ENCODING FOR INPUT-CONSTRAINED CHANNELS

GITIT SADEH
ENCODING FOR INPUT-CONSTRAINED CHANNELS

RESEARCH THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER SCIENCE

GITIT SADEH

SUBMITTED TO THE SENATE OF THE TECHNION — ISRAEL INSTITUTE OF
TECHNOLOGY
Tammuz, 5756
Haifa
June, 1996
THIS RESEARCH THESIS WAS SUPERVISED BY DR. RONNY ROTH UNDER THE AUSPICES OF THE COMPUTER SCIENCE DEPARTMENT

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Ronny Roth for his excellent guidance throughout all stages of research for this thesis; for the endless amount of time and the depth of thought at the stage of developing the ideas, and for the constant help, the fruitful advice, and the enlightening comments at the stage of writing, all of which were very useful in spite of the geographical distance.

I would also like to thank Professor Abraham Lempel for his help in completing this thesis.

Special thanks to my family and to Yaron for the great support and encouragement.

THE GENEROUS FINANCIAL HELP OF THE INTEL GRADUATE FELLOWSHIP IS GRATEFULLY ACKNOWLEDGED
## Contents

Abstract.................................................. 1

List of Symbols and Abbreviations...................... 2

1 Introduction............................................ 3

2 Background and Statement of Results.................. 6

  2.1 Graph Presentations.................................. 6
  2.1.1 Labeled Graphs.................................. 6
  2.1.2 Properties of labeled Graphs.................... 7
  2.1.3 Irreducibility of Graph Presentations.......... 8
  2.1.4 Reduced Graphs and the Shannon Cover.......... 8
  2.1.5 Two Graph Constructions......................... 9

  2.2 Capacity and Perron-Frobenius Theory............. 10

  2.3 Finite-State Encoders.............................. 11

  2.4 The State-Splitting Algorithm.................... 13

    2.4.1 State Splitting................................ 13
    2.4.2 Approximate Eigenvectors and Consistent Splitting 15
    2.4.3 Upper Bounds on the Anticipation of Encoders 16

  2.5 Lower Bounds on the Anticipation – Previous Results. 17

  2.6 Results of This Work............................... 18

    2.6.1 First Main Theorem – Theorem 4.1.............. 18
    2.6.2 Second Main Theorem – Theorem 6.1............. 19

  2.7 Some More Definitions and Properties............. 20

    2.7.1 Input Tagging of Graphs........................ 20
    2.7.2 Equivalence, Reduction, and Expansion of States. 20
    2.7.3 Expansion and State-Splitting.................. 22

3 Determinizing Graphs of Encoders.................... 25

  3.1 Construction of a Determinizing Graph $G_{\mathcal{E}}$............ 25
    3.1.1 States of $G_{\mathcal{E}} - V(G_{\mathcal{E}})$........ 25
    3.1.2 Edges of $G_{\mathcal{E}} - E(G_{\mathcal{E}})$........ 26
    3.1.3 Basic Properties of $G_{\mathcal{E}}$.............. 27
3.1.4 Perron Eigenvector of $G_E$ ................................................. 28
3.2 Splitting of $G_E$ ................................................................. 29
   3.2.1 Lemma 3.3 — Splitting of $G_E$ in $A$ Rounds ..................... 29
   3.2.2 Lemma 3.5 - Splitting of $G^k_E$ in One Round ................. 30
   3.2.3 Proof of Lemma 3.3 ....................................................... 31

4 First Lower Bound on the Anticipation ................................. 38
   4.1 Statement of the Result — Theorem 4.1 ............................ 38
   4.2 Proof of Theorem 4.1 ..................................................... 40
      4.2.1 Construction of $G$ — Step 1 .................................. 40
      4.2.2 Construction of $G$ — Step 2 ................................ 41
      4.2.3 Construction of $G$ — Step 3 ................................. 42
   4.3 Application of Theorem 4.1 — Examples ............................ 43
      4.3.1 Preliminaries ....................................................... 44
      4.3.2 $(1,7)$-RLL Constraint ..................................... 46
      4.3.3 $(2,18,2)$-RLL Constraint ................................. 47

5 Upper Bounds on the Number of States ................................. 52
   5.1 Introduction ............................................................... 52
   5.2 Example Where Splitting Requires Expansion of $G_S$ .......... 53
   5.3 Upper Bounds on the Number of States ............................ 58
      5.3.1 First Upper Bound — The Graph $G'$ ....................... 58
      5.3.2 Second Upper Bound — The Graph $G''$ ...................... 63

6 Second Lower Bound on the Anticipation .............................. 69
   6.1 Statement and Proof of Theorem 6.1 ............................... 69
   6.2 Application of Theorem 6.1 — Examples ........................... 71
      6.2.1 $(2,7)$-RLL Constraint .................................. 72
      6.2.2 $(1,7)$-RLL Constraint ................................ 74
      6.2.3 $(2,18,2)$-RLL Constraint ............................... 74

7 Conclusion ............................................................................. 76

References ............................................................................. 78
List of Figures

2.1 Graph presentation (Shannon cover) of the (1,7)-RLL constraint ........... 7
2.2 Encoder graph for the (2,7)-RLL constraint at rate 1:2 ...................... 12
2.3 Basic out-splitting of state u ............................................ 14
2.4 Equivalent states that are not strongly-equivalent .......................... 21
2.5 Operation of expansion .................................................... 23
2.6 Sequence of basic expansions .............................................. 23

3.1 Construction of state Z in GE ............................................. 26
3.2 Splitting of state Z in GE .................................................. 32

4.1 Shannon cover of \( S^3_{(1,7)} \) ................................................. 46
4.2 Shannon cover of \( S^5_{(2,18,2)} \) .............................................. 48
4.3 Possible outgoing pictures from state \( u^1 \) after the first round ............ 50

5.1 Graph \( G_S \) ............................................................... 54
5.2 Splitting of states \( A \) and \( B \) in the first round ............................... 55
5.3 Splitting options for state \( C \) in the first round ............................... 56
5.4 Possible outgoing pictures of descendants after the first round ............. 56
5.5 Outgoing picture after the first round of the splitting of \( G \) ................. 57
5.6 Second round of the splitting of \( G \) ....................................... 57
5.7 Relationship between descendants in \( G^{(i)} \) and \( G^{(i)} \) .................... 60

6.1 Shannon cover of \( S^2_{(4,7)} \) ............................................... 72
6.2 Outgoing pictures of \( v_2 \) and \( v_3 \) according to \( \bar{y} \) .................... 73
Abstract

Input-constrained channels are models for describing the read-write requirements of secondary storage systems, such as magnetic disks or optical devices. Examples for such requirements are the widely used $(d, k)$-run-length-limited (RLL) constraints, where each run of 0’s between consecutive 1’s in a binary sequence must have length at least $d$ and at most $k$. A constrained system $S$ is defined as the set of constrained sequences obtained by reading the labels of paths of a finite labeled directed graph $G$. The graph $G$ is then called the presentation of $S$.

One goal in the study of constrained systems is designing encoders that map unconstrained sequences into constrained sequences of a given constrained system $S$. A fixed rate $p:q$ finite-state encoder for $S$ encodes $p$-blocks of input bits to $q$-blocks in $S$, in a state-dependent and lossless manner. The anticipation (or decoding look-ahead) of an encoder is the smallest integer $A$, if any, such that the encoder state at each time slot $t$, together with the $q$-blocks generated at times $t, t + 1, \ldots, t + A$, determine uniquely the $p$-block input at time slot $t$.

One of the well known schemes for constructing finite-state encoders is the Adler-Coppersmith-Hassner algorithm, also known as the state-splitting algorithm. Any encoder obtained by the state-splitting algorithm from a deterministic presentation $G$ has finite anticipation.

In this work, we present lower bounds on the anticipation of encoders for given constrained systems, while strengthening the universality of the state-splitting algorithm. It is shown that if there exists an encoder with anticipation $A$ for a given system, then an unreduced version of this encoder can be obtained by the state-splitting algorithm, using $A$ rounds of splitting. Furthermore, by specifying several properties of those splittings, we obtain a lower bound on the anticipation of any encoder for a given constrained system. The new lower bound improves on previous known bounds and in particular, it is tight for several known and widely used systems, such as the $(1,7)$-RLL and the $(2,7)$-RLL constrained systems.
List of Symbols and Abbreviations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLL</td>
<td>Run-Length-Limited</td>
</tr>
<tr>
<td>(V(G), E(G))</td>
<td>Set of states and set of edges of a graph (G)</td>
</tr>
<tr>
<td>(S(G))</td>
<td>Constrained system generated by a graph (G)</td>
</tr>
<tr>
<td>(G_S)</td>
<td>Shannon cover of an irreducible constrained system (S)</td>
</tr>
<tr>
<td>(\mathcal{F}_G(v))</td>
<td>Follower set of a state (v) in a graph (G)</td>
</tr>
<tr>
<td>(v \preceq u)</td>
<td>State (v) is dominated by state (u)</td>
</tr>
<tr>
<td>(\sigma(e), \sigma(\pi))</td>
<td>Initial state of an edge (e) or a path (\pi)</td>
</tr>
<tr>
<td>(\tau(e), \tau(\pi))</td>
<td>Terminal state of an edge (e) or a path (\pi)</td>
</tr>
<tr>
<td>(\mathcal{E})</td>
<td>Encoder graph</td>
</tr>
<tr>
<td>(\mathcal{I}_\mathcal{E}(\pi, w))</td>
<td>Set of all paths in the encoder graph (\mathcal{E}) that begin with a path (\pi) and continue with suffixes that generate the word (w)</td>
</tr>
<tr>
<td>(G_\mathcal{E})</td>
<td>Determinizing graph for an encoder (\mathcal{E})</td>
</tr>
<tr>
<td>(A_G)</td>
<td>Adjacency matrix of a graph (G)</td>
</tr>
<tr>
<td>(\lambda(A))</td>
<td>Perron eigenvalue of a square irreducible nonnegative matrix (A)</td>
</tr>
<tr>
<td>(\bar{x} \preceq \bar{y})</td>
<td>(for vectors (\bar{x} = [x_u]_u) and (\bar{y} = [y_u]_u) of the same length) (x_u \preceq y_u) for every (u).</td>
</tr>
<tr>
<td>(X(A, n))</td>
<td>Set of all ((A, n))-approximate eigenvectors for a given square nonnegative integer matrix (A) and a positive integer (n)</td>
</tr>
<tr>
<td>(C(S))</td>
<td>Capacity of a constrained system (S)</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

*Input-constrained channels* are models for describing the read-write requirements of secondary storage systems, such as magnetic disks or optical devices. Coding for input-constrained channels deals with encoding arbitrary sequences, which need to be stored in a device or transmitted through a channel, into sequences that satisfy some requirements implied by the characteristics of the device or the channel. The encoding should be uniquely decodable.

A widely used family of constraints are the \((d,k)\)-run-length-limited (RLL) constraints, where each run of 0's between two consecutive 1's in a binary sequence must have length at least \(d\) and at most \(k\). The motivation for this constraint is that the binary sequences are often recorded in the NRZI (non-return-to-zero-inverse) notation, where each 1 in the binary sequence indicates a transition in the recorded signal. When transmitting over a bandwidth limited channel, a high transition frequency may cause intersymbol interference. The parameter \(d\) which determines the highest transition frequency protects against this phenomenon. On the other hand, the parameter \(k\) ensures that the transition frequency would be sufficient for synchronizing the read clock according to the transmitted signal.

Many commercial systems today use codes that are designed to satisfy \((d,k)\)-RLL constraints. For example, a code for the \((2,7)\)-RLL constraint is applied in the IBM 3380 system, and a code for the \((1,7)\)-RLL constraint is implemented in the ISS 8470 drive. A code with the \((2,10)\)-RLL constraint is used for compact disks. A related family of constraints is the class of *multiple-spaced RLL codes*, which are characterized by the three parameters \((d,k,s)\). A binary sequence satisfies such a constraint if the length of every run of 0's between two consecutive 1's is of the form \(d + is\), where \(i\) is a nonnegative integer and \(k \geq d + is\). Codes with \(s = 2\) were shown to have some practical value in magnetic and magneto-optical recording.

In all the constraints we will be interested in, the set of allowed sequences are obtained by reading the labels of paths in a labeled directed graph which presents the given constraint. A more detailed definition of constraints through graph presentation will be given in Chapter 2. It should be mentioned that input-constrained channels are often referred to as *constrained systems* or simply *constraints*.
An encoder for a constrained channel is modeled by a synchronized finite-state machine. A rate \( p : q \) finite-state encoder accepts input blocks of \( p \) bits and generates \( q \)-blocks of output symbols. The output \( q \)-block which is produced by the encoder at time slot \( t \) and the state of the encoder at time slot \( t + 1 \) (next state) are both determined by the encoder state at time slot \( t \) (current state) and by the input \( p \)-block at time slot \( t \).

The sequence obtained by concatenating the output blocks generated by the encoder must satisfy the constraints of the channel. Moreover, it is required that any two sequences of input \( p \)-blocks that both lead the encoder from the same initial state to the same terminal state, will be encoded into two distinct output sequences. An encoder which satisfies this elementary requirement is said to be lossless. The losslessness of the encoder ensures that the reconstruction of the input sequence is possible: Given the output sequence as well as the initial and terminal states of the encoder, the input sequence is uniquely determined. Even though the decoding of sequences generated by a lossless encoder is always possible, restoring the first input block may depend on the whole output sequence. Therefore, implementing a decoder for a lossless encoder may be impractical.

