive integer k, then there exists an integer N such that for any two integers p, q, where $\frac{p}{q} \leq \log \lambda$ and t divides q, there is an $(S^q, 2^p)$-encoder with at most N states.

(iii) If λ is not *a rational power of an integer and, in addition,*

$$\lim_{m\to\infty} \left(\frac{p_m}{q_m} - \log\,\lambda \right) \cdot q_m = 0 \;, \tag{11}$$

then for any sequence of $(S^{q_m}, 2^{p_m})$-encoders $\mathcal{E}_m$,

$$\lim_{m\to\infty} |V(\mathcal{E}_m)| = \infty \;.$$

It is well known that the continued fraction approximants to any number β satisfy $\left|\frac{p_m}{q_m} - \beta\right| < \frac{\alpha}{q_m^2}$ for some absolute constant α, and thus the continued fraction approximants for $\log\,\lambda$ satisfy (11). So, in case (iii) the fastest approach to capacity necessarily forces the number of states to grow without bound. On the other hand, in case (ii), from the continued fraction approximations to $\log k$, we can find sequences that satisfy (11) and for which encoders can be realized with a uniformly bounded number of states.

Proof of Theorem 4. *Cases (i) and (ii):* Let G be a deterministic presentation of S. Since $\lambda = k^{s/t}$, the matrix $(A_G)^t$ has an integer largest eigenvalue, namely $\lambda^t = k^s$, and an associated integer nonnegative eigenvector $\mathbf{x} = [x_u]_{u\in V(G)}$. Clearly, we also have $(A_G)^{tm}\mathbf{x} = \lambda^{tm}\mathbf{x}$ for any positive integer m. Since $\mathbf{x}$ is an eigenvector for $A_{G^{tm}}$ associated with the eigenvalue $\lambda^{tm} = k^{sm}$, we can apply the state splitting algorithm on $\mathbf{x}$, yielding an (S^{tm}, k^{sm})-encoder $\mathcal{E}_m$ of at most $N = \sum_{u\in V(G)} x_u$ states. Now let p and q be integers such that $p/q \le \log\,\lambda$ and $q = tm$. Delete edges (possibly none) from $\mathcal{E}_m$ to have out-degree $2^p \le \lambda^q = \left(k^{s/t}\right)^{tm} = k^{sm}$ at each state. We thus obtain an $(S^q, 2^p)$-encoder with at most N states.

Case (iii): The idea is to show that, for sequences $\{p_m\}$ and $\{q_m\}$ satisfying (11), the $\left((A_G)^{q_m}, 2^{p_m}\right)$-approximate eigenvectors approach an (A_G, λ)-eigenvector which must contain an irrational entry. Therefore, the largest components in such approximate eigenvectors tend to infinity. The result then follows from Corollary 1.

Let G be a deterministic presentation of S. We first reduce to the case where G is primitive. Let d be as in Lemma 4, let $G_1, G_2, \ldots, G_k$ be the irreducible components of G^d and let $S_i = S(G_i)$ (i.e., S_i has a primitive deterministic presentation G_i). We show that if the theorem holds for each S_i, then it holds for S. For this, first observe that for any $(S^s, 2^r)$-encoder $\mathcal{E}$, the graph $\mathcal{E}^d$ is an $(S^{ds}, 2^{dr})$-encoder (with the same number of states as

$\mathcal{E}$). Now, let $\mathcal{E}'$ be any irreducible sink in $\mathcal{E}^d$. Then $S(\mathcal{E}') \subseteq S^{ds} = S(G^{ds})$ is an irreducible constrained system and, so, by Lemma 2 and Lemma 4 (part (ii)), for some $j = j(\mathcal{E}')$ we have $S(\mathcal{E}') \subseteq S_j^s$, i.e., $\mathcal{E}'$ is an $(S_j^s, 2^{dr})$-encoder (note that $|V(\mathcal{E})| \geq |V(\mathcal{E}')|$).

Now, let p_m, q_m satisfy (11) and $\mathcal{E}_m$ be a sequence of $(S^{q_m}, 2^{p_m})$-encoders; then there are $(S_{j_m}^{q_m}, 2^{dp_m})$-encoders $\mathcal{E}'_m$, $j_m = j(\mathcal{E}'_m)$, such that $|V(\mathcal{E}_m)| \geq |V(\mathcal{E}'_m)|$. Since $\lim_{m\to\infty} \frac{p_m}{q_m} = \log\lambda$, we must have $\lim_{m\to\infty} c(S_{j_m}) = c(S^d) = \log\lambda^d$. However, the set of capacities $\{c(S_i)\}_{i=1}^k$ is finite; therefore, $c(S_{j_m}) = c(S^d) = \log\lambda^d$ for sufficiently large m. Now, $\lim_{m\to\infty}\left(\frac{dp_m}{q_m} - \log\lambda^d\right) \cdot q_m = 0$; hence, assuming the theorem holds for each S_i, $i = 1, 2, \ldots, k$, we have that $|V(\mathcal{E}'_m)|$ grows without bound, and therefore so must $|V(\mathcal{E}_m)|$.

So, we may assume that S has a deterministic primitive presentation G and therefore (9) applies. Let $\{p_m\}$ and $\{q_m\}$ be integer sequences satisfying (11). By Corollary 1, the existence of $(S^{q_m}, 2^{p_m})$-encoders $\mathcal{E}_m$ implies the existence of $\left((A_G)^{q_m}, 2^{p_m}\right)$-approximate eigenvectors $\mathbf{x}_m = [x_{m;u}]_{u\in V(G)}$ such that

$$|V(\mathcal{E}_m)| \geq \max_{u\in V(G)} x_{m;u} ,$$

and we have

$$\frac{(A_G)^{q_m}}{\lambda^{q_m}}\mathbf{x}_m \geq \frac{2^{p_m}}{\lambda^{q_m}}\mathbf{x}_m . \tag{12}$$

We will now show that $\max_{u\in V(G)} x_{m;u}$ grows unboundedly when $m \to \infty$, and therefore so does $|V(\mathcal{E}_m)|$. Assume, to the contrary, that there is a subsequence m_i such that $\max_{u\in V(G)} x_{m_i;u}$ is bounded. Passing to a further subsequence, we may assume that $\mathbf{x}_{m_i}$ converges to a nonnegative integer vector $\mathbf{x} \not\equiv \mathbf{0}$, i.e., $\mathbf{x}_{m_i} = \mathbf{x}$ for sufficiently large i. By (9), (11) and (12), we obtain

$$P_G\mathbf{x} \geq \mathbf{x} .$$

But P_G is an irreducible matrix with spectral radius 1, and therefore, by Theorem 2 (part (iii)) $\mathbf{x}$ is an eigenvector for P_G associated with eigenvalue 1. Hence, by part (ii) of Theorem 2, $\mathbf{x}$ is a multiple of $\mathbf{r}$ and thus

$$A_G\mathbf{x} = \lambda\mathbf{x} .$$

However, the left-hand side of this equation has integer entries, and so λ is an integer (i.e., a rational root of an integer monic polynomial), contrary to the assumption of case (iii). $\square$

7. Stronger coding properties

As we mentioned earlier, the losslessness condition in the definition of encoder is the minimal requirement that one would want to make in order to guarantee decodability. However, that condition is not too practical for decoding. A stronger and more reasonable condition is the following. A graph G is called *lossless of finite order* if there is an integer N such that any two paths of length N with the same initial state and labelling must have the same initial edge. The smallest N for which this holds is called the *order* of G and denoted $\mathcal{O}(G)$. In case G is not lossless of finite order we define $\mathcal{O}(G) = \infty$. Note that any deterministic graph has order 1. So, a graph which is lossless of finite order can be viewed as "deterministic with bounded delay".

The notion of losslessness of finite order was introduced by Huffman [17] (see also Even [9]). In symbolic dynamics, losslessness of finite order is called "right-closing", and this is closely related to (but not the same as) the notion of "bounded deciphering delay" in automata theory.

First, we give a lower bound on the order of an encoder. By virtue of Proposition 3, it suffices to consider encoders for irreducible constrained systems only.