A stronger requirement is that the encoder has a finite anticipation. The anticipation of an encoder is the minimal integer \( A \) such that every pair of sequences of \( A + 1 \) input \( p \)-blocks that can be encoded from the same initial state into the same output sequence, must have the same first input \( p \)-block. Output sequences generated by an encoder of finite anticipation \( A \) are on-line decodable: Given the encoder state and the output \( q \)-block at time slot \( t \), as well as the next \( A \) output \( q \)-blocks, the decoding of the input \( p \)-block at time slot \( t \) is unique. The anticipation of the encoder is therefore equal to the amount of decoding look-ahead needed for any decoder for such an encoder. It follows that finite-state decoders exist for every encoder of finite anticipation. Since the decoding of an input \( p \)-block is delayed until the next \( A \) output blocks are read, the anticipation of the encoder is also referred to as the smallest decoding delay possible for any decoder for such an encoder. When transmitting through a noisy channel, it is preferable to use encoders for which there exist state independent decoders. Such encoders are called sliding-block decodable encoders, in which the input \( p \)-block at time slot \( t \) is reconstructed by applying a decoding function (which is independent of \( t \)) on the received output \( q \)-blocks at time slots \( t-m, t-m+1, \ldots, t+a-1, t+a \), for some prescribed \( m \) and \( a \). In particular, sliding-block decodable encoders have finite anticipation.

So far, we have mentioned losslessness, finite anticipation, and sliding-block decodability as properties that should be considered when designing encoders for input-constrained channels. Another important goal in such a design is having the encoding rate \( p/q \) as high as possible. Shannon proved in [Sha48] that the encoding rate cannot exceed a positive quantity which is unique for every input-constrained channel and is called the (Shannon) capacity of the channel. He also gave a nonconstructive proof for the existence of encoders at rational rates which approach the capacity of the channel; those rates however are strictly below the capacity.

Several schemes have been suggested in the literature for constructing finite-state encoders. The most notable one is the algorithm of Adler, Coppersmith, and Hassner [ACH83], which is known as the state-splitting algorithm. The significance of the state-splitting algorithm is that it enables the construction of encoders at any rational rate which is not
greater than the capacity of the channel. Moreover, encoders which are constructed using this algorithm always have finite anticipation (and are therefore lossless). In many cases they also have the property of sliding-block decodability.

Although the state-splitting algorithm is guaranteed to come up with an encoder for any constrained channel at any rate not greater than capacity, there are often many choices that the designer can make during the course of the algorithm. Different choices can end up with different encoders, with different number of states or different anticipations. However, one of the problems is that it is not clear which choice to take in order to obtain an encoder with desirable properties, such as a smallest anticipation or a smallest number of states. A graph presentation of the constraint serves as the input to the state-splitting algorithm. However, the graph presentation of a constraint is not unique, and the user of the state-splitting algorithm should also decide which presentation to start with.

In this work, we prove that every finite state encoder with finite anticipation can be obtained using the state-splitting algorithm and an additional operation known as reduction of states. In particular, the algorithm can generate encoders with the smallest possible anticipation, with respect to any specific constraint and any rate not greater than capacity. Furthermore, we specify some partial characterization on the way in which the state-splitting algorithm should run so that the resulting encoder will have the smallest anticipation possible. We also provide an upper bound on the size of the graph presentation that should be used as an input to the algorithm, so it can generate such an optimal encoder.

Two lower bounds on the anticipation of encoders are presented in this work. The first one is stated by means of the state-splitting algorithm, and the second one extends two former lower bounds on the anticipation of encoders, stated in [MR91] and [AMR95b]. Applying the two new bounds to some well-known constraints, we managed to prove that finite state encoders for the (1,7)-RLL constraint at rate 2:3 must have anticipation at least 2, encoders for the (2,7)-RLL constraint at rate 1:2 must have anticipation at least 3, and encoders for the (2,18,2)-RLL constraint at rate 2:5 cannot have anticipation smaller than 3. The three lower bounds are in fact tight for these constrained system, as there exist some known encoders which attain these bounds.

This work is organized as follows: In Chapter 2, we present the background, the definitions, and the terminology needed for this work. A formal statement of the results and an analysis of their significance are provided in Section 2.6. In Chapter 3, we state and prove the main lemmas that are used to obtain the two lower bounds mentioned above. In Chapter 4, we prove a necessary and sufficient condition for the existence of encoders of finite anticipation. The condition is given by means of state-splitting and it implies the first lower bound on the anticipation of encoders. Chapter 4 also includes the applications of the first lower bound. In Chapter 5, we present the upper bound on the size of the graph presentation which can serve as an input to the state-splitting algorithm that yields the encoder with the smallest anticipation possible. In Chapter 6, we present the second lower bound on the anticipation of encoders and we apply this bound to the (1,7)-RLL, (2,7)-RLL, and (2,18,2)-RLL constraints. Chapter 7 is the conclusion chapter and it contains a discussion of the achieved results and the questions that remain open.
Chapter 2

Background and Statement of Results

The background chapter is organized as follows: Sections 2.1 – 2.5 provide a summary of known results and include all the definitions and terminology needed for a formal presentation of the results. The results are then stated in Section 2.6. In Section 2.7, we bring some additional definitions and observations which are later used in the proofs and the examples of this work.

2.1 Graph Presentations

2.1.1 Labeled Graphs

In the area of coding for input-constrained channels, the set of finite sequences that satisfy a specific constraint can often be described by reading the labels of the paths in a labeled directed graph $G = (V, E, L)$ which consists of

1. a finite set of states $V = V(G)$;

2. a finite set of edges $E = E(G)$; the initial state and the terminal (target) state of an edge $e$ will be denoted by $\sigma(e)$ and $\tau(e)$, respectively;

3. output labels on the edges taken from an alphabet $\Sigma$; the labeling is denoted by a function $L : E \rightarrow \Sigma$.

All the graphs (labeled or unlabeled) that are referred to in this work are directed even if it is not said so explicitly. The notation $(u \xrightarrow{l} v)$ will denote an edge labeled $l$ from state $u$ to state $v$. A path in $G$ is a finite sequence of edges $e_1 \ldots e_m$ such that $\sigma(e_i) = \tau(e_{i-1})$. A word over $\Sigma$ is defined to be a finite sequence of symbols from $\Sigma$. If a word $w = w_1 \ldots w_m$ is obtained by reading the labels of the edges in a path $\pi$ of $G$, we say that $w$ is generated by the path $\pi$. 
Given a labeled graph $G$, we denote by $S(G)$ the set of words generated by paths in $G$. The graph $G$ is called a graph presentation of $S = S(G)$. A set of words $S$ for which there exists some graph presentation $G$ of $S$ is called a constrained system (or constraint). A specific constrained system has many different graph presentations. For example, the set of all the binary words that satisfy the (1,7)-RLL constraint is a constrained system. The graph in Figure 2.1 is one of the graph presentations of this constrained system. The notion of constrained systems is similar (although not identical) to that of regular languages in automata theory (see [HU79]).

An adjacency matrix is associated with the underlying directed graph of a labeled graph $G$ and is denoted by $A_G$. The matrix $A_G$ is a $|V(G)| \times |V(G)|$ matrix in which the entry $(A_G)_{uv}$ indicates how many edges go from $u$ to $v$ in $G$.

In our model of encoding and decoding, an encoder for a constrained system $S$ is a finite state machine whose state diagram is a labeled graph $E$ referred to as an encoder graph. Any output sequence generated by the encoder machine is a word which can be generated by some path of the encoder graph $E$, and every internal state of the finite-state machine encoder corresponds to some state of the graph $E$.

### 2.1.2 Properties of labeled Graphs

A labeled graph is called deterministic if for each state of the graph, the labels on the edges going out of the state are all distinct. The graph in Figure 2.1 is therefore a deterministic presentation for the (1,7)-RLL constrained system. Every constrained system has deterministic graph presentations. Later in our work, we present one way to construct a deterministic presentation of a system $S$ according to another nondeterministic presentation $G$ of $S$. The deterministic graph obtained in this case is called a determinizing graph of $G$.

Two other properties which a labeled graph may have and which are weaker than determinism is losslessness and finite anticipation. A labeled graph is lossless if for any pair of states $u$ and $v$ in $V(G)$ and for any word $w$, there cannot be more than one path in $G$ that starts at $u$, terminates at $v$, and generates the word $w$. An encoder graph must be lossless to ensure the reconstruction of the input sequence, as will be explained in Section 2.3.

A labeled graph $G$ has a finite anticipation if there is an integer $A$ such that any two paths of length $A + 1$ in $G$, that have the same initial state and generate the same word, must have the same initial edge. The anticipation of a graph is the smallest integer $A$ for which this holds. The anticipation of an encoder graph $E$ is in fact the amount of look-ahead.
required by any decoder for such an encoder, as will be explained in Section 2.3.

Note that the anticipation of a deterministic graph is zero. Determinism therefore implies finite anticipation. In case that a labeled graph includes paths of arbitrarily large lengths, finite anticipation of the graph implies its losslessness. It should be mentioned that a graph with finite anticipation is sometimes said to be lossless of finite order, as in [Huff59] and [Ewen65].

2.1.3 Irreducibility of Graph Presentations

A directed graph $G$ is irreducible if it is strongly connected, i.e., for every ordered pair of states $u$ and $v$ in $G$ there is a path from $u$ to $v$ in $G$. An irreducible component $G_0$ of a graph $G$ is a maximal irreducible subgraph of $G$, that is, there is no other irreducible subgraph of $G$ that contains $G_0$. An irreducible sink is an irreducible component $H$, such that every outgoing edge of a state in $H$ is an incoming edge of a state in $H$. Any finite graph is composed of a finite number of irreducible components, at least one of which is an irreducible sink.

A constrained system $S$ is an irreducible system if for every two words $w, w' \in S$ there exists some word $z \in S$ such that $uzw'$ is also a word of $S$. A system presented by an irreducible labeled graph is an irreducible system. The following lemma which implies the opposite direction is taken from [MR91]:

**Lemma 2.1** Let $S$ be an irreducible constrained system and let $G$ be a labeled graph such that $S \subseteq S(G)$. Then for some irreducible component $G'$ of $G$, $S \subseteq S(G')$.

A corollary of the above lemma is that every irreducible constrained system can be presented by some irreducible labeled graph.

2.1.4 Reduced Graphs and the Shannon Cover

Given a graph $G$ and a state $u \in V(G)$, the follower set of $u$ which is denoted by $F_G(u)$ is the set of words generated by paths in $G$ for which $u$ is the initial state. Two states $u$ and $v$ in $G$ are said to be equivalent states if they have the same follower set, i.e., $F_G(u) = F_G(v)$. A concept similar to equivalence of states is domination of states. A state $u$ in a graph presentation $G$ is dominated by a state $u'$ in a graph presentation $G'$, $u \preceq u'$, if $F_G(u) \subseteq F_G(u')$.

A reduction of states in a graph presentation means replacing a set of equivalent states with one representative of this set. All the incoming edges of the replaced (reduced) states are redirected into the representative state. More specifically:

**Definition 2.1** Let $v_1, \ldots, v_k$ be states in a graph presentation $G$, where $F_G(v_1) = F_G(v_2) = \ldots = F_G(v_k)$. Reducing $v_1, \ldots, v_k$ means turning $G$ into $G'$ by the following three operations:
(a) choosing one specific $v_i$ out of $v_1, \ldots, v_k$,
(b) removing $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k$ from $G$, including their outgoing and incoming edges,
(c) for every state $u \in G'$, if $G$ contains an edge $(u \overset{i}{\rightarrow} v_j)$, $j \in 1, \ldots, k$, then an edge $(u \overset{i}{\rightarrow} v_i)$ is added to $G'$.

Reducing equivalent states in an arbitrary graph $G$ does not change the constrained system presented by the graph, since every word generated in the original graph is generated also in the reduced graph (from an equivalent state), and vice versa. A reduced graph is a graph that does not contain any pair of equivalent states. Given a deterministic graph $G$, there exists an algorithm, known as the Moore algorithm, for constructing a reduced deterministic graph $H$ where $S(H) = S(G)$ (see [Koh78]).

The Shannon cover $G_S$ of an irreducible constrained system $S$ is a reduced deterministic graph presentation of $S$. The following proposition, taken from [MR91], states some well-known properties of the Shannon cover.

**Proposition 2.2**

1. The Shannon cover $G_S$ of an irreducible system $S$ is unique (up to a graph isomorphism).

2. Suppose that $G'$ is an irreducible deterministic presentation of $S$. Then for every state $u' \in V(G')$ there is a unique state $u \in V(G_S)$ so that $F_{G'}(u') = F_{G_S}(u)$.

3. Suppose that $G$ and $G'$ are two irreducible deterministic presentations of the same system $S$. Then for every $u' \in V(G')$ there is at least one $u \in V(G)$ such that $F_{G'}(u') = F_G(u)$.

4. Suppose that $G$ and $G'$ are irreducible deterministic presentations of the two systems $S$ and $S'$, respectively, where $S' \subseteq S$. Then for every $u' \in V(G')$ there is at least one $u \in V(G)$ such that $u' \leq u$, i.e., $F_{G'}(u') \subseteq F_G(u)$.

Note that the graph in Figure 2.1 is the Shannon cover of the $(1,7)$-RLL constraint.