Theorem 5. *Let S, G, n and $\mathcal{E}$ be as in Theorem 3. Then,*

$$\mathcal{O}(\mathcal{E}) \;\geq\; 1 + \log_n \left(\min_{[y_u]_u \in \mathcal{X}(A_G, n)} \; \max_{u \in V(G)} y_u \right) \;.$$

Proof. The theorem trivially holds if $\mathcal{O}(\mathcal{E}) = \infty$; therefore, we assume that $\mathcal{E}$ is lossless of finite order. Let $\mathbf{x} = [x_u]_{u \in V(G)}$ be as in the proof of Theorem 3. We recall that by the way $\mathbf{x}$ was constructed, each nonzero component of $\mathbf{x}$ is a size of some subset $Z = T_{\mathcal{E}}(\mathbf{w}, v)$ of states in $\mathcal{E}$ which are accessible from $v \in V(\mathcal{E})$ by paths labelled $\mathbf{w}$.

Let $T_{\mathcal{E}}(\mathbf{w}, v)$ be such a subset whose size equals the largest component of $\mathbf{x}$, and let $\ell = \ell(\mathbf{w})$ (length of $\mathbf{w}$). Since the out-degree of $\mathcal{E}$ is n, the number of paths of length ℓ emanating from v in $\mathcal{E}$ is n^ℓ and, therefore, we must have

$$n^\ell \;\geq\; |T_{\mathcal{E}}(\mathbf{w}, v)| \;=\; \max_{u \in V(G)} x_u \;,$$

implying

$$\ell \geq \log_n \left(\min_{[y_u]_u \in \mathcal{X}(A_G, n)} \max_{u \in V(G)} y_u \right) .$$

Therefore, when $\mathcal{O}(\mathcal{E}) \geq \ell + 1$, we are done.

Assume now that $\ell > \mathcal{O}(\mathcal{E}) - 1$. Since $\mathcal{E}$ is lossless of finite order, the first $\ell - \mathcal{O}(\mathcal{E}) + 1$ edges of any path in $\mathcal{E}$ labelled $\mathbf{w}$ are uniquely determined once we know the initial state v. It thus follows that the paths from v to $T_\mathcal{E}(\mathbf{w}, v)$ labelled $\mathbf{w}$ may differ only in their last $\mathcal{O}(\mathcal{E}) - 1$ edges. Hence, we can have at most $n^{\mathcal{O}(\mathcal{E})-1}$ such paths. Recalling that the number of such paths is $|T_\mathcal{E}(\mathbf{w}, v)|$, we have

$$n^{\mathcal{O}(\mathcal{E})-1} \geq |T_\mathcal{E}(\mathbf{w}, v)| = \max_{u \in V(G)} x_u \geq \min_{[y_u]_u \in \mathcal{X}(A_G, n)} \max_{u \in V(G)} y_u . \qquad \square$$

A special case of the bound of Theorem 5 appears in the work of Franaszek [14].

Example 2. Consider the irreducible constrained system S over $\Sigma = \{a, b, c, d\}$ presented by the graph G of Figure 2.

G :

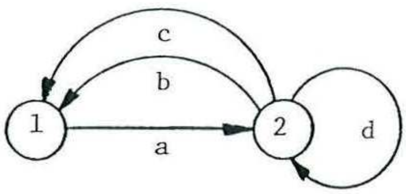


Figure 2: Example for the bound of Theorem 5.

The largest eigenvalue of A_G is 2 and the associated eigenvector is $\mathbf{x} = [1\ 2]^T$. Since S is separable, the lower bound 3 on the number of states of any $(S, 2)$-encoder (Corollary 2) can be attained by the state splitting algorithm (Proposition 4) which, in this case, provides an encoder $\mathcal{E}_1$ of order 2 (Figure 3), thereby achieving the bound of Theorem 5.

$\mathcal{E}_1$:

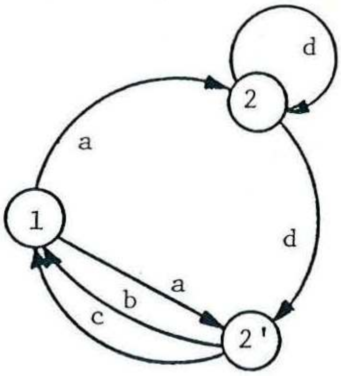


Figure 3: Encoder of order 2 for the system presented by Figure 2.

On the other hand, there are $(S, 2)$-encoders, such as the one in Figure 4, which are not lossless of finite order. •

$\mathcal{E}_2$:

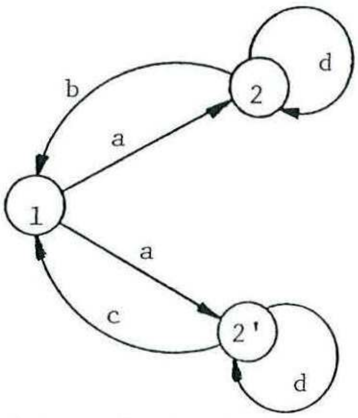


Figure 4: Encoder of infinite order for the system presented by Figure 2.

The following example shows that the bound in Theorem 5 is not tight.

Example 3. Let S be the irreducible constrained system of capacity $\log 3$ over $\Sigma =$

$\{a, b, c, d, e, f\}$, presented by the graph G of Figure 5.

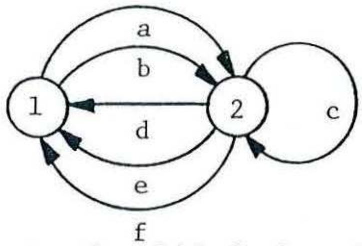
$G:$ 

Figure 5: Constrained system for which the bound of Theorem 5 is not tight.

In Appendix A we show that, for $n = 3$, the lower bound of Theorem 5 is 2, whereas any $(S, 3)$-encoder $\mathcal{E}$ is shown to have $\mathcal{O}(\mathcal{E}) \geq 3$. •

Ashley [5, Lemma 7] showed that, for a given irreducible graph G and a positive integer $n \leq \lambda(A_G)$, there is always an (A_G, n)-approximate eigenvector whose maximal component is at most $n^{2|V(G)|}$. Therefore the lower bound of Theorem 5 is at most $1 + 2\,|V(G)|$. On the other hand, there is a linear *upper* bound on the smallest order of an (S, n)-encoder. This can be deduced from Theorem 1 in [5].

Even losslessness of finite order is usually not adequate for practical decoding purposes: error propagation can cause catastrophic error events. What we usually require is that the encoder have a *sliding block decoder* i.e., the inverse of the encoder is given by a machine that decodes sequences by examining the contents of a sliding window and does not rely on external state information.

More precisely, let $\mathcal{E}$ be an (S, n) encoder; for each state $v \in V(\mathcal{E})$ we assign distinct *tags* to the outgoing edges from v, each tag is an integer between 1 and n. The encoding process can be regarded as reading the labelling of a path defined by some initial state v and an input sequence of tags. Given a tagged encoder $\mathcal{E}$, we say that $\mathcal{E}$ is *sliding-block decodable* if there exist positive integers ℓ and $r \leq \ell$ such that for any ℓ-symbol word $\mathbf{w} = w_1 w_2 \ldots w_{r-1} w_r w_{r+1} \ldots w_l$ generated by $\mathcal{E}$ (starting from any state), the r-th tag is uniquely defined. The smallest ℓ for which the above holds is called the *minimum-window length* of $\mathcal{E}$ and denoted by $\mathcal{L}(\mathcal{E})$. If $\mathcal{E}$ is not sliding-block decodable, define $\mathcal{L}(\mathcal{E}) = \infty$.

A state v in a graph G is called *degenerate* if v has no incoming edges in G. Dropping degenerate states from G does not change the constrained system in a significant way, e.g., it does not lower the capacity. So, we may as well assume that encoders do not have degenerate states.

Proposition 7. *Let $\mathcal{E}$ be a tagged encoder with no degenerate states. Then,*

$$\mathcal{L}(\mathcal{E}) \geq \mathcal{O}(\mathcal{E}) . \tag{13}$$

Proof. The proposition trivially holds if $\mathcal{L}(\mathcal{E}) = \infty$; therefore we can assume that $\mathcal{E}$ is sliding-block decodable. Let $\mathbf{w} = w_1 w_2 \ldots w_\ell$ be a word generated by $\mathcal{E}$ starting at state v, where $\ell = \mathcal{L}(\mathcal{E})$. Since v is nondegenerate, there exist sequences $\mathbf{w}'$ of any length which can be generated by $\mathcal{E}$ while terminating in v. Prefixing such sufficiently long $\mathbf{w}'$ to $\mathbf{w}$, we can recover the edge tag at w_1 out of $\mathbf{w}'\mathbf{w}$; this tag identifies the edge outgoing from v. $\square$

Clearly, if an encoder $\mathcal{E}$ does contain degenerate states, we can delete states from $\mathcal{E}$ consecutively, until we end up with an encoder $\mathcal{E}'$ with no degenerate states; in this case Proposition 7 implies $\mathcal{L}(\mathcal{E}) \geq \mathcal{L}(\mathcal{E}') \geq \mathcal{O}(\mathcal{E}')$.

The relationships between the aforementioned various (labelling) conditions are summarized as follows.

Proposition 8. *Let $\mathcal{E}$ be a tagged encoder with no degenerate states.*

There exists a sliding block decoder for $\mathcal{E}$ $\Rightarrow$
$\mathcal{E}$ is lossless of finite order $\Rightarrow$
$\mathcal{E}$ is lossless.

Also,

$\mathcal{E}$ is deterministic $\Rightarrow$ $\mathcal{E}$ is lossless of finite order.

We point out that it is possible to determine whether a given encoder is lossless or lossless of finite order by efficient algorithms [9][17]. We remark that there is an effective algorithm to determine whether a given encoder is sliding-block decodable.

The state splitting algorithm, applied to a graph which is lossless of finite order (in particular, deterministic) always produces encoders which are lossless of finite order. So, in

Proposition 4, the realization of the lower bound is always lossless of finite order (in fact, that realization has also a sliding block decoder). However, this is not the case for Proposition 5. The construction in Proposition 5 uses the state splitting algorithm and an additional construction which potentially destroys the losslessness of finite order (although, recall that the encoders obtained are "backwards deterministic"). Indeed, in the next example we present a linearly ordered constrained system S with $c(S) = \log n$ and yet any (S, n)-encoder which realizes the bound

$$|V(\mathcal{E})| = \min_{\mathbf{x} \in \mathcal{X}(A_G, n)} \max_{u \in V(G)} x_u$$

cannot be lossless of finite order (and therefore cannot have a sliding block decoder).

Example 4. Consider the linearly ordered irreducible constrained system S of capacity $\log 3$ over $\Sigma = \{a, b, c, d\}$ presented by the graph G of Figure 6.

G :

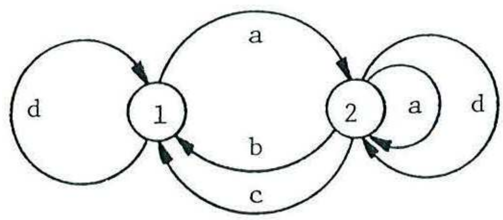


Figure 6: Constrained system for which any encoder of minimum number of states must have infinite order.

In Appendix B we show that for this S and $n = 3$, the lower bound of Proposition 5 above is 2, and this bound is attained by a unique 2-state $(S, 3)$-encoder; however, this encoder is not lossless of finite order. •

We remark that sliding block decoders are not always necessary to avoid non-catastrophic error propagation; there are non-catastrophic constructions in [19] that do not have sliding block decoders — in fact, there are situations where sliding block decoders are not possible to construct, but non-catastrophic codes are still possible.

Finally, we mention that Kamabe [18] also studied the problem of finding the minimum-window length of sliding block decoders — especially for run-length limited constrained systems.

8. Stronger lower bounds

In this section, we improve the lower bound of Theorem 3. The new bound involves the following notations.

Let S be a constrained system presented by a deterministic graph G. For a state $u \in V(G)$ and a word $\mathbf{w} \in \mathcal{F}_G(u)$, let $T_G(\mathbf{w}, u)$ be the terminal state of the path in G that begins in u and generates $\mathbf{w}$. For a word $\mathbf{w} \notin \mathcal{F}_G(u)$ define $T_G(\mathbf{w}, u) = \emptyset$.

Let n be a positive integer and $\mathbf{x} = [x_u]_{u \in V(G)}$ be an (A_G, n)-approximate eigenvector. For a word $\mathbf{w}$ and a subset $U \subseteq V(G)$, let $I_{\mathbf{x}}(\mathbf{w}, U)$ denote a state $u \in U$ such that $x_{T_G(\mathbf{w},u)}$ is maximal (for the case where $T_G(\mathbf{w}, u) = \emptyset$ we define $x_\emptyset \triangleq 0$).

A list of words Γ over a finite alphabet is called *complete* if any word over the alphabet either has a prefix in Γ or is a prefix of a word in Γ. The lengths of words in such word lists satisfy the following known inequality:

Lemma 5. (The reverse Kraft-MacMillan inequality). *Let Γ be a finite complete list over an alphabet of n elements. Then,*

$$\sum_{\mathbf{w} \in \Gamma} n^{-\ell(\mathbf{w})} \geq 1 ,$$

where $\ell(\mathbf{w})$ stands for the length of $\mathbf{w}$.

Proof. Let $\ell_{\max} = \max_{\mathbf{w} \in \Gamma} \ell(\mathbf{w})$. Let $R(\mathbf{w})$ be the set of all n-ary words of length $\ell_{\max}$ which have $\mathbf{w}$ as a prefix. Since Γ is a complete list, $\cup_{\mathbf{w} \in \Gamma} R(\mathbf{w})$ must exhaust all n-ary words of length $\ell_{\max}$. Hence,

$$\sum_{\mathbf{w} \in \Gamma} n^{\ell_{\max} - \ell(\mathbf{w})} = \sum_{\mathbf{w} \in \Gamma} |R(\mathbf{w})| \geq n^{\ell_{\max}} . \qquad \square$$

For $U \subseteq V(G)$, a list C of words is *U-complete* if every word in $\mathcal{F}_G(U)$ either has a prefix in C or is a prefix of a word in C. Let $\mathcal{C}(U)$ denote the set of all finite U-complete lists. For example, the list $C_m(U)$, of all words of length m that can be generated from states of U, belongs to $\mathcal{C}(U)$.

Given an integer n, an (A_G, n)-approximate eigenvector $\mathbf{x}$, a subset U of $V(G)$ and a list C of words, define $\mu_{\mathbf{x};n}(U, C)$ by

$$\mu_{\mathbf{x};n}(U, C) \;\triangleq\; \sum_{u \in U} x_u \;-\; \sum_{\mathbf{w} \in C} n^{-\ell(\mathbf{w})} \sum_{u \in U - \{I_{\mathbf{x}}(\mathbf{w}, U)\}} x_{T_G(\mathbf{w}, u)} \;. \tag{14}$$

Theorem 6. *Let S, G, n and $\mathcal{E}$ be as in Theorem 3. Then*

$$|V(\mathcal{E})| \;\geq\; \min_{\mathbf{y} \in \mathcal{X}(A_G, n)} \; \max_{U \subseteq V(G)} \; \sup_{C \in \mathcal{C}(U)} \; \mu_{\mathbf{y};n}(U, C) \;.$$

In particular, for all $U \subseteq V(G)$ and m,

$$|V(\mathcal{E})| \geq \min_{\mathbf{y} \in \mathcal{X}(A_G, n)} \mu_{\mathbf{y};n}(U, C_m(U)) \;.$$

Before we prove Theorem 6, we observe that it includes Theorem 3. To see this, note that for sufficiently large m, if $u, u' \in V(G)$ and $\mathcal{F}_G^{\infty}(u) \cap \mathcal{F}_G^{\infty}(u') = \emptyset$, then $\mathcal{F}_G(u) \cap \mathcal{F}_G(u') \cap C_m\Big(V(G)\Big) = \emptyset$; thus,

$$|V(\mathcal{E})| \geq \min_{\mathbf{y} \in \mathcal{X}(A_G, n)} \; \max_{U} \mu_{\mathbf{y};n}(U, C_m(U)) = \min_{\mathbf{y} = [y_u]_u \in \mathcal{X}(A_G, n)} \; \max_{U} \sum_{u \in U} y_u \;, \tag{15}$$

where the maximum is taken over all $U \subseteq V(G)$ satisfying the disjointness condition of Theorem 3 (note that for such sets U and for any sufficiently long word $\mathbf{w}$, we have $x_{T_G(\mathbf{w}, u)} \neq 0$ in (14) only if $u = I_{\mathbf{x}}(\mathbf{w}, U)$). However, the bound (15) is the conclusion of Theorem 3.