### 2.1.5 Two Graph Constructions

Let $G$ be a labeled graph. The $k$th power graph of $G$, denoted by $G^k$, is a labeled graph with the same set of states as $G$, in which the edges represent the paths of length $k$ in $G$. An edge labeled $\overrightarrow{l} = l_1 \ldots l_k$ goes in $G^k$ from state $u$ to state $v$ if and only if a path of length $k$ generating $\overrightarrow{l}$ goes in $G$ from $u$ to $v$. The system presented by $G^k$ is denoted by $S^k$. It is well-known that the adjacency matrices of two graphs $G$ and $G^k$ satisfy $A_{G^k} = A_G^k$.

Let $H$ and $G$ be two labeled graphs. The label product (or fiber product) of the two graphs is the labeled graph $H \ast G$ in which the states are defined by

$$V(H \ast G) = V(H) \times V(G) = \{(u', u) \mid u' \in V(H), u \in V(G)\}$$

An edge $(u', u) \overset{i}{\rightarrow} (v', v)$ exists in $H \ast G$ if and only if an edge $u' \overset{i}{\rightarrow} v'$ exists in $H$ and an edge $u \overset{i}{\rightarrow} v$ exists in $G$. It is easy to verify that $S(H \ast G) = S(H) \cap S(G)$.
2.2 Capacity and Perron-Frobenius Theory

The capacity of a constrained system $S$ is denoted by $C(S)$ and indicates the maximal possible rate for encoding arbitrary binary sequences into words of the system $S$. Let $N(m; S)$ denote the number of words of length $m$ in $S$. The capacity of $S$ is defined by:

$$C(S) = \lim_{m \to \infty} \frac{1}{m} \log_2 N(m; S)$$

The effective way to compute the capacity of constrained systems is based on a branch of linear algebra, known as the Perron-Frobenius theory, which deals with the properties of nonnegative matrices. The following theorems, including their proofs, can be found in [Sen80] (Chapter 1) or in [MRS].

Consider the adjacency matrix $A_G$ of a graph presentation $G$. The matrix $A_G$ is a real nonnegative square matrix, i.e., all its entries are real and nonnegative numbers. If $G$ is an irreducible graph, then $A_G$ is an irreducible matrix. Algebraically, a nonnegative real square matrix $A$ is called irreducible if for every two indexes $u$ and $v$, there exists some positive integer $k$ such that $A^k_{u,v} > 0$.

For an irreducible matrix $A$, we denote by $\lambda(A)$ the spectral radius of $A$, which is the largest among the absolute values of the eigenvalues of $A$. The following theorem is known as the Perron-Frobenius Theorem for irreducible matrices.

**Theorem 2.3** Let $A$ be an irreducible matrix.

1. $\lambda(A)$ is an eigenvalue of $A$ and $A$ has positive right and left eigenvectors associated with the eigenvalue $\lambda(A)$;

2. $\lambda(A)$ is a simple eigenvalue of $A$, i.e., it appears as a root of the characteristic polynomial of $A$ with multiplicity 1. All the right (left) eigenvectors associated with $\lambda(A)$ are scalar multiples of one another.

3. $$\min_u \sum_v (A)_{u,v} \leq \lambda(A) \leq \max_u \sum_v (A)_{u,v}$$

In the sequel we will refer to $\lambda(A)$ as the Perron eigenvalue of the matrix $A$ and the right eigenvectors of $A$, associated with $\lambda(A)$, will be considered as the Perron eigenvectors of $A$ (or of the graph $G$ in case that $A = A_G$). Consider an irreducible graph $G$ which is a regular graph, i.e., the outdegrees of the states in $G$ are all equal. By property 3 of the Perron-Frobenius Theorem, the Perron eigenvalue of the adjacency matrix $A_G$, where $G$ is a regular irreducible graph, is equal to the outdegree of the states in $G$. The corresponding right Perron eigenvectors are (scalar multiples of) the all-one vector.

A method for computing the capacity of a constrained system $S$ is described in the following theorem.
Theorem 2.4 Let \( S \) be an irreducible constrained system and let \( G \) be an irreducible lossless presentation of \( S \). Then,

\[
C(S) = \log_2 \lambda(A_G)
\]

Remark: Hereafter all logarithms are taken to base 2 unless another base is specified.

2.3 Finite-State Encoders

Let \( S \) be a constrained system and let \( n \) be a positive integer. An \((S, n)\) encoder is a labeled graph \( \mathcal{E} \) such that:

- \( S(\mathcal{E}) \subseteq S \),
- \( \mathcal{E} \) is lossless,
- \( \mathcal{E} \) is regular of outdegree \( n \).

Note that in case of an irreducible \((S, n)\) encoder, the Shannon capacity of \( S(\mathcal{E}) \) is exactly \( \log_2 n \). Indeed, since \( \mathcal{E} \) is an irreducible regular graph of outdegree \( n \) we get by Theorem 2.3 (part 3) that \( \lambda(A_E) = n \); and by the losslessness of \( \mathcal{E} \) we get by Theorem 2.4 that \( C(S(\mathcal{E})) = \log \lambda(A_E) = \log n \). As \( S(\mathcal{E}) \subseteq S \), we must have \( \log n \leq C(S) \).

A (legal) input tagging of the edges of an \((S, n)\) encoder \( \mathcal{E} \) is a function \( f : E(\mathcal{E}) \rightarrow \{0, \ldots, n-1\} \) such that the restriction of \( f \) to the set of outgoing edges in each state of \( \mathcal{E} \) is one-to-one and onto the set \( \{0, \ldots, n-1\} \). A tagged encoder is an encoder graph with a specific legal tagging of its edges. Each edge of the encoder then has an (output) label and an (input) tag.

A rate \( p : q \) encoder machine for a constrained system \( S_0 \) can be based on any irreducible \((S, n)\) encoder graph \( \mathcal{E} \), where \( S = S_0^* \) and \( n = 2^p \). The encoder \( \mathcal{E} \) is legally tagged with the set of binary words of length \( p \). To encode an arbitrary sequence of bits, we divide the input sequence into consecutive words of length \( p \), and we choose an arbitrary \( \mathcal{E} \)-state as an initial state. In any stage of the encoding, the current state \( u \) of \( \mathcal{E} \) and the current input word of length \( p \) uniquely determine the outgoing edge \( e \) of \( u \) whose input tag is identical to the input word. The next state is defined to be \( \tau(e) \) and the current output \( q \)-block is defined to be \( L(e) \). An encoder graph for the \((2,7)\)-RLL constraint at rate \( 1:2 \) is illustrated in Figure 2.2. The encoder is due to Franaszek (see [How89]).

Suppose that the encoder graph \( \mathcal{E} \) is of finite anticipation \( A \) (a property which is connected only to the labeling of the edges and not to the input tagging). We already mentioned that \( A \) is the amount of the look-ahead required by every decoder for decoding output sequences of \( \mathcal{E} \). A state-dependent decoder, for example, restores the input sequence by reconstructing the path of the encoder by which the output sequence was generated. When given a state \( v \) of \( \mathcal{E} \) and a label \( l \) in the output sequence which was generated from \( v \), \( A \) additional labels are needed (in the worst case) for a state dependent decoder in order to restore the edge by which \( l \) was generated. Finding that edge is equivalent to finding the input tag that corresponds to \( l \).
A special case of encoders with finite anticipation are sliding-block decodable encoders, which are tagged encoders that satisfy the following: there exist two integers $m$ and $a$ such that for any two paths $e_m e_{m+1} \ldots e_0 e_1 \ldots e_a$ and $e'_m e'_{m+1} \ldots e'_0 e'_1 \ldots e'_a$ that generate the same word, the edges $e_0$ and $e'_0$ carry the same input tag. We then say that the encoder is $(m,a)$-sliding-block decodable. In case of an $(m,a)$-sliding block decodable encoder, a corresponding decoder can be constructed which is called a sliding block decoder. Such a decoder determines the input tag at timeslot $t$ in a state independent manner, according to a “window” of size $m+a+1$ which contains the output labels in timeslots $t-m, \ldots, 0, \ldots, t+a$. Such encoders and decoders are preferred in cases of noisy channels, since every error in the output sequence of the encoder will cause a bounded number of errors in the decoded sequence when a sliding block decoder is used. Every $(m,a)$-sliding block decodable encoder is of finite anticipation $a$.

We note that in addition to the model of $(S,n)$ encoders, there exist some other well-known encoding models, such as:

- **Variable length encoders** that are based on underlying labeled graphs in which the lengths of the edge labels are not necessarily the same on all the edges.

- **Look-ahead encoders** in which the next state of the encoder depends not only on the current state and the first coming input tag, but also on some upcoming input tags in the sequence. Techniques for constructing encoders of this type are presented for example in [LemCo82].

- **Bounded delay encoders** which are a generalization of look-ahead encoders. Some good encoders of this type are constructed for well-known constraints in [Holl95].
Let $S$ be a constrained system (not necessarily irreducible) which is presented by a deterministic graph $G$. It follows from Lemma 2.1 (see [MR91, Proposition 3]) that if $E$ is an $(S, n)$ encoder, then any irreducible sink $E'$ of $E$ is an $(S', n)$ encoder for an irreducible constrained system $S'$ which is generated by an irreducible component of $G$. Consequently, there is no loss of generality in requiring the existence of an irreducible encoder $E$ of finite anticipation, since the existence of an $(S, n)$ encoder with finite anticipation implies the existence of some irreducible $(S, n)$ encoder with finite anticipation.

### 2.4 The State-Splitting Algorithm

#### 2.4.1 State Splitting

The state-splitting algorithm due to Adler-Coppersmith-Hassner [ACH83] allows the construction of $(S, n)$ encoders with finite anticipation for every constrained system $S$ and every positive integer $n$ such that $\log n \leq C(S)$. A full description of the algorithm, including some improvements, examples, and variations, appears in [MSW92]. The algorithm starts with a deterministic presentation $G$ of $S$ and, in a finite number of steps, called rounds of state-splitting, it turns $G$ into another presentation of $S$ in which the outdegree of every state is at least $n$. In each such round, a graph presentation of $S$ is obtained whose anticipation is equal to, or greater by 1 than, the anticipation of the graph obtained one round before.

A basic out-splitting of a state $u$ in a graph presentation $G$ means turning the graph $G$ into another graph $G'$ by the following operations:

1. Partitioning $E_u$ — the set of outgoing edges of the state $u$ — into two disjoint sets, $E_u = E_u^{(1)} \cup E_u^{(2)}$.
2. Removing state $u$ from the graph $G$, including its outgoing and incoming edges, and replacing $u$ with two new states, $u^{(1)}$ and $u^{(2)}$. States $u^{(1)}$ and $u^{(2)}$ are considered as descendant states of $u$, and $u$ is considered as the parent state of $u^{(1)}$ and $u^{(2)}$.
3. For every edge $e = (u \xrightarrow{i} v)$ which belongs to $E_u^{(i)}$ ($i = 1, 2$) and for which $v \neq u$, we draw an edge $e' = (u^{(i)} \xrightarrow{i} v)$ in $G'$. We say that $u^{(i)}$ inherits the edge $e$ of $u$. The edge $e'$ is considered as the descendant edge of the edge $e$.
4. For every edge $(v \xrightarrow{i} u)$ in $G$, where $v \neq u$, the graph $G'$ contains the two edges $(v \xrightarrow{i} u^{(1)})$ and $(v \xrightarrow{i} u^{(2)})$.
5. For every self loop $e = (u \xrightarrow{i} u)$ in $G$, where $e \in E_u^{(i)}$ ($i = 1, 2$), the graph $G'$ contains the two edges $(u^{(i)} \xrightarrow{i} u^{(1)})$ and $(u^{(i)} \xrightarrow{i} u^{(2)})$.

Figure 2.3 illustrates the effect of a basic out-splitting on a graph.

In a round of state out-splitting of a labeled graph $G$, the set $E_u$ of outgoing edges of every state $u$ in $G$ is partitioned into $N(u)$ disjoint nonempty sets $E_u = E_u^{(1)} \cup \ldots \cup E_u^{(N(u))}$.  

13
A new labeled graph $G'$ is constructed out of $G$, in which the states are the descendant states $u^{(1)}, \ldots, u^{(N(u))}$ of every state $u$ of $G$. For every edge $e = (u \xrightarrow{i} v)$ of $G$, where $e \in E^{(i)}$, the graph $G'$ contains the edges

$$\{e^{(r)} = (u^{(i)} \xrightarrow{j} v^{(r)}) ; \ r = 1, 2, \ldots N(v)\}$$

Notice that for every state $u$ in $G$, the number $N(u)$ of descendant states in $G'$ is not smaller than 1 and not greater than the outdegree of $u$ in $G$. In addition, note that the splitting of every state in $G$ does not depend on the splitting of any other state of $G$ in the same round. It should be mentioned that if a graph $G'$ is obtained from $G$ by a round of out-splitting, then $G'$ can be obtained from $G$ by a finite sequence of basic splittings.

The properties of a graph $G'$ obtained from $G$ by a round of out-splitting are described in the following proposition:

**Proposition 2.5** Let $G'$ be a graph obtained from $G$ by a round of out-splitting. Then,

- $S(G') = S(G)$
- If the anticipation of $G$ is $A$, then the anticipation of $G'$ is at most $A + 1$.
- If $G$ is an irreducible graph, then $G'$ is also irreducible.

The operation of *in-splitting* is defined by recasting all of the above in terms of the incoming edges of the states, instead of their outgoing edges. Unless otherwise said, by the term *state-splitting* we mean a state *out*-splitting.
2.4.2 Approximate Eigenvectors and Consistent Splitting

When designing an $(S, n)$ encoder by applying out-splittings on a graph presentation of $S$, the tool of approximate eigenvectors, as defined right below, is used to direct the designers in choosing the out-splittings which lead to a graph presentation in which the outdegree of every state is at least $n$.