Proof of Theorem 6. Exactly as in the proof of Theorem 3, given an (S, n)-encoder $\mathcal{E}$, we construct the determinizing graph H, an irreducible sink H' of H, and an approximate eigenvector $\mathbf{x} \in \mathcal{X}(A_G, n)$ that reflects $\mathcal{E}$. Recall that a state $Z \in V(H)$ can be regarded as a subset of $V(\mathcal{E})$; it is clear from context whether we view Z as a state of H or as a subset of $V(\mathcal{E})$. Recall also, from the construction of $\mathbf{x}$, that for each $u \in V(G)$, $x_u = \max\{|Z| : Z \in V(H') \,\text{and}\, \mathcal{F}_{H'}(Z) \subseteq \mathcal{F}_G(u)\}$, and we set $Z(u) \in V(H')$ to be a state which achieves this maximum (when $x_u = 0$, we set $Z(u) = \emptyset$).

For $U \subseteq V(G)$ let $Z(U) = \cup_{u \in V(G)} Z(u) \subseteq V(\mathcal{E})$. We will prove that

$$\text{For any } U \subseteq V(G) \text{ and } C \in \mathcal{C}(U), \quad |Z(U)| \geq \mu_{\mathbf{x};n}(U, C) \;; \tag{16}$$

and since $|V(\mathcal{E})| \geq |Z(U)|$, this yields Theorem 6 immediately.

For $v \in V(\mathcal{E})$ let U_v denote the set of G-states $u \in U$ such that $v \in Z(u)$. We have

$$|Z(U)| \;=\; \sum_{u \in U} |Z(u)| \;-\; \sum_{v \in Z(U)} \Big(|U_v| - 1\Big) \;, \tag{17}$$

where the last term in (17) rectifies the over-counting of the states of $Z(U)$ by the sum $\sum_{u \in U} |Z(u)|$.

Therefore, in order to prove (16), it suffices to show that

$$\sum_{v \in Z(U)} \Big(|U_v| - 1\Big) \;\le\; \sum_{\mathbf{w} \in C} n^{-\ell(\mathbf{w})} \sum_{u \in U - \{I_{\mathbf{x}}(\mathbf{w}, U)\}} x_{T_G(\mathbf{w}, u)} \;. \tag{18}$$

For a word $\mathbf{w}$ and a state $v \in V(\mathcal{E})$, let $T_{\mathcal{E}}(\mathbf{w}, v)$ denote (as before) the set of $\mathcal{E}$-states that are accessible from v by paths labelled $\mathbf{w}$. As argued in the proof of Theorem 3, due to the losslessness of $\mathcal{E}$, no two such paths can terminate at the same state of $\mathcal{E}$ and, therefore, $|T_{\mathcal{E}}(\mathbf{w}, v)|$ is also the number of paths in $\mathcal{E}$ which start at v and generate $\mathbf{w}$. Our proof is based on the inequality

$$\sum_{\mathbf{w} \in C} \frac{|T_{\mathcal{E}}(\mathbf{w}, v)|}{n^{\ell(\mathbf{w})}} \ge 1 \quad \text{for every } v \in Z(U) \text{ and } C \in \mathcal{C}(U) \;. \tag{19}$$

To prove (19), tag the edges of $\mathcal{E}$, that is, for each state in $V(\mathcal{E})$, mark the outgoing edges by distinct integers between 1 and n (as in Section 7). Now, for a given state $v \in Z(U)$, let $\Gamma(v)$ be the list of words over $\{1, 2, \ldots, n\}$ obtained by reading the tags of the paths in $\mathcal{E}$ (if any) which start at v and which generate words in C. We now show that $\Gamma(v)$ is a complete list over $\{1, 2, \ldots, n\}$ which, by Lemma 5, implies (19).

Let $\mathbf{z}$ be a word over $\{1, 2, \ldots, n\}$; reading the labels on the path γ in $\mathcal{E}$ which starts at v and defined by the tag sequence $\mathbf{z}$, we obtain a word $\mathbf{w}$ over Σ. Now,

$$\mathbf{w} \in \mathcal{F}_{\mathcal{E}}(v) \subseteq \mathcal{F}_{\mathcal{E}}\Big(Z(U)\Big) \subseteq \mathcal{F}_G(U) \;.$$

Recalling that C is U-complete, assume first that $\mathbf{w}$ has a prefix $\mathbf{w}'$ in C; in this case, read the first $\ell(\mathbf{w}')$ tags of γ; this yields a prefix $\mathbf{z}'$ of $\mathbf{z}$ which is contained in $\Gamma(v)$.

Now, if $\mathbf{w}$ does not have a prefix in C, continue the path γ until we obtain an (overall) path δ which generates a word in C (since C is finite and U-complete, this will always occur). Clearly $\mathbf{z}$ is a prefix of the word $\mathbf{z}'$ obtained by reading the tags on the edges of δ. Therefore, $\Gamma(v)$ is a complete list.

Having shown (19), by (18) we have

$$\begin{aligned}\sum_{v\in Z(U)}\Big(|U_v|-1\Big) &\le \sum_{v\in Z(U)}\Big(|U_v|-1\Big)\sum_{\mathbf{w}\in C}\frac{|T_{\mathcal{E}}(\mathbf{w},v)|}{n^{\ell(\mathbf{w})}}\\ &= \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{v\in Z(U)}\Big(|U_v|-1\Big)\,|T_{\mathcal{E}}(\mathbf{w},v)|\ . \end{aligned}\tag{20}$$

Let $U_v(\mathbf{w})$ be the set U_v with one G-state deleted so that $I_{\mathbf{x}}(\mathbf{w},U)\notin U_v(\mathbf{w})$ (note that $I_{\mathbf{x}}(\mathbf{w},U)$ need not be contained in U_v). By (20) we have

$$\begin{aligned}\sum_{v\in Z(U)}\Big(|U_v|-1\Big) &\le \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{v\in Z(U)}|U_v(\mathbf{w})|\cdot|T_{\mathcal{E}}(\mathbf{w},v)|\\ &= \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{v\in Z(U)}\sum_{u\in U_v(\mathbf{w})}|T_{\mathcal{E}}(\mathbf{w},v)|\\ &= \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{u\in U}\sum_{v\in Z(U)\text{ s.t. }u\in U_v(\mathbf{w})}|T_{\mathcal{E}}(\mathbf{w},v)|\\ &= \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{u\in U-\{I_{\mathbf{x}}(\mathbf{w},U)\}}\sum_{v\in Z(U)\text{ s.t. }u\in U_v(\mathbf{w})}|T_{\mathcal{E}}(\mathbf{w},v)| &&(21)\\ &\le \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{u\in U-\{I_{\mathbf{x}}(\mathbf{w},U)\}}\sum_{v\in Z(U)\text{ s.t. }u\in U_v}|T_{\mathcal{E}}(\mathbf{w},v)|\\ &= \sum_{\mathbf{w}\in C} n^{-\ell(\mathbf{w})}\sum_{u\in U-\{I_{\mathbf{x}}(\mathbf{w},U)\}}\sum_{v\in Z(u)}|T_{\mathcal{E}}(\mathbf{w},v)|\ , &&(22)\end{aligned}$$

where (21) follows from the fact that for $u=I_{\mathbf{x}}(\mathbf{w},U)$, the inner sum is over an empty set of vertices v.

Now, let $u\in V(G)$ be such that $Z(u)\neq\emptyset$ i.e., $Z(u)$ is a (proper) state of the irreducible sink H' of H. By construction of H', for every such u, the set $Z_{\mathbf{w}}(u)\triangleq\cup_{v\in Z(u)}T_{\mathcal{E}}(\mathbf{w},v)$, if nonempty, is a state of H' which is accessible from $Z(u)$ by a path (in H') labelled $\mathbf{w}$. Recalling that $\mathcal{F}_{H'}\Big(Z(u)\Big)\subseteq\mathcal{F}_G(u)$ and that H' is deterministic, we thus have

$$\mathcal{F}_{H'}\Big(Z_{\mathbf{w}}(u)\Big)\subseteq\mathcal{F}_G\Big(T_G(\mathbf{w},u)\Big)\ .$$

Hence, by construction of $\mathbf{x}$ we obtain

$$|Z_{\mathbf{w}}(u)|\ \le\ x_{T_G(\mathbf{w},u)}\ . \tag{23}$$

On the other hand, since $\mathcal{E}$ is lossless, we also have

$$|Z_{\mathbf{w}}(u)|\ =\ \Big|\cup_{v\in Z(u)}T_{\mathcal{E}}(\mathbf{w},v)\Big|\ =\ \sum_{v\in Z(u)}|T_{\mathcal{E}}(\mathbf{w},v)|\ . \tag{24}$$

Plugging (23) and (24) into (22) we finally obtain

$$\sum_{v \in Z(U)} \Big(|U_v| - 1 \Big) \le \sum_{\mathbf{w} \in C} n^{-\ell(\mathbf{w})} \sum_{u \in U - \{I_{\mathbf{x}}(\mathbf{w},U)\}} x_{T_G(\mathbf{w},u)} ,$$

as claimed. $\square$

Example 5. Let S be the irreducible constrained system over $\Sigma = \{a,b,c,d,e,f,g,h\}$ presented by the graph G of Figure 7.

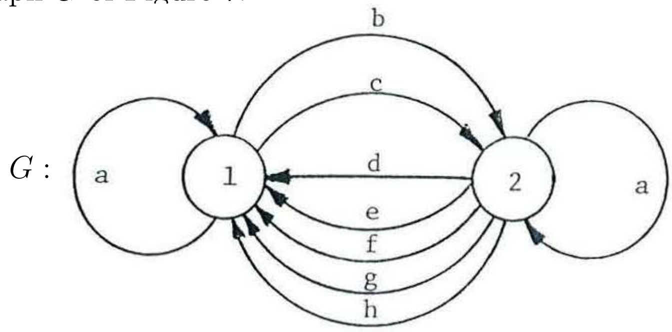


Figure 7: Example for the bound of Theorem 6.

Note that, except for the two edges labelled a, all edges are labelled distinctly. The adjacency matrix of G is given by

$$A_G = \begin{bmatrix} 1 & 2 \\ 5 & 1 \end{bmatrix} .$$

It is easy to verify that $c(S) \ge \log 4$. We claim that the number of states in a smallest $(S,4)$-encoder is 5. A straightforward computation shows that a vector $\mathbf{x} = (x_1, x_2)^T \not\equiv \mathbf{0}$ satisfies $A_G \mathbf{x} \ge 4\mathbf{x}$ if and only if

$$x_1 \ne 0 \quad \text{and} \quad \frac{3}{2} \le \frac{x_2}{x_1} \le \frac{5}{3} . \tag{25}$$

Thus, $\mathbf{x} \in \mathcal{X}(A_G, n)$ if and only if x_1, x_2 are positive integers and (25) holds. Set $U = \{1,2\}$; then, by Theorem 6 we have, for any $(S,4)$-encoder $\mathcal{E}$,

$$\begin{aligned} |V(\mathcal{E})| &\ge \min_{\mathbf{x} \in \mathcal{X}(A_G,4)} \mu_{\mathbf{x};4}(U, C_1(U)) \\ &= \min_{\mathbf{x} \in \mathcal{X}(A_G,4)} \left\{ x_1 + x_2 - \frac{\min(x_1,x_2)}{4} \right\} \\ &\ge \min_{\mathbf{x} \in \mathcal{X}(A_G,4)} \left\{ x_1 + x_2 - \frac{x_1}{4} \right\} \\ &= \min_{\mathbf{x} \in \mathcal{X}(A_G,4)} \left\{ \tfrac{3}{4} x_1 + x_2 \right\} . \end{aligned}$$

The minimization is subject to the constraints (25) and that x_1, x_2 be positive integers. Now, by (25), $x_1 \geq 2$, and therefore $x_2 \geq 3$. Thus, $\frac{3}{4}x_1 + x_2 \geq \frac{3}{2} + 3 > 4$. Therefore, the minimum number of states in an $(S, 4)$-encoder is at least 5. On the other hand, the vector $[2\ 3]^T$ is an $(A_G, 4)$-approximate eigenvector and, therefore, the state splitting algorithm yields such an encoder with five states. Thus, the minimum number of states in an $(S, 4)$-encoder is exactly 5 as desired. Note that Theorem 3 gives a lower bound of only three states. •

A widely used class of constrained systems is the class of run-length-limited (RLL) systems, defined as follows. The (d, k)-RLL system is the set of binary sequences such that the number r of 0's in between two nearest appearances of 1 satisfies $d \leq r \leq k$. The $(1, 7)$ and $(2, 7)$-RLL systems are commonly used in data storage applications.

Example 6. Let S be the $(1, 7)$-RLL system over $\Sigma = \{0, 1\}$, presented by the graph G depicted in Figure 8.

G :

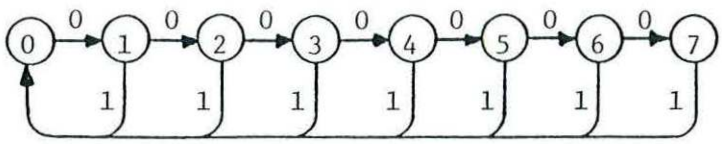


Figure 8: The $(1, 7)$*-RLL system.*

Since G is deterministic, the capacity of S is determined by the largest eigenvalue of A_G, and a computation yields $c(S) \approx .679$; so, it is possible to construct codes at rate $2 : 3$, and this is the standard coding rate for this constrained system. These codes are represented by $(S^3, 4)$-encoders. In Appendix C, we show that a lower bound on the number of states in an $(S^3, 4)$-encoder is 4. A 4-state encoder was recently constructed by Weathers [26] (the encoder used in practice is due to Adler, Hassner and Moussouris [3] and has five states; see also [16]); thus Weathers' encoder has the smallest possible number of encoder states. •

The $(2, 7)$-RLL-system has capacity $\approx .517$. So, a rate $1 : 2$ code is possible. The $(S^2, 2)$-encoder that is used in practice is due to Franaszek [12] and has six states (see also [8][16]).

Howell [16] showed how to construct a 5-state encoder. One can check, in a manner similar to that for the $(1,7)$ constraint, that 5 is a lower bound on the number of encoder states, and thus Howell's encoder has the smallest possible number of encoder states.

Finally, we deduce a corollary of Theorem 6 in case $c(S) = \log n$.

Given a deterministic graph G, an approximate eigenvector $\mathbf{x} = [x_u]_{u \in V(G)}$ and a word $\mathbf{w} \in S(G)$, let $T_{\mathbf{x}}(\mathbf{w})$ denote the state $T_G\big(\mathbf{w}, I_{\mathbf{x}}(\mathbf{w}, V(G))\big)$, i.e., a terminal state u of a path in G that generates $\mathbf{w}$ such that x_u is maximal.