Given a nonnegative integer square matrix $A$ and an integer $n$, an $(A, n)$-approximate eigenvector is a nonnegative integer vector $x = [x_u]_u$ such that $\bar{x} \neq \bar{0}$ and $A\bar{x} \geq n\bar{x}$. The set of all $(A, n)$-approximate eigenvectors is denoted by $X(A, n)$.

**Proposition 2.6** Let $A$ be a nonnegative integer square matrix and let $n$ be a positive integer. Then,

1. $X(A, n) \neq \emptyset$ if and only if $n \leq \lambda(A)$.
2. In the case where $A$ is an irreducible matrix and $n = \lambda(A)$, every $(A, n)$-approximate eigenvector is a (true) right eigenvector of $A$, associated with the eigenvalue $n$.

Given a nonnegative integer vector $\bar{x}$, we denote by $X(A, n, \bar{x})$ the set of $(A, n)$-approximate eigenvectors $\tilde{x}$ such that $\bar{x} \leq \tilde{x}$ (the inequality is componentwise). For every two $(A, n)$-approximate eigenvectors, $\tilde{x}$ and $\tilde{x}'$, the vector defined by $[\max(x_u, x_u')]_u$ is also an $(A, n)$-approximate eigenvector. As a result of this observation, if the set $X(A, n, \bar{x})$ is nonempty, then there exists an $(A, n)$-approximate eigenvector which is the largest componentwise among the vectors in $X(A, n, \bar{x})$.

There is an algorithm for computing approximate eigenvectors due to Franaszek [Fra82]. Given a nonnegative integer square matrix $A$, an integer $n$ and a nonnegative integer vector $\bar{x}$, the Franaszek algorithm computes the largest (componentwise) vector in $X(A, n, \bar{x})$, if this set is nonempty. Otherwise, the output is the zero vector. Applying the algorithm on a vector $\bar{x}$ whose components are all equal to a constant $K$, we can find the largest $(A, n)$-approximate eigenvector, if there exists one, among those with maximal component not greater than $K$.

Let $A_G$ be the adjacency matrix of a graph $G$ and let $\bar{x} = [x_u]_{u \in V(G)}$ be an $(A_G, n)$-approximate eigenvector. Let us define the weight of a state $u$ in $G$ to be its respective component $x_u$ in the vector $\bar{x}$. The inequality $A\bar{x} \geq n\bar{x}$ then means that the sum of the weights of the outgoing neighbors of $u$ in $G$ is at least $n$ times the weight of $u$ itself. More formally,

$$\sum_{e \in E_u} x_{r(e)} \geq nx_u$$

Let $G$ be a labeled graph and let $\bar{x} = [x_u]_{u \in V(G)}$ be an $(A_G, n)$-approximate eigenvector. The idea behind the concept of $\bar{x}$-consistent splitting is to preserve the approximate eigenvector inequality also in the split graph. An $\bar{x}$-consistent out-splitting of the graph $G$ is defined by an out-splitting of $G$, and by dividing the weight $x_u$ of each $u \in V(G)$ among its descendant states. If $u$ is split into the states $u^{(1)}, \ldots, u^{(N(u))}$, then the weight $x_u$ is divided...
into $N(u)$ positive integers $x^{(1)}_u + \ldots + x^{(N(u))}_u = x_u$. The vector $x'$ defined by $x_{u(j)}' = x_{u(j)}$ is an $(A_{G'}, n)$-approximate eigenvector, where $G'$ is the split $G$.

It can be proved that for every irreducible graph $G$, unless the all-one vector $1$ is an $(A_G, n)$-approximate eigenvector, an $\varepsilon$-consistent splitting of $G$ exists, where $\varepsilon$ is an $(A_G, n)$-approximate eigenvector. Hence, the state-splitting algorithm for constructing an $(S, n)$ encoder, where $\log n \leq C(S)$, works as follows:

1. Select an irreducible and deterministic graph presentation $G$ of $S$ and an $(A_G, n)$-approximate eigenvector $y$.

2. Remove from $G$ all the states of weight zero (according to $y$), including their outgoing and incoming edges, and let $H$ be an irreducible sink of the resulting graph. ($H$ is an irreducible presentation of $S'$, where $S' \subseteq S$.) Let $\bar{x}$ be the $(A_H, n)$-approximate eigenvector in which the component of every state of $H$ is equal to the weight of this state according to $y$.

3. If the all-one vector is not an $(A_H, n)$-approximate eigenvector, then
   - Execute some $\bar{x}$-consistent splitting to get a new irreducible presentation $H'$ of $S'$ and a corresponding $(A_{H'}, n)$-approximate eigenvector $\bar{x}'$. Note that there may exist several $\bar{x}$-consistent splittings of $H$ and the designer should make a decision which splitting to choose.
   - $H \rightarrow H'; \bar{x} \rightarrow \bar{x}'$
   - Go back to 3.

4. Since every $\bar{x}$-consistent splitting decreases some weight in the graph, after a finite number of splittings the all-one vector becomes an $(A_H, n)$-approximate eigenvector. When this occurs, the outdegree of every state in $H$ is at least $n$. Remove excess edges to get a regular graph of outdegree $n$. By the properties of state-splitting, this graph is an $(S, n)$ encoder and its anticipation is not greater than the (finite) number of rounds of splitting that were executed.

Notice that by using the state-splitting algorithm, we can design many different $(S, n)$ encoders, depending on the deterministic presentation of $S$ we start with, the approximate eigenvector we choose, and the $\varepsilon$-consistent splittings we decide to execute.

A full splitting of a graph presentation is a splitting process which ends with a graph $H$ for which the all-one vector is an $(A_H, n)$-approximate eigenvector.

### 2.4.3 Upper Bounds on the Anticipation of Encoders

An upper-bound on the anticipation of $(S, n)$ encoders is derived in [MSW92] from the properties of the state-splitting algorithm. Given a constrained system $S$, presented by a deterministic graph $G$, and a positive integer $n$ such that $\log n \leq C(S)$, the state-splitting algorithm, when applied to $G$ according to an $(A_G, n)$-approximate eigenvector $\bar{x}$, yields an $(S, n)$ encoder $E$ whose anticipation is not greater than the number of rounds required to
split $G$ according to $\bar{x}$. In every round of the splitting, at least one state is replaced by descendant states of smaller weights. Therefore,

$$A(\mathcal{E}) \leq \min_{\bar{x} \in \mathcal{F}(A_G, n)} \sum_{u \in V(G)} (\bar{x})_u - 1$$

This upper bound might be exponential in the number of states $V(G)$, as proved by an example given in [MR91]. There are some other upper bounds on the anticipation of $(S, n)$ encoders, which are linear in the number of states $V(G)$. In [Ash87] and [Ash88], upper bounds are derived by examining the splitting capabilities of $G^t$, for $t = O(|V(G)|)$. Other upper bounds which are linear in $|V(G)|$ are obtained in [AMR95a] and in [AshII] using the so-called stethering method for constructing $(S, n)$ encoders.

### 2.5 Lower Bounds on the Anticipation – Previous Results

As this work deals with lower bounds on the anticipation of encoders, we first present previous results in this area. The following lower bound is due to [MR91].

**Theorem 2.7** Let $S$ be a constrained system presented by a deterministic graph $G$ and let $\mathcal{E}$ be an $(S, n)$ encoder. Then,

$$A(\mathcal{E}) \geq \min_{\bar{x} \in \mathcal{F}(A_G, n)} \log_n \left( \max_{u \in V(G)} (\bar{x})_u \right)$$

Equivalently, the existence of an $(S, n)$ encoder with anticipation $A$ implies the existence of some $(A_G, n)$-approximate eigenvector in which the maximal component is not greater than $n^A$. This necessary condition holds for every deterministic presentation $G$ of $S$, particularly for the Shannon cover of $S$, when $S$ is irreducible.

A second lower bound on the anticipation of $(S, n)$ encoders is provided in [AMR95b] and is summarized in the following theorem.

**Theorem 2.8** Let $S$, $G$, $n$, and $\mathcal{E}$ be as in Theorem 2.7. The anticipation $A$ of $\mathcal{E}$ is at least the smallest integer $t$ such that there exists an $(A_G, n)$-approximate eigenvector $\bar{x}$ and the graph $G^t$ can be fully split in one round, in consistency with $\bar{x}$. In the split $G^t$, each state has outdegree not smaller than $n^t$.

The split $G^t$ can be easily converted to an $(S^A, n^A)$ encoder by removing excess edges. It is also shown in [AMR95b] that this $(S^A, n^A)$ encoder can be converted to an $(S, n)$ encoder with anticipation not greater than $2A - 1$. The number of states in the $(S, n)$ encoder can be bounded from above by a number which depends on sizes of follower sets in $S$.

The proofs of these two lower bounds are based on a specific construction of a deterministic graph for the encoder graph $\mathcal{E}$. In this work, two improvements of these results are presented, and the proofs are also based on a (slightly different) construction of a deterministic graph for $\mathcal{E}$.
2.6 Results of This Work

We now state the two main theorems of this work, Theorem 4.1 and Theorem 6.1. Both theorems can be viewed as lower bounds on the anticipation of encoders for a given constrained system $S$ and a given rate not greater than $C(S)$. The proofs of the theorems are based on the construction of a certain determinizing graph, the properties of which are presented in Chapter 3.

2.6.1 First Main Theorem – Theorem 4.1

Let $S$ be an irreducible constrained system and let $n$ be a positive integer where $C(S) \geq \log n$. Suppose there is an irreducible deterministic presentation $G$ of $S$ that can be fully split in $A$ rounds according to some $(A_G, n)$-approximate eigenvector. By the known properties of state splitting, there exists an $(S, n)$ encoder whose anticipation is not greater than $A$. Actually, the split graph obtained from $G$, after deleting excess edges, is an $(S, n)$ encoder graph whose anticipation is not greater than $A$. Theorem 4.1 states a converse result and it also gives a characterization of a splitting process in which a given encoder can be obtained:

**Theorem 4.1:** Let $S$ be an irreducible constrained system and let $n$ be a positive integer where $C(S) \geq \log n$. Suppose there exists some irreducible $(S, n)$ encoder $E$, of a finite anticipation $A$. Then there exists an irreducible deterministic graph $G$ which is a presentation of $S$, not necessarily reduced, and $G$ satisfies the following:

(i) There is an $(A_G, n)$-approximate eigenvector $\bar{e}$, whose largest component is not greater than $n^A$.

(ii) The graph $G$ can be fully split according to $\bar{e}$ in $A$ rounds of splitting. An $(S, n)$ encoder $E_G$, whose anticipation is $A$, is obtained after deleting excess edges.

(iii) In each round of splitting, each of the states is split into no more than $n$ states.

(iv) In the $i$th round of splitting, the components of the resulting approximate-eigenvector are not greater than $n^{A-i}$.

(v) The encoder $E$ can be obtained from $E_G$ by a reduction of equivalent states.

(In fact, we prove a stronger relationship between the encoders $E$ and $E_G$, as will be explained in the sequel.)

The significance of Theorem 4.1 is manifested by the following two main aspects:

1. Improving the lower bound on the anticipation of encoders, and —

2. Strengthening the universality of the state-splitting algorithm.

**Lower Bound on Anticipation**

The characterization of the splitting process in which $G$ can be fully split enables us to use
Theorem 4.1 as a lower bound on the anticipation of any \((S, n)\) encoder. Suppose that for a given constraint \(S\), a given integer \(n\) and every integer \(A \leq a\), there is no deterministic presentation that can be split as required in Theorem 4.1. We then conclude that no \((S, n)\) encoder exists with anticipation \(a\) and therefore, \(a + 1\) is a lower bound on the anticipation of \((S, n)\) encoders.

In Section 4.3, we use Theorem 4.1 to improve the lower bounds on the anticipation of encoders for the \((1,7)\)-RLL constrained system at rate 2:3 and for the \((2,18,2)\)-RLL constrained system at rate 2:5. The new lower bounds achieved are proved to be tight, as there exists some known encoders for these constrained systems and rates which attain the new bounds.

**Universality of State-Splitting**

Theorem 4.1 actually states that every \((S, n)\) encoder of finite anticipation can be constructed using the state-splitting algorithm, combined with reductions of equivalent states. Given an \((S, n)\) encoder \(E\), we guarantee the existence of an irreducible and deterministic presentation \(G\) of \(S\), to which the state-splitting algorithm can be applied in order to get an unreduced version of \(E\).

A parallel result is derived in [AM95] for the case of sliding-block decodable encoders (which are a special case of encoders with finite anticipation). Given a sliding block decoder (or decoding function) \(D\), it is proved that there is a deterministic presentation \(G\) of \(S\) and there is an \((A_G, n)\)-approximate eigenvector \(\tilde{\pi}\) so that \(G\) can be split according to \(\tilde{\pi}\), yielding an \((S, n)\) encoder, corresponding to the decoder \(D\).

Theorem 4.1 does not guarantee that the Shannon cover of \(S\) can always serve as the graph \(G\). In Chapter 5, we give a specific example in which the Shannon cover of a system does not split as required by Theorem 4.1, but another deterministic presentation of the same system does satisfy the requirements of the theorem. In that chapter, we also present two upper bounds on the number of states in a deterministic presentation of \(S\) which can also be split in \(A\) rounds, provided that there is an \((S, n)\) encoder with anticipation \(A\).