Corollary 3. *Let S, G, n and $\mathcal{E}$ be as in Theorem 3 and, in addition, assume that $c(S) = \log n$. Then for every finite prefix-free list P of words in S,*

$$|V(\mathcal{E})| \geq \min_{\mathbf{x}=[x_u]_u \in \mathcal{X}(A_G,n)} \sum_{\mathbf{w} \in P} \frac{x_{T_{\mathbf{x}}(\mathbf{w})}}{n^{\ell(\mathbf{w})}} \; . \tag{26}$$

Note that since $c(S) = \log n$, by Remark 1, the set $\mathcal{X}(A_G, n)$ consists of all positive integer multiples of some positive eigenvector $\mathbf{x}$; so, the minimum is achieved at this particular $\mathbf{x}$.

Proof of Corollary 3. We may assume that P is $V(G)$-complete in addition to being prefix-free; otherwise extend P to a $V(G)$-complete list, thus making the right-hand side of (26) at least as big.

By Theorem 6 we have

$$|V(\mathcal{E})| \;\geq\; \min_{\mathbf{x}=[x_u]_u \in \mathcal{X}(A_G,n)} \left\{ \sum_{u \in V(G)} x_u \;-\; \sum_{\mathbf{w} \in P} n^{-\ell(\mathbf{w})} \sum_{u \neq I_{\mathbf{x}}(\mathbf{w},V(G))} x_{T_G(\mathbf{w},u)} \right\} . \tag{27}$$

Recalling that $\mathbf{x}$ is a true eigenvector and that P is both prefix-free and $V(G)$-complete, we have, for every $u \in V(G)$,

$$x_u \;=\; \sum_{\mathbf{w} \in P} \frac{x_{T_G(\mathbf{w},u)}}{n^{\ell(\mathbf{w})}} \; ,$$

thus yielding,

$$\sum_{u \in V(G)} x_u \;=\; \sum_{\mathbf{w} \in P} n^{-\ell(\mathbf{w})} \sum_{u \in V(G)} x_{T_G(\mathbf{w},u)} \; . \tag{28}$$

Plugging (28) into (27) we obtain

$$\begin{aligned} |V(\mathcal{E})| \;&\geq\; \min_{\mathbf{x} \in \mathcal{X}(A_G,n)} \sum_{\mathbf{w} \in P} \frac{x_{T_G(\mathbf{w}, I_{\mathbf{x}}(\mathbf{w},V(G)))}}{n^{\ell(\mathbf{w})}} \\ &=\; \min_{\mathbf{x} \in \mathcal{X}(A_G,n)} \sum_{\mathbf{w} \in P} \frac{x_{T_{\mathbf{x}}(\mathbf{w})}}{n^{\ell(\mathbf{w})}} \; . \qquad \square \end{aligned}$$

9. Computing the bounds

The bound (5) in Corollary 1 can be computed easily by an integer programming method adapted to this setting by P.A. Franaszek [13, Appendix] (see also [1, Appendix]). For the sake of completeness, we describe this algorithm here, along with proof of correctness and computational analysis.

The input to the following algorithm (Franaszek's algorithm) is a nonnegative matrix A, a positive integer n and a nonnegative integer vector $\mathbf{z}$. The output is a nonnegative integer vector $\mathbf{x}$, the properties of which are summarized in Proposition 9 (part (ii)) below.

> $\mathbf{y} \leftarrow \mathbf{z}$ **;**
> $\mathbf{x} \leftarrow \mathbf{0}$ **;**
> **while** $\mathbf{x} \neq \mathbf{y}$ **do**
> **begin**
> $\mathbf{x} \leftarrow \mathbf{y}$ **;**
> $\mathbf{y} \leftarrow \min\left\{ \left\lfloor \frac{1}{n} A\mathbf{x} \right\rfloor, \mathbf{x} \right\}$ **/*** $\lfloor\cdot\rfloor$ **stands for the floor function */**
>   **/* both** $\lfloor\cdot\rfloor$ **and** $\min\{\cdot,\cdot\}$ **are applied componentwise. */**
> **end ;**
> **output:** $\mathbf{x}$ **.**

For a nonnegative matrix A, a positive integer n and a nonnegative integer vector $\mathbf{z} = [z_u]_u$, let $\mathcal{X}_{\mathbf{z}}(A, n)$ denote the set of all elements $\mathbf{x} = [x_u]_u$ of $\mathcal{X}(A, n)$ that are dominated by $\mathbf{z}$ (i.e., $x_u \leq z_u$ for all u, or, in short, $\mathbf{x} \leq \mathbf{z}$).

Proposition 9. *Let A be a nonnegative matrix and n be a positive integer.*

(i) If $\mathbf{x}, \mathbf{x}' \in \mathcal{X}(A, n)$, then the vector defined by $[\max(x_u, x'_u)]_u$ belongs to $\mathcal{X}(A, n)$. Thus, for any nonnegative integer vector $\mathbf{z}$ there is a largest (componentwise) element of $\mathcal{X}_{\mathbf{z}}(A, n)$.

(ii) Franaszek's algorithm eventually halts for any input vector $\mathbf{z}$; and the output is either the zero vector (if $\mathcal{X}_{\mathbf{z}}(A, n) = \emptyset$) or the largest (componentwise) element of $\mathcal{X}_{\mathbf{z}}(A, n)$.

Proof. Part (i) is a straightforward computation. As for part (ii), let $\mathbf{x}$ be an element of $\mathcal{X}_{\mathbf{z}}(A, n)$ and let $\mathbf{y}_m$ denote the value of $\mathbf{y}$ at the beginning of the m-th iteration of the main loop, where $\mathbf{y}_1 = \mathbf{z}$. We show inductively on m that $\mathbf{x} \le \mathbf{y}_m$. Clearly, this holds for $m = 1$. Now, assuming that $\mathbf{x} \le \mathbf{y}_m$, we also have $\mathbf{x} \le \frac{1}{n}A\mathbf{x} \le \frac{1}{n}A\mathbf{y}_m$. Since $\mathbf{x}$ is an integer vector we obtain

$$\mathbf{x} \le \min\left\{ \left\lfloor \frac{1}{n}A\mathbf{y}_m \right\rfloor, \mathbf{y}_m \right\} = \mathbf{y}_{m+1} .$$

It remains to show that the algorithm halts and produces the required vector $\mathbf{x}$. Assume first that $\mathbf{y}_{m+1} \ne \mathbf{y}_m$. Recalling that $\mathbf{y}_{m+1}$ is dominated by $\mathbf{y}_m$, the sum of the entries of $\mathbf{y}_{m+1}$ in this case is strictly smaller than that of $\mathbf{y}_m$. Therefore, since all vectors involved are nonnegative integer, the algorithm must eventually halt with $\mathbf{y}_{m+1} = \mathbf{y}_m$. At this point we have $\mathbf{y}_m = \mathbf{y}_{m+1} \le \left\lfloor \frac{1}{n}A\mathbf{y}_m \right\rfloor$. Hence, either $\mathbf{y}_m \in \mathcal{X}_{\mathbf{z}}(A, n)$ or $\mathbf{y}_m = \mathbf{0}$. Furthermore, if $\mathcal{X}_{\mathbf{z}}(A, n) \ne \emptyset$, we must have $\mathbf{y}_m \not\equiv \mathbf{0}$, as $\mathbf{y}_m$ dominates any vector in $\mathcal{X}_{\mathbf{z}}(A, n)$. $\square$

By the proof of Proposition 9 it also follows that the number of iterations of the main loop of the algorithm is at most $1 + \sum_u z_u$.

In order to compute the bound of Corollary 1, we apply the above algorithm to the adjacency matrix A_G of the constrained system with various vectors $\mathbf{z} = [\beta\, \beta\, \ldots\, \beta]^T$. The value B of the bound equals the smallest β for which the above procedure yields a nonzero output $\mathbf{x}$ (by Proposition 1 (part (ii)), such β always exists if $n \le \lambda(A_G)$). Performing a binary search on β for the value of B, the number of integer operations thus totals to $O\Big(|V(G)|^2 \cdot B \log B\Big)$ (recall that there are cases though where B is exponential in $|V(G)|$). Although the above algorithm is polynomial in the *value* of its output B (and not in the size of its *representation*), we may still regard such an algorithm as efficient: since B, being a lower bound on $|V(\mathcal{E})|$ for any encoder $\mathcal{E}$, is also a lower bound on the size of the representation of $\mathcal{E}$.