### 2.6.2 Second Main Theorem – Theorem 6.1

The significance of Theorem 6.1 is that it enables us to obtain a lower bound on the anticipation of encoders for a given irreducible system \(S\) in terms of any deterministic presentation of \(S\) and particularly the Shannon cover of \(S\). The lower bound we present in Theorem 6.1 improves on the bounds stated in [MR91] and in [AMR95b] and can be viewed as an extension of both of those results. We recall that the bound in [MR91] states that the existence of an \((S, n)\) encoder with anticipation \(A\) implies that every deterministic presentation \(G\) of \(S\) has an \((A_G, n)\)-approximate eigenvector whose maximal component is not greater than \(n^A\). The following theorem adds some more conditions that must be satisfied by such an approximate eigenvector.

**Theorem 6.1:** Let \(S\) be an irreducible constrained system, let \(n\) be a positive integer where \(C(S) \geq \log n\), and let \(G\) be any irreducible deterministic presentation of \(S\). Suppose that there exists some irreducible \((S, n)\) encoder whose anticipation is \(A\). Then there exists an \((A_G, n)\)-approximate eigenvector \(\tilde{\pi}\) (which is also an \((A_G^k, n^k)\)-approximate eigenvector),
such that the following holds:

1. The largest component of \( \bar{z} \) is not greater than \( n^A \).

2. For every integer \( k \) in the range \( 1 \leq k \leq A \), the states of \( G^k \) can be split in one round consistently with the \( (A G^k, n^k) \)-approximate eigenvector \( \bar{z} \), such that the components of the resulting approximate eigenvector are not greater than \( n^{A-k} \) and each of the states in \( G^k \) is split into no more than \( n^k \) states.

Notice that Condition 1 is equivalent to the requirement of the bound from [MR91], while the lower bound in [AMR95b] is equivalent to Condition 2 for the special case \( k = A \). In Chapter 6, we use Theorem 6.1 to derive tight lower bounds on the anticipation of encoders for the (2,7)-RLL constraint at rate 1:2, for the (1,7)-RLL constraint at rate 2:3, and for the (2,18,2)-RLL constraint at rate 2:3.

Unlike Theorem 4.1, Theorem 6.1 gives only a necessary condition on the existence of encoders with anticipation \( A \). Theorem 6.1 does not guarantee that the capability of the graphs \( G^k \) to be fully split in one round is also a sufficient condition for the existence an \((S, n)\)-encoder with anticipation \( A \). Investigating whether this property is valid or not is still an open problem.

## 2.7 Some More Definitions and Properties

In this section we bring some more definitions and observations which are used in the following chapters in the various proofs and examples.

### 2.7.1 Input Tagging of Graphs

A *tagged encoder* is an encoder graph with a specific legal tagging of its edges, as defined in Section 2.3. Each edge of the encoder then has an (output) label and an (input) tag. If we consider these pairs of label and tag as a new labeling of the encoder graph, then every tagged encoder becomes a deterministic graph. A legal tagging can be defined also for nonregular graphs, where the \( d_u \) outgoing edges of a state \( u \) are mapped one-to-one and onto the set \( \{0, \ldots, d_u - 1\} \).

### 2.7.2 Equivalence, Reduction, and Expansion of States

Two states, \( u \) and \( u' \), in a graph presentation \( G \) are *equivalent states* if they have the same follower sets. We now define the concept of strong equivalence. Two states \( u \) and \( u' \) are \( \theta \)-strongly equivalent simply if they are equivalent states. The states \( u \) and \( u' \) are \( t \)-strongly-equivalent if the following conditions hold:

1. The set \( E_u \) of edges that go out of \( u \) contains the same number of edges as does the set \( E_{u'} \) of edges that go out of \( u' \). A one-to-one and onto function \( \varphi : E_u \rightarrow E_{u'} \) can be defined in such a way that for every \( e \in E_u \), both \( e \) and \( \varphi(e) \) have the same labels.
2. For every $e \in E_u$, the target states of $e$ and $\varphi(e)$ are $(t-1)$-strongly-equivalent.

Notice that for every nonnegative integers $r$ and $t$ such that $r < t$, two states which are $t$-strongly equivalent are also $r$-strongly equivalent, and in particular they are equivalent states.

We consider two states as *strongly-equivalent states* if for every $t \geq 0$ the states are $t$-strongly-equivalent. The meaning of strong equivalence of states is that the endless layer trees going out of $u$ and $u'$ are isomorphic to each other, where the isomorphism preserves the labeling of the edges. In a deterministic graph, two states are equivalent if and only if they are strongly-equivalent. On the other hand, in a nondeterministic graph there may be two states which are equivalent to each other but are not strongly-equivalent. For example, consider the pair of states $u$ and $v$ in Figure 2.4, which are equivalent states but are not strongly-equivalent.

Let $H$ and $H'$ be two graph presentations with labels from the same alphabet, and assume the existence of some specific input tagging $f$ for the graph $H$. Reductions of strongly-equivalent states in $H'$, with respect to the (output) labeling, can turn the graph $H'$ into the graph $H$ if and only if there is a legal tagging $f'$ of $H'$ such that $H'$, tagged according to $f'$ and regarded as a deterministic graph by this tagging, can be reduced to the graph $H$, tagged according to $f$.

The following proposition states that the anticipation of a labeled graph cannot get bigger as a result of reducing strongly-equivalent states.

**Proposition 2.9** Let $H'$ be an irreducible graph presentation and let $H$ be a graph which can be obtained from $H'$ by a reduction of strongly-equivalent states. Suppose that the anticipation of $H'$ is a finite integer $A'$. Then the anticipation $A$ of $H$ is not greater than $A'$.

**Proof:** Suppose to the contrary that $A > A'$. Then there must exist some state $u$ in $H$ and two distinct paths of length $A' + 1$ that start at $u$ at distinct edges, and generate the same word $w$. Let $e_1$ and $e_2$ be the first edges of these two paths and let $v_1$ and $v_2$ be their target states, respectively. Assume that $e_1$ and $e_2$ are both labeled by $w_1$, and let us denote by $w^1$ the word $w$ without its first label $w_1$. 

![Figure 2.4: Equivalent states that are not strongly-equivalent](image-url)
Let \( u' \) be a state of \( H' \) which is reduced to the state \( u \) in \( H \). The states \( u' \) and \( u \) are strongly-equivalent states and, as a result, there are two distinct edges, labeled with \( w_1 \), that go out of \( u' \) towards two different states, \( v'_1 \) and \( v'_2 \). States \( v'_1 \) and \( v'_2 \) are strongly-equivalent to \( v_1 \) and \( v_2 \), respectively. Since \( v_1 \) is equivalent to \( v'_1 \), a path of length \( A' \) generating the word \( w^1 \) must go out of \( v'_1 \). The same argument holds for \( v'_2 \). Hence, two paths of length \( A' + 1 \), generating the word \( w \), go out of the state \( u' \) in \( H' \), but do not begin with the same edge. This contradicts the assumption that \( A' \) is the anticipation of \( H' \). We can conclude that the anticipation \( A \) of \( H \) must be not greater than that of \( H' \).

An expansion of a graph presentation means that each state may be duplicated, including its set of outgoing edges, to a finite number of states, while each incoming edge of an original state can go, instead, into one of its duplicates. The formal definition is as follows:

**Definition 2.2** We say that \( G' \) is an expansion of a graph presentation \( G \) if \( G' \) is obtained from \( G \) by the following two steps:

(a) Every state \( v \in V(G) \) is replaced in \( G' \) with some states \( v_1, \ldots, v_{k_v} \), where \( k_v \geq 1 \).

(b) An edge \( (u \to v) \) in \( G \) is replaced in \( G' \) with the \( k_u \) (\( = k \)) edges \( (u_1 \to v_i), (u_2 \to v_i), \ldots, (u_k \to v_i) \), where each of \( i_1, \ldots, i_k \) can be any one of the elements of \( \{1, \ldots, k_v\} \).

**Observation 2.10** The definition of expansion implies that all the duplicates of a state, which are generated by an expansion operation, are, in fact, strongly-equivalent states. Thus, if a graph \( G' \) is obtained from \( G \) by an expansion operation, then \( G' \) can turn back into \( G \) by a reduction of strongly-equivalent states.

In the opposite direction, if the graph \( G \) is obtained from \( G' \) after reducing all the strongly-equivalent states in \( G' \), then the graph \( G' \) can be obtained from \( G \) by an operation of expansion.

### 2.7.3 Expansion and State-Splitting

The operation of expansion is not equivalent to the operation of a round of state-in-splitting. A round of in-splitting is in fact a special case of expansion where all the terminal states \( v_1, \ldots, v_{k_v} \) from Definition 2.2 are the same state.

It is known that a graph \( G' \) can be obtained from \( G \) by a round of state-in-splitting if and only if it can be obtained from \( G \) by a sequence of basic in-splittings, where a single state is split in each operation. In case of expansions, however, it is not always true that if \( G' \) can be obtained from \( G \) by an operation of expansion, then \( G' \) can also be obtained from \( G \) by a sequence of basic expansions, in which a single state is duplicated in each operation.

For example, consider a pair of states, \( u \) and \( v \), connected with two anti-parallel edges, \( (u \to v) \) and \( (v \to u) \). By a single operation of expansion, in which both \( u \) and \( v \) are
duplicated, it is possible to get a graph which includes two isolated irreducible components as in Figure 2.5. Each of the irreducible components is isomorphic to the original graph. This graph cannot be obtained in two steps of expansion, where only one state is duplicated in each step, as illustrated in Figure 2.6.

**Observation 2.11** Let $H'$ be an irreducible deterministic graph, obtained from another irreducible deterministic graph $H$ by an operation of expansion. A function $f : V(H') \rightarrow V(H)$ can be defined so that every state of $H'$ is mapped to its origin state in $H$. Given an $(A_{H}, n)$-approximate eigenvector, $\bar{x}$, an $(A_{H'}, n)$-approximate eigenvector, $\bar{x}'$, can be defined as follows:

$$\bar{x}'_{v'} = x_{f(v')}; \quad v' \in V(H')$$  \hspace{1cm} (2.1)

The vector $\bar{x}'$ is then considered as an unreduced version of $\bar{x}$.

Now, suppose that after one round of out-splitting of $H$, according to the vector $\bar{x}$, a graph $H^{(1)}$ is obtained with a respective $(A_{H^{(1)}}, n)$-approximate eigenvector $\bar{x}^{(1)}$. Then the graph $H'$ can be split according to the vector $\bar{x}'$, yielding a graph $H'^{(1)}$, which in turn can be obtained from $H^{(1)}$ by an expansion operation. The respective $(A_{H'^{(1)}}, n)$-approximate eigenvector, $\bar{x}'^{(1)}$, is an unreduced version of $\bar{x}^{(1)}$.

One way in which $H'$ can be split to satisfy the above is by imitating the splitting of $H$: The set of edges going out of $v'$ in $H'$ is identical to the set of edges going out of $v = f(v')$ in $H$, in the number of edges, their labels, and the components corresponding to their target states in the vectors $\bar{x}'$ and $\bar{x}$, respectively.
As a result, if $v$ is split into states $v_1, \ldots, v_k$, with respective components $w_1, \ldots, w_k$ in $\pi^{(1)}$, then $v'$ can be split into $v'_1, \ldots, v'_k$, with the same respective components $w_1, \ldots, w_k$ in $\pi'^{(1)}$. Every descendant state $v'_r$ of $v'$ in $H'^{(1)}$ inherits the outgoing edges corresponding to those assigned to the descendant state $v_r$ of $v$ in $H^{(1)}$. It is easy to verify that the resulting graph, $H'^{(1)}$, is indeed an unreduced version of $H^{(1)}$.

**Corollary 2.12** If a graph $H^{(m)}$ is obtained from an irreducible graph $H$ by $m$ rounds of state-out-splitting, with respect to an $(A_H, n)$-approximate eigenvector, $\tilde{x}$, then any unreduced version $H'$ of $H$ can be split in $m$ rounds according to an $(A_H, n)$-approximate eigenvector, $\tilde{x}'$, which is an unreduced version of $\tilde{x}$. The resulting graph $H'^{(m)}$ can be obtained from $H^{(m)}$ by an expansion operation. The resulting $(A_{H^{(m)}}, n)$-approximate eigenvector is an unreduced version of the resulting $(A_{H^{(m)}}, n)$-approximate eigenvector.
Chapter 3

Determinizing Graphs of Encoders

In this chapter, we introduce a construction of a determinizing graph for a given encoder graph. The construction here differs from the construction in [MR91] and [AMR95b] as will be explained in the sequel. The determinizing graph of an encoder \( E \) will be denoted by \( G_E \). Some properties of \( G_E \), mainly related to its splitting capabilities, are summarized in Lemma 3.3 and Lemma 3.5. The proofs of the two main theorems in this work, Theorem 4.1 and Theorem 6.1, are based on these two main lemmas, respectively. As a matter of fact, the proofs of the lemmas are the crucial part of the whole work.

3.1 Construction of a Determinizing Graph \( G_E \)

Let \( S \) be an irreducible constrained system and let \( n \) be a positive integer where \( C(S) \geq \log n \). Let \( E \) be an irreducible \((S, n)\) encoder, and define \( S_E = S(E) \subseteq S \). Let \( A \) be the anticipation of the encoder \( E \). In the determinizing graph construction used in [MR91] and [AMR95b], the states of the determinizing graph are the sets of terminal states of paths in \( E \). In our construction, the states in \( G_E \) are defined to be the sets of the paths themselves. In addition, we concentrate only on paths of length equal to the anticipation \( A \).