As for the other lower bounds presented in this paper, it is still open whether there exist efficient algorithms for computing, or approximating, these bounds in general.

Appendix A

Here we show that the lower bound on the order of an encoder, given by Theorem 5, is not tight. For this we show that the constrained system S of Example 3 has no $(S, 3)$-encoder whose order equals the bound of Theorem 5. Let G be the presentation in Example 3. Then

$$A_G = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}.$$

Since the column sums of G are both 3, we have $\lambda(A_G) = 3$. Thus, by Proposition 1 (part (i)), $c(S) = \log 3$, and by Remark 1, the $(A_G, 3)$-approximate eigenvectors are the positive integer multiples of the smallest positive integer eigenvector, which, as is easily verified, is $[2\ 3]^T$. Thus, the lower bound of Theorem 5 on the order of any $(S, 3)$-encoder is $1 + \log_3(\max\{2, 3\}) = 2$.

We claim that there is no $(S, 3)$-encoder which is lossless of order 2. So, assume the contrary and let $\mathcal{E}$ be an $(S, 3)$-encoder which is lossless of order 2.

By a *monochromatic* state, we mean a state of $\mathcal{E}$ such that all (three) of its outgoing edges are labelled the same. We first show that there must be a monochromatic state with label (i.e., label of its outgoing edges) a, b or c.

For this, first observe that since $\mathcal{E}$ is not deterministic, there is a state $v \in V(\mathcal{E})$ that has two outgoing edges e_1, e_2 with the same label r. Since $\mathcal{E}$ is lossless, the terminal states t_1, t_2 of e_1, e_2 are distinct.

Case 1: $r = d, e,$ or f. Inspection of G reveals that the only possible labels for the edges outgoing from t_1, t_2 are a, b; if one of these labels appeared on an outgoing edge of both t_1 and t_2, then the order of $\mathcal{E}$ would be at least 3. Thus, t_1 (and t_2) are monochromatic with label a or b.

Case 2: $r = a, b,$ or c. There are a total of six edges outgoing from t_1, t_2. Inspection of G reveals that the only possible labels for these edges are c, d, e, f; again, if one of these labels appeared on an outgoing edge of both t_1 and t_2, then the order of $\mathcal{E}$ would be at least 3. Thus, either one of these states is monochromatic with label c or one of these states has two outgoing edges with the same label $d, e,$ or f, thereby reducing to case 1.

So, we now have a monochromatic state v with label $a, b,$ or c. Since $\mathcal{E}$ is lossless, the terminal states u_1, u_2, u_3 of the three outgoing edges from v are all distinct. The only possible labels of these outgoing edges are c, d, e, f. Now, just as before, if one of these labels appeared on an outgoing edge from more than one of u_1, u_2, u_3, then the order of $\mathcal{E}$ would be at least 3.

Thus each label can appear as an outgoing edge from at most one of u_1, u_2, u_3. Since there are four possible labels, two of these states, say u_1, u_2 must be monochromatic, and thus at least one of these states, say u_1, has all three of its outgoing edges having the same label — either d or e or f (but not c). Since $\mathcal{E}$ is lossless, the terminal states v_1, v_2, v_3 of these outgoing edges are all distinct. Now, again because $\mathcal{E}$ has order 2, there can be no label which appears as an outgoing edge of more than one of v_1, v_2, v_3. But there are only two possible labels: a and b. This leaves a state with no possible labels for its outgoing edges, a contradiction.

Appendix B

Here we give an example of a constrained system S and an integer n which meets the hypotheses of Proposition 5 (i.e., is irreducible, linearly ordered and has $c(S) = \log n$), and yet any (S, n)-encoder which realizes the bound in Corollary 1 cannot be lossless of finite order.

Let G be the graph in Example 4 and let $S = S(G)$. Then

$$A_G = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \quad ,$$

$c(S) = \log 3$, and the smallest positive integer eigenvector is $[1\ 2]^T$. Thus, the bound in Corollary 1 is 2. Since, as is readily verified, this system is linearly ordered, according to Proposition 5, there is a 2-state $(S, 3)$-encoder; indeed, the encoder $\mathcal{E}$ in Figure 9 is such an example.

$\mathcal{E}$:

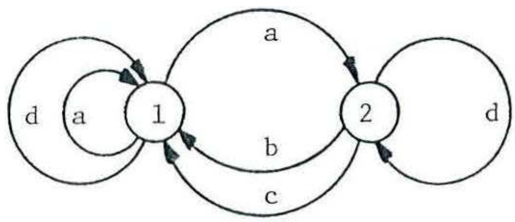


Figure 9: Encoder for the system of Example 4 (Figure 6).

Note that this graph is not lossless of finite order. We now show that this is the only 2-state $(S, 3)$-encoder, and therefore there cannot be a 2-state $(S, 3)$-encoder which is lossless of finite order.

So, let $\mathcal{E}$ be an $(S, 3)$-encoder with $V(\mathcal{E}) = \{1, 2\}$. Since $c(S(\mathcal{E})) = \log 3 = c(S)$ and $S(\mathcal{E}) \subseteq S$ and S is irreducible, we must have $S(\mathcal{E}) = S$. So, every word in S is actually generated by $\mathcal{E}$.

Now observe that since b and c can make only isolated appearances, they cannot be labels of self-loops; moreover, since b, c cannot follow one another they must appear as labels of edges with the same initial states and follower states; thus, we may assume that there are two edges from state 2 to state 1 labelled b, c. Now, since bdb is not a word in S, no edge from state 1 to state 2 can be labelled d; on the other hand, both bd and db appear in S; thus, there must be self-loops at both states labelled d. At this point, we have determined the subgraph of the 2-state encoder as shown in Figure 10.

$\mathcal{E}$:

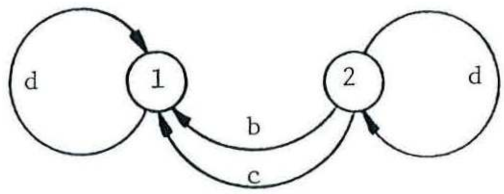


Figure 10: Partial encoder for the system of Example 4 (Figure 6).

Now, the outgoing edges (there are only three of them) from state 2 are all accounted for; Thus, there must be an outgoing edge from state 1 labelled a; since aa is in S, there must be a self-loop at state 1 labelled a, and since ab is allowed, there must be an edge from state 1 to state 2 labelled a. We have now completely reconstructed the encoder above, as desired.

Appendix C

Let S be the $(1, 7)$-RLL constrained system (Figure 8). Here we show that a lower bound on the number of states in any $(S^3, 4)$-encoder is 4, and thus the encoder due to Weathers [26] has the smallest number of states for any rate $2 : 3$ encoder for the $(1, 7)$-RLL system (see the discussion on RLL constrained systems in Section 8).

Note that the graph in Figure 8 is deterministic (in fact it is the Shannon cover) and its adjacency matrix is

$$A_G \quad = \quad \begin{bmatrix} & 1 & & & & & & \\ 1 & & 1 & & & & & \\ 1 & & & 1 & & & & \\ 1 & & & & 1 & & & \\ 1 & & & & & 1 & & \\ 1 & & & & & & 1 & \\ 1 & & & & & & & 1 \\ 1 & & & & & & & \end{bmatrix} .$$

We present S^3 by G^3; as G^3 has a large number of edges, we represent it in the following matrix form (the labels in the (u, v)-entry are the labels of the edges from u to v):

G^3 :

	0	1	2	3	4	5	6	7
0	001	010		000				
1	001, 101	010	100		000			
2	001, 101	010	100			000		
3	001, 101	010	100				000	
4	001, 101	010	100					000
5	001, 101	010	100					
6	101	010	100					
7	101		100					

.