Definition 3.1 Let \( E \) be an \((S, n)\) encoder, let \( \pi \) be a path in \( E \) which starts at \( v = \sigma(\pi) \) and terminates in \( \tau(\pi) \), and let \( w \) be a word of \( S_E \). We define \( T_E(\pi, w) \) to be the set of all different paths in \( E \) which start at the prefix \( \pi \) and continue from \( \tau(\pi) \) with suffixes that generate the word \( w \). (Clearly, all the paths in \( T_E(\pi, w) \) start at the same state \( v = \sigma(\pi) \).)

3.1.1 States of \( G_E \) — \( V(G_E) \)

The states of the graph \( G_E \) are all the nonempty sets of paths \( T_E(\pi, w) \) in which

1. \( \pi \) is the empty path that contains only the initial state \( v \) and will be denoted by \( \pi_\phi(v) \), and —
2. \( w \) is of length equaling the anticipation \( A \).

More formally,

\[
V(G_E) = \{ \mathcal{T}_E(\pi, w) : v \in V(\mathcal{E}), \ \pi = \pi_\phi(v), \ w \in S_\mathcal{E}, \ |w| = A, \ \mathcal{T}_E(\pi, w) \neq \phi \} \tag{3.1}
\]

A state \( \mathcal{T}_E(\pi_\phi(v), w) \) in \( G_E \) is actually a collection of \( \mathcal{E} \)-paths of length \( A \) that share their initial state \( v \) and generate the same word of length \( A \), \( w = w_1 w_2 \ldots w_A \). Figure 3.1 (a) illustrates the construction of a state \( Z \), where the parameters of the corresponding encoder \( \mathcal{E} \) are \( n = 2 \) and \( A = 2 \).

3.1.2 Edges of \( G_E \) — \( E(G_E) \)

An edge labeled \( l \) goes out of \( \mathcal{T}_E(\pi_\phi(v), w) \) if and only if the set \( \mathcal{T}_E(\pi_\phi(v), wl) \) is nonempty, i.e., paths of length \( A + 1 \) generating \( wl \) go out of \( v \) in \( \mathcal{E} \). In case \( \mathcal{T}_E(\pi_\phi(v), wl) \) is nonempty, all the paths in this set begin with the same edge \( e_1 \); otherwise, there would exist two paths of length \( A + 1 \) in \( \mathcal{E} \) which would generate the same word \( wl \) and start at the same state, \( v \), at two different edges, thus contradicting the assumption that \( A \) is the anticipation of \( \mathcal{E} \). Let \( e_1 \) be the (common) first edge in the paths of \( \mathcal{T}_E(\pi_\phi(v), wl) \) and let \( v_1 \) be the target state of \( e_1 \). For a word \( z \) of length \( m \), we denote by \( (z)^k \) the suffix of length \( m - k \) of \( z \), so \( (wl)^1 \) is the suffix of \( wl \) of length \( A \). In case where \( \mathcal{T}_E(\pi_\phi(v), wl) \) is nonempty, the set \( \mathcal{T}_E(\pi_\phi(v_1), (wl)^1) \) is also nonempty, and the latter is also a state of \( G_E \). An edge labeled
l goes in \( G_\mathcal{E} \) from \( T_\mathcal{E}(\pi_\phi(v), w) \) to \( T_\mathcal{E}(\pi_\phi(v_1), (wl)^1) \), for every label \( l \) in the alphabet \( \Sigma \). Figure 3.1 (b) illustrates the local neighborhood of state \( Z \) in \( G_\mathcal{E} \), where the parameters of \( \mathcal{E} \) are supposed to be \( n = 2 \) and \( A = 2 \). The numbers inside the states are the weights of the states, as will be explained in the sequel.

### 3.1.3 Basic Properties of \( G_\mathcal{E} \)

**Proposition 3.1** Given an irreducible \((S, \nu)\) encoder \( \mathcal{E} \), the graph \( G_\mathcal{E} \) is a deterministic and irreducible presentation of the system \( S_\mathcal{E} \), i.e., a determinizing graph for the encoder \( \mathcal{E} \).

**Proof:** First notice that in case where \( A = 0 \), we have \( G_\mathcal{E} = \mathcal{E} \); as \( \mathcal{E} \) in this case is a deterministic and irreducible \((S_\mathcal{E}, \nu)\) encoder, Proposition 3.1 becomes trivial. We assume from now on that \( A > 0 \) and show that \( G_\mathcal{E} \) is (a) deterministic, (b) irreducible, and that (c) \( S(G_\mathcal{E}) = S_\mathcal{E} \).

(a) **Deterministic:** This property is implied by the definition of \( G_\mathcal{E} \): since all the \( \mathcal{E} \)-paths that start at \( v \) and generate the word \( wl \) begin with the same edge \( e_1 \), there is only one state \( T_\mathcal{E}(\pi_\phi(v_1), (wl)^1) \) in \( G_\mathcal{E} \) which is accessible from \( T_\mathcal{E}(\pi_\phi(v), w) \) by an edge labeled \( l \). As explained in Section 3.1.2, the uniqueness of the edge \( e_1 \) and its terminal state \( v_1 \) is due to the fact that the length of \( w \) is equal to the anticipation \( A \).

(b) **Irreducibility:** We need to prove that any arbitrary state in \( G_\mathcal{E} \) is connected by a path to any other state in \( G_\mathcal{E} \). Let \( T_\mathcal{E}(\pi_\phi(v), w) \) and \( T_\mathcal{E}(\pi_\phi(v'), w') \) be two arbitrary states in \( G_\mathcal{E} \). Let \( u \) be the terminal state of one of the paths in \( T_\mathcal{E}(\pi_\phi(v), w) \), i.e., there is a path of length \( A \) generating \( w \) from state \( v \) to state \( u \) in \( \mathcal{E} \). As \( \mathcal{E} \) is irreducible, there must also be a path from state \( u \) to state \( v' \), generating some word \( w'' = l_1l_2\ldots l_m \) of length \( m \). By the irreducibility of \( \mathcal{E} \) we can further assume that \( m > A \). The word \( ww''w' \) is then generated in \( \mathcal{E} \) by a path which has the form:

\[
\begin{align*}
  v & \xrightarrow{w} v_1 \xrightarrow{w_2} \ldots \xrightarrow{w_m} v_A = \underbrace{u \xrightarrow{l_1} u_1 \xrightarrow{l_2} \ldots \xrightarrow{l_A} u_A}_{l_{A+1}} \ldots \xrightarrow{l_m} u_m = v' \xrightarrow{w_1'} \xrightarrow{w_2'} \ldots \xrightarrow{w_A'} v_A'
\end{align*}
\]

The existence of this path in \( \mathcal{E} \) implies the existence of a path generating \( w''w' \) in \( G_\mathcal{E} \), where the initial state is \( T_\mathcal{E}(\pi_\phi(v), w) \) and the terminal state is \( T_\mathcal{E}(\pi_\phi(v'), w') \). The path in \( G_\mathcal{E} \) has the form:

\[
\begin{align*}
  T_\mathcal{E}(\pi_\phi(v), w) & \xrightarrow{l_1} T_\mathcal{E}(\pi_\phi(v_1), w_1l_1) \xrightarrow{l_2} \ldots \xrightarrow{l_A} T_\mathcal{E}(\pi_\phi(u), l_1l_2\ldots l_A) \xrightarrow{l_{A+1}} \ldots \xrightarrow{l_m} T_\mathcal{E}(\pi_\phi(u_{m-A}), l_{m-A+1}l_{m-A+2}\ldots l_m) \xrightarrow{w_1'} \ldots \xrightarrow{w_A'} T_\mathcal{E}(\pi_\phi(u_m), w'_1w'_2\ldots w'_{A'}) = T_\mathcal{E}(v', \pi_\phi)w'.
\end{align*}
\]

(Remark: As we assume \( A > 0 \), we can write \( w_1l_1 \) instead of \( (wl_1)^1 \).)

(c) \( S(G_\mathcal{E}) = S_\mathcal{E} \): We show that a word \( w' \) is generated by \( G_\mathcal{E} \) iff it belongs to \( S_\mathcal{E} \), i.e., generated by \( \mathcal{E} \). Both \( \mathcal{E} \) and \( G_\mathcal{E} \) are irreducible graphs. Therefore, if a word \( w' \) of length smaller than \( A \) is generated by one of these graphs, then another word of length
\( \mathcal{A} \) or more that contains \( w' \) is also generated by this graph and vice versa. Hence, we can consider in our proof only words of length \( \mathcal{A} \) or more. Suppose that a word \( w' = l_1 l_2 \ldots l_m \) of length \( m \geq \mathcal{A} \) is generated by \( G_E \). This means that there must be a state \( v \) in \( E \) and a word \( w = w_1 w_2 \ldots w_\mathcal{A} \) of length \( \mathcal{A} \) in \( S_E \), such that \( w' \) is generated in \( G_E \) from the state \( Z = \mathcal{T}_E(\pi_v(v), w) \). The path generating \( w' \) in \( G_E \) has the form:

\[
\begin{align*}
\mathcal{T}_E(\pi_v(v), w) & \xrightarrow{l_1} \mathcal{T}_E(\pi_v(v_1), w_1 l_1) \xrightarrow{l_2} \cdots \xrightarrow{l_\mathcal{A}} \mathcal{T}_E(\pi_v(v_\mathcal{A}), l_1 l_2 \ldots l_\mathcal{A}) \xrightarrow{l_{\mathcal{A}+1}} \\
\mathcal{T}_E(\pi_v(v_1), l_2 l_3 \ldots l_{\mathcal{A}+1}) & \xrightarrow{l_{\mathcal{A}+2}} \cdots \xrightarrow{l_m} \mathcal{T}_E(\pi_v(u_m - \mathcal{A}), l_m - \mathcal{A} + 1 l_m - \mathcal{A} + 2 \ldots l_m)
\end{align*}
\]

(3.2)

The existence of such a path in \( G_E \) implies, in turn, the existence of a path in the encoder \( E \) which has the form:

\[
v \xrightarrow{w_1} v_1 \xrightarrow{w_2} \cdots \xrightarrow{w_\mathcal{A}} v_\mathcal{A} \xrightarrow{l_1} u_1 \xrightarrow{l_2} \cdots \xrightarrow{l_\mathcal{A}} u_\mathcal{A} \xrightarrow{l_{\mathcal{A}+1}} u_{\mathcal{A}+1} \xrightarrow{l_{\mathcal{A}+2}} \cdots \xrightarrow{l_m} u_m
\]

Thus, the word \( ww' \) — and so the word \( w' \) — are generated by paths in \( E \). As the word \( w' \) belongs to \( S_E \), we get \( S(G_E) \subseteq S_E \). On the other hand, suppose that the word \( w' \) of length \( m \) is generated in \( E \) by a path which starts at state \( v \). As \( E \) is an irreducible graph, there is a path of length \( \mathcal{A} \) in \( E \) that starts at some state \( v \), generates some word \( w = w_1 w_2 \ldots w_\mathcal{A} \) and terminates in state \( u \). The path generating \( ww' \) in \( E \) has the form of (3.3), where the state \( v_\mathcal{A} \) is now state \( u \). A path generating \( w' \) then starts in \( G_E \) in the state \( \mathcal{T}_E(\pi_v(v), w) \) and has the form of (3.2). So \( S_E \subseteq S(G_E) \), and therefore \( S(G_E) = S_E \).

\[\blacksquare\]

### 3.1.4 Perron Eigenvector of \( G_E \)

The Shannon capacity of the system \( S_E = S(E) \) is exactly \( \log n \), as \( E \) is a lossless irreducible regular graph of outdegree \( n \) (see Section 2.3). Since \( G_E \) is also an irreducible presentation of \( S_E \), \( n \) is the Perron eigenvalue of the matrix \( A_{G_E} \). One of the eigenvectors of \( A_{G_E} \), associated with the eigenvalue \( n \), can be constructed as follows: We define the weight of a state \( Z = \mathcal{T}_E(\pi_v(v), w) \) in \( G_E \) to be the number of \( E \)-paths associated with \( Z \). Due to the losslessness of \( E \), the weight \( |Z| \) is also the number of distinct states in \( E \) which are accessible from \( v \) by paths generating the word \( w \). The weights of the states in Figure 3.1 (b) are written inside the states. The vector \( \bar{e}_E = [(e_E)Z]_{Z \in V(G_E)} \) is defined by:

\[
(e_E)Z = |Z| \quad ; \quad Z \in V(G_E)
\]

(3.4)

Notice that the components of \( \bar{e}_E \) are all positive integer numbers.

**Proposition 3.2** The vector \( \bar{e}_E \), which is defined by (3.4), is an eigenvector of the matrix \( A_{G_E} \), associated with the eigenvalue \( n \).

**Proof:** Let \( Z = \mathcal{T}_E(\pi_v(v), w) \) be some state in \( G_E \). Since the outdegree of \( E \) is \( n \), the number of edges in \( E \) that go out of the terminal states of \( Z \) is exactly \( n |Z| \). We denote by
Every $E_l$-path of $Z_l$ terminates with an edge of $E_l$, and every edge of $E_l$ is a terminal edge of exactly one path of $Z_l$, because of the losslessness of $E$. Therefore, the number of $E_l$-paths in $Z_l$, which is defined to be the weight $|Z_l|$, is equal to the number of edges in $E_l$. Summing up the weights of the neighbors $Z_l$ over all the labels $l$ in the alphabet $\Sigma$, we get:

$$\begin{align*}
(A_{E_l}\bar{\varepsilon}_E)Z &= \sum_{Z' \in V(G_{E_l})} (A_{E_l})Z, Z'(\varepsilon E)Z' \\
&= \sum_{Z' \in V(G_{E_l})} (A_{E_l})Z, Z'|Z'| = \sum_{l \in \Sigma; Z_l \neq \emptyset} |Z_l| \\
&= \sum_{l \in \Sigma} |E_l| = n |Z| = n(\varepsilon E)Z
\end{align*}$$

### 3.2 Splitting of $G_{E_l}$

In this section, we present two main lemmas that deal with two significant splitting properties of the determinizing graph $G_{E_l}$. These two lemmas are the basis for the proofs of Theorem 4.1 and Theorem 6.1.

#### 3.2.1 Lemma 3.3 — Splitting of $G_{E_l}$ in $A$ Rounds

**Lemma 3.3** Let $E$ be an irreducible $(S_E, n)$ encoder, where $S(E) = S_E$. Suppose that the anticipation of $E$ is a finite integer $A$, and assume some legal (input) tagging function $f : E(E) \to \{0, \ldots, n-1\}$. Let $G_{E_l}$ be the irreducible determinizing graph of $E$ as defined in Section 3.1 and let $\bar{\varepsilon}_E$ be the eigenvector of $G_{E_l}$, associated with $n$, which is defined by (3.4). The graph $G_{E_l}$ and its eigenvector $\bar{\varepsilon}_E$ satisfy the following:

1. The largest component of $\bar{\varepsilon}_E$ is not greater than $n^A$.
2. The graph $G_{E_l}$ can be split consistently with $\bar{\varepsilon}_E$ in $A$ rounds of splitting, until the all-one vector is obtained. The resulting graph is an $(S_E, n)$ encoder, $E'$, whose anticipation is also $A$.
3. In each of the splitting rounds, every state is split into $n$ states at most.
4. In the $i$th round of splitting, each component of the resulting eigenvector, $\bar{\varepsilon}_E^{(i)}$, is not greater than $n^{A-i}$.
5. The encoder $E$ can be obtained from $E'$ by a reduction of strongly equivalent states. In other words, there is a legal tagging function $f' : E(E') \to \{0, \ldots, n-1\}$ such that the tagged $E'$, treated as a deterministic graph according to the labeling by pairs of input tag and output label, can be reduced to the original encoder $E$, tagged according to $f$.

The proof of Lemma 3.3 is rather long and is given in Section 3.2.3.
3.2.2 Lemma 3.5 - Splitting of $G_E^k$ in One Round

Lemma 3.5 below is a corollary of Lemma 3.3 and will serve as our second main lemma. We first state and prove the following simple result:

**Lemma 3.4** Let $A$ be a nonnegative integer matrix and let $n$ be a positive integer. Then every $(A,n)$-approximate eigenvector is also an $(A^k,n^k)$-approximate eigenvector for every positive integer $k$.

**Proof:** The proof of Lemma 3.4 is by induction on $k$. The basis where $k = 1$ is obvious. Suppose that $\bar{\mathbf{x}}$ is an $(A^k,n^k)$-approximate eigenvector for $k = l$, i.e.,

$$A^l \bar{\mathbf{x}} \geq n^l \cdot \bar{\mathbf{x}}$$  \hspace{1cm} (3.6)

For $k = l + 1$ we have

$$A^{l+1} \bar{\mathbf{x}} = A \cdot A^l \bar{\mathbf{x}} \geq A(n^l \bar{\mathbf{x}}) = n^l A \bar{\mathbf{x}} \geq n^l \cdot n \bar{\mathbf{x}} = n^{l+1} \bar{\mathbf{x}}$$  \hspace{1cm} (3.7)

Thus, $\bar{\mathbf{x}}$ is an $(A^{l+1},n^{l+1})$-approximate eigenvector, and we conclude that for every integer $k$ greater than zero, the vector $\bar{\mathbf{x}}$ is an $(A^k,n^k)$-approximate eigenvector. \hfill \blacksquare

We now state Lemma 3.5 which describes the splitting capabilities of the power graphs $G_E^k$ according to the $(A_G^k,n^k)$ (true) eigenvector $\bar{\mathbf{e}}_\mathcal{G}$.

**Lemma 3.5** Let $\mathcal{E}$, $n_\mathcal{E}$, $n$, $A$, $G_\mathcal{E}$, and $\bar{\mathbf{e}}_\mathcal{E}$ be as in Lemma 3.3. For every integer $k$, where $1 \leq k \leq A$, there is a $\bar{\mathbf{e}}_\mathcal{E}$-consistent round of splitting of $G_\mathcal{E}^k$ with respect to the $(A_G^k,n^k)$ (true) eigenvector $\bar{\mathbf{e}}_\mathcal{G}$, such that every state of $G_\mathcal{E}^k$ is split into no more than $n^k$ states, and each component of the resulting eigenvector is not greater than $n^{A-k}$.

The proof of Lemma 3.5 is immediate from Lemma 3.3, using Lemma 3.4 and Proposition 3.6 given right below. Proposition 3.6 describes the connection between the splitting capabilities of a graph $G$ and the splitting capabilities of a power graph of $G$.

**Proposition 3.6** Suppose that an irreducible graph $G$ can be split in $k$ rounds consistently with an $(A_G,n)$-approximate eigenvector $\bar{\mathbf{x}}$, yielding a graph $H$ and a vector $\bar{\mathbf{y}}$. Then the graph $G^k$ can be split in one round consistently with the vector $\bar{\mathbf{x}}$, yielding the graph $H^k$ and the vector $\bar{\mathbf{y}}$.

**Proof:** By Lemma 3.4 and by the identity $(A_G)^k = A_G^k$, the $(A_G,n)$-approximate eigenvector $\bar{\mathbf{x}}$ is an $(A_G^k,n^k)$-approximate eigenvector. Given that $G$ can be split in $k$ rounds according to $\bar{\mathbf{x}}$, yielding the graph $H$, we now describe a way in which $G^k$ can be split in one round consistently with $\bar{\mathbf{x}}$.

Since the result depends only on the underlying unlabeled graph $G$, we can re-label $G$ so that each of its edges is labeled with a distinct label, representing the name of the edge. Using this labeling, the graph $G$ becomes a deterministic graph.
Let $u$ be a state of $G$ (and of $G^k$), and suppose that the graph $H$ contains $m$ descendant states of $u$, say $u_1, \ldots, u_m$. Then state $u$, regarded as a state of $G^k$, can be split in one round into $m$ descendant states, $u'_1, \ldots, u'_m$. Let $e$ be an outgoing edge of $u$ in $G^k$, carrying the label $l_1 l_2 \ldots l_k$, which means that a path of length $k$, generating the word $l_1 l_2 \ldots l_k$, goes out of $u$ in $G$. The edge $e$ is inherited in this case by the descendant state $u'_j$ if and only if a path generating $l_1 l_2 \ldots l_k$ goes out of $u_j$ in $H$.

The distribution of edges described above implies that the graph obtained from $G^k$ is indeed $H^k$ and therefore the vector $\vec{y}$ is indeed its approximate eigenvector. The only thing that remains to be proved is that the way suggested for distributing the edges of $G^k$ satisfies the definition of state splitting, i.e., a partition is defined over the edges of each state in $G^k$. In order to prove that, it is enough to prove that every path $\pi$ of length $k$ that goes out of a state $u$ in $G$ is inherited by a single descendent state of $u$ in $H$.

We prove by induction on $k$ that if a path generating $l_1 l_2 \ldots l_k$ goes out of $u$ in $G$, then paths generating $l_1 l_2 \ldots l_k$ go out of a single descendant state of $u$ in $H$. By the properties of state-splitting, this is true for $k = 1$. Assume now the correctness for $k - 1$. Let $H'$ be the graph obtained from $G$ by $k - 1$ rounds and let $H$ be the graph obtained from $H'$ by one more round. Let $u_j$ be the unique descendant of $u$ in $H'$ which is an initial state of paths that generate $l_1 l_2 \ldots l_{k-1}$. It is clear then that $u_j$ is also the only descendant state of $u$ in $H'$ which is an initial state of paths that generate $l_1 l_2 \ldots l_k$.

Suppose to the contrary that $H$ contains two descendant states $u_{j_1}$ and $u_{j_2}$ of $u$, both being initial states of paths in $H$ that generate $l_1 l_2 \ldots l_k$. Then $u_{j_1}$ and $u_{j_2}$ must be generated by the splitting of $u_j$ in round $k$. Since the graph $G$ is irreducible, the graph $H'$ is also irreducible, and there must be some incoming edge of $u_j$ in $H'$, $e = (v \rightarrow u_j)$, which carries some label $l_0$. Let $v_{j_1}$ be the descendant state of $v_j$ in $H$ which inherits this edge $e$. Then $v_{j_1}$ has two outgoing edges labeled by $l_0$ that go into $u_{j_1}$ and $u_{j_2}$. As a result, two paths of length $k + 1$, which generate $l_0 l_1 \ldots l_k$, go out of $v_{j_1}$ in $H$ and they do not start with the same first edge. This contradicts the fact that the anticipation of $H$ should not be greater than $k$.

\[ \Box \]

### 3.2.3 Proof of Lemma 3.3

Before giving the formal proof of Lemma 3.3, we describe intuitively the way in which the graph $G_{\mathcal{E}}$ can be split as claimed. Let $Z = T_\mathcal{E}(\pi_{\phi}(v), w)$ be some state in $G_{\mathcal{E}}$. A splitting of $Z$ in our case will mean a partition of the set of $\mathcal{E}$-paths associated with $Z$, i.e., the paths which start at $v$ and generate the word $w$. Let $e_1, \ldots, e_m$ be the edges that appear as a first edge in some path of $Z$, and let $v_1, \ldots, v_m$ be their corresponding target states. The paths of $Z$ are then divided into $m$ subsets, where the $j$th subset contains all the paths of $Z$ that start with the edge $e_j$.

Each subset of paths defines a descendant state of $Z$. The $j$th descendant state, which corresponds to the $j$th subset of paths, will therefore be denoted by $T_\mathcal{E}(e_j, w^1)$. Such a descendant state inherits all the outgoing edges of $Z$ that go to states of the form $T_\mathcal{E}(\pi_{\phi}(v_j), w^1 l)$, for all possible labels $l$. This distribution of the edges satisfies the definition of a state splitting, since it implies a partition of the outgoing edges of $Z$. The
number of edges $e_1, \ldots, e_m$ cannot be greater than the outdegree $n$ of $v$. Therefore, state $Z$ has no more than $n$ descendant states. Figure 3.2 illustrates the splitting of a state $Z$ in the first round, where $n = 2$ and $A = 2$. The dashed edges symbolize the distribution of the edges of $Z$ between its two descendant states. The numbers inside the states represent the weights of the states, as explained in the sequel.

Continuing in a similar manner, the descendant states of $Z$ after the $i$th round of splitting will correspond to subsets of paths in $Z$, where the paths in each of the subsets share their first $i$ edges. Such a state will therefore be denoted by $T_{\mathcal{E}}(\pi, w^i)$, where the path $\pi$ of length $i$ generates the prefix (of length $i$) of $w$. After $A$ rounds of splitting, each descendant state of $Z$ will correspond to a single $\mathcal{E}$-path that generates the word $w$. Such a descendant state will be denoted by $T_{\mathcal{E}}(\pi_w, w_{\phi})$. It turns out that such a descendant state is strongly equivalent to the terminal $\mathcal{E}$-state of $\pi_w$, and the whole split graph can therefore be reduced to the original encoder, $\mathcal{E}$.

The rest of this section is devoted to the proof of Lemma 3.3, through a sequence of definitions and propositions.

In case where $A = 0$, the proof of this lemma is straightforward, as $G_{\mathcal{E}}$ is the graph $\mathcal{E}$ and thus can turn “in zero rounds” into an $(S_{\mathcal{E}}, n)$ encoder. All the other properties mentioned in Lemma 3.3 are also obvious in case that $A = 0$. So we assume from now on that the anticipation of $\mathcal{E}$ is greater than zero. In order to prove the lemma, we define a sequence of graphs $\{G_{\mathcal{E}}^{(i)}\}_{i=0}^A$ as follows:

The states of the graph $G_{\mathcal{E}}^{(i)}$ are defined as all the nonempty sets $T_{\mathcal{E}}(\pi, w)$ of $\mathcal{E}$-paths in
which \( \pi \) is of length \( i \) while \( w \) is of length \( A - i \):

\[
V(G_e^{(i)}) = \{ T_\mathcal{E}(\pi, w) : \ v \in V(\mathcal{E}), \ v = \sigma(\pi), \ |\pi| = i, \ w \in S_\mathcal{E}, \ |w| = A - i, \ T_\mathcal{E}(\pi, w) \neq \phi \}
\]

(3.8)

A state \( T_\mathcal{E}(\pi, w) \) of \( G_e^{(i)} \) thus consists of the \( \mathcal{E} \)-paths of length \( A \) that start at a common prefix \( \pi \) of length \( i \), and then continue from \( \tau(\pi) \) with suffixes of length \( A - i \) that generate the word \( w \).