Let $\mathcal{E}$ be an $(S^3, 4)$-encoder and let $U = \{0, 1\} \subseteq V(G^3)$. From Theorem 6 we obtain

$$|V(\mathcal{E})| \geq \min_{\mathbf{x} \in \mathcal{X}((A_G)^3, 4)} \max \Big\{ \max_{u \in V(G)} x_u \; ; \; \mu_{\mathbf{x};4}(U, C_1(U)) \Big\} .$$

As mentioned in Section 9, we can find $\min_{\mathbf{x} \in \mathcal{X}((A_G)^3,4)} \max_{u \in V(G)} x_u$ by a binary search that uses Franaszek's algorithm. In this case, one simply inputs the vectors $\mathbf{z} = [\beta \, \beta \, \ldots \, \beta]^T$ for $\beta = 1, 2, \ldots$ until the algorithm does not yield the zero vector; this first happens when $\beta = 3$. Thus,

$$\min_{\mathbf{x} \in \mathcal{X}((A_G)^3,4)} \max_{u \in V(G)} x_u \; = \; 3.$$

The output of Franaszek's algorithm, when $\beta = 3$, is the vector

$$\mathbf{x}^{(1)} = [2\,3\,3\,3\,2\,2\,2\,1]^T \; .$$

Recall from Proposition 9 that this is the largest (componentwise) $(A_{G^3}, 4)$-approximate eigenvector dominated (componentwise) by $[3\,3\,\ldots\,3]^T$. Also, by Proposition 9, to find the other approximate eigenvectors dominated by $[3\,3\,\ldots\,3]^T$, one can input into Franaszek's algorithm all of the vectors obtained from $\mathbf{x}^{(1)}$ by decreasing exactly one component by 1 — and then continuing iteratively. When one does this, one sees that the algorithm yields the zero vector except when the vector, obtained by decreasing the last component by one, is input to the algorithm; in that case, the input is already an approximate eigenvector:

$$\mathbf{x}^{(2)} = [2\,3\,3\,3\,2\,2\,2\,0]^T \; .$$

Thus, applying the same procedure to this vector (i.e., for each component, decrease the component by 1), one always gets the zero vector as the output of the algorithm. Hence, the only approximate eigenvectors whose maximal component equals 3 are $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. Hence, if $\mathbf{x} \in \mathcal{X}((A_G)^3, 4)$ and $\mathbf{x} \neq \mathbf{x}^{(1)}, \mathbf{x}^{(2)}$, then

$$\max\Bigl\{ \max_{u \in V(G)} x_u \; ; \; \mu_{\mathbf{x};4}(U, C_1(U)) \Bigr\} \;\geq\; \max_{u \in V(G)} x_u \;\geq\; 4 \; .$$

Now, inspection of G^3 reveals that the labels that are common to the outgoing edges of states 0 and 1 are 001, 010, and 000; the terminal states of these edges starting at state 0 (respectively state 1) are 0 (respectively, 0), 1 (respectively, 1), 3 (respectively, 4). Therefore,

$$\mu_{\mathbf{x};4}(U, C_1(U)) \;=\; x_0 + x_1 - \frac{x_0 + x_1 + \min\{x_3, x_4\}}{4} \; ,$$

and for both $\mathbf{x} = \mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ this quantity is 3.25. Hence, for any $(S^3, 4)$-encoder $\mathcal{E}$ we must have $|V(\mathcal{E})| \geq 3.25$ and, so, $|V(\mathcal{E})| \geq 4$, as desired. We have therefore shown that Weathers' code has the smallest number of states possible.

The analysis is completely similar for the $(2, 7)$-RLL constrained system. In this case, the rate is $1 : 2$ and the only $(A_{G^2}, 2)$-approximate eigenvectors with maximal component at most 4 are $[2\,3\,4\,4\,3\,3\,2\,1]^T$ and $[2\,3\,4\,4\,3\,3\,1\,1]^T$. An application of Theorem 6, with the choice of states $U = \{0, 2\}$ and the choice of U-complete list $C_1(U)$, yields a lower bound of 5. Thus, Howell's code [16] has the smallest number of states possible.

Acknowledgement

The first author would like to thank Joel Friedman for helpful discussions on these topics several years ago. We also thank Henk Hollmann for his comments on the original version of this paper.

References

[1] R. Adler, D. Coppersmith, M. Hassner, *Algorithms for sliding block codes — an application of symbolic dynamics to information theory, IEEE Trans. Inform. Theory,* IT-29 (1983), 5–22.

[2] R. Adler, J. Friedman, B. Kitchens, B. Marcus, *State splitting for variable-length graphs, IEEE Trans. Inform. Theory,* IT-32 (1986), 108–113.

[3] R. Adler, M. Hassner, J. Moussouris, *Method and apparatus for generating a noiseless sliding block code for a* $(1,7)$ *channel with rate 2/3,* US patent 4,413,251 (1982).

[4] J. Ashley, *On the Perron-Frobenius eigenvector for non-negative integral matrices whose largest eigenvalue is integral, Linear Algebra and its Applications,* 94 (1987), 103–108.

[5] J. Ashley, *A linear bound for sliding block decoder window size, IEEE Trans. Inform. Theory,* IT-34 (1988), 389–399.

[6] M.-P. Beal, *The method of poles: a coding method for constrained channels, IEEE Trans. Inform. Theory,* IT-36 (1990), 763–772.

[7] R.E. Blahut, *Digital Transmission of Information,* Addison-Wesley, Reading, Massachusetts, 1990.

[8] J. Eggenberger, P. Hodges, *Sequential encoding and decoding of variable length, fixed rate data codes,* US patent 4,115,768 (1978).

[9] S. Even, *On information lossless automata of finite order, IEEE Trans. Electron. Computers,* EC-14 (1965), 561–569.

[10] R. Fischer, *Sofic systems and graphs, Monats. fur Math.* 80 (1975), 179–186.

[11] R. Fischer, *Graphs and symbolic dynamics, Colloq. Math. Soc. Janos Bolyai, Topics in Information Theory,* 16 (1975), 229–243

[12] P.A. Franaszek, *Run-length-limited variable length coding with error propagation limitation,* US patent 3,689,899 (1972).

[13] P.A. Franaszek, *Construction of bounded delay codes for discrete channels, IBM J. Res. Dev.,* 26 (1982), 506–514.

[14] P.A. Franaszek, *Coding for constrained channels: a comparison of two approaches, IBM J. Res. Dev.,* 33 (1989), 602–607.

[15] C. Heegard, B. Marcus, P.H. Siegel, *Variable length state splitting with applications to average run-length constrained (ARC) codes,* preprint, 1989.

[16] T.D. Howell, *Statistical properties of selected recording codes, IBM J. Res. Dev.,* 32 (1989), 60–72.

[17] D.A. Huffman, *Canonical forms for information lossless finite-state machines, IRE Trans. Circuit Theory,* CT-6, Special Supplement (1959), 41–59.

[18] H. Kamabe, *Minimum scope for sliding block decoder mappings, IEEE Trans. Inform. Theory,* IT-35 (1989), 1335-1340.

[19] R. Karabed, B. Marcus, *Sliding-block coding for input-restricted channels, IEEE Trans. Inform. Theory,* IT-34 (1988), 2–26.

[20] A. Khayrallah, D. Neuhoff, *Subshift models and finite-state codes for input-constrained noiseless channels: a tutorial,* preprint, 1990.

[21] B. Marcus, P.H. Siegel, *Constrained codes for partial response channels, Proc. Beijing Int'l Workshop Inform. Theory,* 1988, DI-1.1–1.4.

[22] H. Minc, *Nonnegative Matrices,* Wiley, New York, 1988.

[23] E. Seneta, *Non-negative Matrices and Markov Chains,* Springer, New York, 1973.

[24] C.E. Shannon, *The mathematical theory of communication, Bell Sys. Tech. J.,* 27 (1948), 379–423.

[25] R.S. Varga, *Matrix Iterative Analysis,* Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

[26] A. Weathers, Ph.D. Thesis (Ch. 6), Univ. Calif. San Diego, 1990.