An edge labeled \( l \) goes out of a state \( Z = T_\mathcal{E}(\pi, w) \) in \( G_e^{(i)} \) iff the set \( T_\mathcal{E}(\pi, w) \) of paths is nonempty. All the paths of \( T_\mathcal{E}(\pi, w) \) share their first edge, even when \( i = 0 \), because the anticipation of \( \mathcal{E} \) is \( A \). Let \( e_1 \) be the first (common) edge in the paths of \( T_\mathcal{E}(\pi, w) \). Note that when \( i > 0 \), the edge \( e_1 \) is simply the first edge in the common prefix \( \pi \) of the paths in \( T_\mathcal{E}(\pi, w) \). Edges labeled \( l \) go in \( G_e^{(i)} \) from \( T_\mathcal{E}(\pi, w) \) to all states \( T_\mathcal{E}(\pi_1, (wl)^1) \), where \( \pi_1 \) ranges over all paths of length \( i \) in \( \mathcal{E} \) such that

1. \( e_1 \pi_1 \) is a path equaling \( \pi e_{i+1} \) for some edge \( e_{i+1} \);

2. the label \( L(e_{i+1}) \) of \( e_{i+1} \) in \( \mathcal{E} \) is the first label in \( wl \).

**Proposition 3.7** The first graph in the sequence, \( G_\mathcal{E}^{(0)} \), is the graph \( G_\mathcal{E} \), as defined in Section 3.1.

**Proof:** To verify that \( G_\mathcal{E} \) is \( G_\mathcal{E}^{(0)} \), recall that the states of \( G_\mathcal{E} \) are all the nonempty sets of paths \( T_\mathcal{E}(\pi_\mathcal{E}(v), w) \) where \( \pi_\mathcal{E}(v) \) is of length 0 and \( w \) is of length \( A = A - 0 \). Hence, the states of \( G_\mathcal{E} \) are those of \( G_\mathcal{E}^{(0)} \). The definitions of edges in \( G_\mathcal{E} \) and \( G_\mathcal{E}^{(0)} \) are also identical: notice that in the case of \( i = 0 \), the edge \( e_1 \) is the same edge as \( e_{i+1} \). This edge is not an edge of \( \pi \) but rather it is the first edge in a path generating \( w \). The path \( \pi_1 \) in this case is also an empty path, "starting" at \( \tau(e_1) \).

**Proposition 3.8** The last graph in the sequence, \( G_\mathcal{E}^{(A)} \), is an \((S_\mathcal{E}, n)\) encoder of anticipation \( A \) that can be turned into the original encoder \( \mathcal{E} \) by a reduction of strongly equivalent states.

**Proof:** Considering the graph \( G_\mathcal{E}^{(A)} \), its states are all the nonempty sets \( T_\mathcal{E}(\pi, w_\mathcal{E}) \) in which \( \pi \) is of length \( A \) and \( w_\mathcal{E} \) is the empty word. A state \( Z = T_\mathcal{E}(\pi, w_\mathcal{E}) \) in this case consists of the single path \( \pi \) that starts at \( v = \sigma(\pi) \) and ends in \( \tau(\pi) \). There is a one-to-one correspondence between the outgoing edges of \( Z \) in \( G_\mathcal{E}^{(A)} \) and the outgoing edges of \( \tau(\pi) \) in \( \mathcal{E} \): An edge from \( Z = T_\mathcal{E}(\pi, w_\mathcal{E}) \) to \( Z' = T_\mathcal{E}(\pi_1, w_\mathcal{E}) \) corresponds to the unique edge \( e_{A+1} \) outgoing from \( \tau(\pi) \) in \( \mathcal{E} \) such that \( \pi_1 \) is the suffix of length \( A \) of \( \pi e_{A+1} \). Let \( Z_l \) be a state in \( G_\mathcal{E}^{(A)} \), which is accessed from \( Z \) by an edge labeled \( l \) (there may be several such states). State \( Z_l \) consists of a single path whose terminal \( \mathcal{E} \)-state is accessed from \( \tau(\pi) \) by an edge labeled \( l \).

Suppose we rename the states of \( G_\mathcal{E}^{(A)} \) by replacing each \( T_\mathcal{E}(\pi, w_\mathcal{E}) \) with \( \tau(\pi) \). According to the observations above, if a state in \( G_\mathcal{E}^{(A)} \) is renamed as the \( \mathcal{E} \)-state \( u \), then its local outgoing neighborhood looks the same as the local neighborhood of \( u \) in \( \mathcal{E} \). Thus, the outdegree of every state in \( G_\mathcal{E}^{(A)} \) is \( n \). Due to the irreducibility of \( \mathcal{E} \), every state of \( \mathcal{E} \) is a
terminal state of some path of length \mathcal{A}. Thus, for every state \( u \) of \( \mathcal{E} \), there exists some state \( Z = T_{\mathcal{E}}(\pi, w) \) in \( G^{(1)}_{\mathcal{E}} \) for which \( u = \tau(\pi) \) and therefore \( Z \) is renamed \( u \). Consequently, after the renaming, every \( u \in V(\mathcal{E}) \) appears in \( G^{(A)}_{\mathcal{E}} \), with its whole local neighborhood exactly as in \( \mathcal{E} \).

However, an \( \mathcal{E} \)-state \( u \) might appear in the renamed \( G^{(A)}_{\mathcal{E}} \) more than once, as \( u \) might be a terminal state of several different paths of length \( \mathcal{A} \) in \( \mathcal{E} \), i.e., \( u \) might be \( \tau(\pi) \) in more than one \( T_{\mathcal{E}}(\pi, w) \). The graph \( G^{(A)}_{\mathcal{E}} \) is thus an unreduced version of \( \mathcal{E} \), and \( G^{(A)}_{\mathcal{E}} \) is an \((S_{\mathcal{E}}, u)\) encoder. All the states of \( G^{(A)}_{\mathcal{E}} \) that are equivalent to one \( \mathcal{E} \)-state, \( u \), are actually strongly equivalent states, as their local neighborhoods are all identical to that of \( u \) in \( \mathcal{E} \). Consequently, \( G^{(A)}_{\mathcal{E}} \) is an \((S_{\mathcal{E}}, u)\) encoder that can be turned into the original encoder \( \mathcal{E} \) by a reduction of strongly equivalent states.

It is left to show that the anticipation of \( G^{(A)}_{\mathcal{E}} \) is also \( \mathcal{A} \). Due to Proposition 2.9, the anticipation cannot get bigger while reducing strongly equivalent states, so the anticipation of \( G^{(A)}_{\mathcal{E}} \) cannot be smaller than the anticipation of \( \mathcal{E} \) which is \( \mathcal{A} \). In the sequel we prove that \( G^{(A)}_{\mathcal{E}} \) is obtained from \( G_{\mathcal{E}} \) by \( \mathcal{A} \) rounds of state splitting, which implies that the anticipation of \( G^{(A)}_{\mathcal{E}} \) is also not greater than \( \mathcal{A} \), and therefore it is exactly \( \mathcal{A} \).

**Proposition 3.9** For \( i = 1, 2, \ldots, \mathcal{A} \), the graph \( G^{(i)}_{\mathcal{E}} \) can be obtained from the graph \( G^{(i-1)}_{\mathcal{E}} \) by one round of state splitting. In the \( i \)th round of splitting, \( 1 \leq i \leq \mathcal{A} \), every state of \( G^{(i-1)}_{\mathcal{E}} \) is split into no more than \( n \) states.

**Proof:** Let \( Z = T_{\mathcal{E}}(\pi, w) \) be a state of \( G^{(i-1)}_{\mathcal{E}} \). Since \( i - 1 < \mathcal{A} \), the word \( w \) has at least one label, and each of the paths in \( Z \) has at least one more edge following the prefix \( \pi \). Let \( E_{Z}^{i} \) be the set of \( i \)th edges of paths in \( Z \). All these edges go out of \( \tau(\pi) \) in \( \mathcal{E} \), and all are labeled by the first label of \( w \). The state \( Z = T_{\mathcal{E}}(\pi, w) \) of \( G^{(i-1)}_{\mathcal{E}} \) is replaced in \( G^{(i)}_{\mathcal{E}} \) by \( |E_{Z}^{i}| \) descendant states which are defined by:

\[
descendant(Z = T_{\mathcal{E}}(\pi, w)) = \{ T_{\mathcal{E}}(\pi e, w) : e \in E_{Z}^{i} \} \tag{3.9}
\]

A state \( T_{\mathcal{E}}(\pi e, w) \) of \( G^{(i)}_{\mathcal{E}} \) is therefore a descendant state of \( T_{\mathcal{E}}(\pi, L(e)w) \).

Due to the previous definition of states in the graphs \( \{G^{(i)}_{\mathcal{E}}\} \), it is clear that if \( T_{\mathcal{E}}(\pi, w) \) is a state of \( G^{(i-1)}_{\mathcal{E}} \), then \( T_{\mathcal{E}}(\pi e, w) \) is nonempty and therefore it is a state of \( G^{(i)}_{\mathcal{E}} \), whenever \( e \in E_{Z}^{i} \). Conversely, if \( T_{\mathcal{E}}(\pi e, w) \) is a state of \( G^{(i)}_{\mathcal{E}} \), then the word \( L(e)w \) is generated from \( \tau(\pi) \) and, as a result, \( T_{\mathcal{E}}(\pi, L(e)w) \) is nonempty and must be a state of \( G^{(i-1)}_{\mathcal{E}} \). Thus, all the states of \( G^{(i)}_{\mathcal{E}} \) can be obtained as descendant states of states in \( G^{(i-1)}_{\mathcal{E}} \). We see that in the transition from \( V(G^{(i-1)}_{\mathcal{E}}) \) to \( V(G^{(i)}_{\mathcal{E}}) \), the path \( \pi \) gets longer while the word \( w \) gets shorter. Every edge that appears as an \( i \)th edge of a path in \( Z \in V(G_{\mathcal{E}}) \) defines a single descendant state of \( Z \) in \( G^{(i)}_{\mathcal{E}} \).

We now identify the partitioning and assignment of the edges among the descendant states. Every edge that goes in \( G^{(i-1)}_{\mathcal{E}} \) from \( T_{\mathcal{E}}(\pi, w) \) to \( T_{\mathcal{E}}(\pi_1, w_1) \) is inherited by the descendant state \( T_{\mathcal{E}}(\pi e, w_1) \), where \( e \) is the last edge of \( \pi_1 \). Clearly, this defines a partition
of the outgoing edges of $Z$ in $G_{E}^{(i-1)}$. It remains to prove that this partition of the edges implies the previous definition of edges in $G_{E}^{(i)}$.

Let $Z = T_{E}(\pi, w)$ be a state in $G_{E}^{(i-1)}$ and let $Y = T_{E}(\pi e, w^{1})$ be one of its descendant states in $G_{E}^{(i)}$. The edge $e$ goes out of $\tau(\pi)$ in $E$ and its label is the first label of $w$. Suppose that the state $Y$ inherited an edge labeled $l$ from $Z$ to $Z' = T_{E}(\pi_{1}, w^{1}l)$, where $\pi_{1}$ is obtained from $\pi$ by adding the edge $e$ as the last edge and then deleting the first edge. The state $Z'$ is also split in the $i$th round, and its descendant states have the form $\{Y'\} = \{T_{E}(\pi_{1}e', (w^{1}l)^{1})\}$, for the various edges $e'$ that appear as the $(i+1)$-st edge in the paths $T_{E}(\pi, wl)$. Hence, after the round of splitting of $G_{E}^{(i-1)}$, edges labeled $l$ go from the state $Y = T_{E}(\pi e, w^{1})$ to each of the states $Y' = T_{E}(\pi_{1}e', (w^{1}l)^{1})$, for all possible edges $e'$ that define a state $Y' = T_{E}(\pi_{1}e', (w^{1}l)^{1})$ in $G_{E}^{(i)}$. As the path $\pi_{1}e'$ is obtained by appending an edge $e'$ to $\pi e$ and then deleting the first edge of the resulting path, we conclude that the splitting assigns to $Y$ exactly the set of edges it has in $G_{E}^{(i)}$.

Notice that the splitting of a state $Z$ in $G_{E}^{(i-1)}$ does not depend on the splitting of any other state in $G_{E}^{(i-1)}$ and, therefore, all the states of $G_{E}^{(i-1)}$ can be split simultaneously in one round. Hence, the graph $G_{E}^{(i)}$ can be obtained from $G_{E}^{(i-1)}$ in one round of state splitting. In addition, the definition of the splitting process of $G_{E}^{(i-1)}$ implies that each state of $G_{E}^{(i-1)}$ is split into $n$ states at most. This is due to the fact that the number of edges $\{e\}$ that go out of the state $\tau(\pi)$ cannot be greater than the outdegree $n$ of $\tau(\pi)$ in $E$.

We can conclude that in $A$ rounds of state splitting, the graph $G_{E}$, which is an irreducible deterministic presentation of $S_{E}$, can become an $(S_{E}, n)$ encoder $G_{E}^{(A)}$. The original encoder $E$ can be obtained from $G_{E}^{(A)}$ by a reduction of strongly equivalent states. In the $i$th round of splitting, $1 \leq i \leq A$, every state of the graph $G_{E}^{(i-1)}$ is split into $n$ states at most.

According to the definition of $G_{E}^{(i)}$, the states of this graph are sets of $E$-paths. We define the weight of a state $Z = T_{E}(\pi, w)$ in $G_{E}^{(i)}$ to be the number of $E$-paths associated with $Z$. Due to the losslessness of $E$, the weight $|Z|$ is equal to the number of distinct $E$-states that are terminal states of the paths of $T_{E}(\pi, w)$. The vector $\widetilde{e}_{E}^{(i)} = [(e_{E}^{(i)})_{Z}]_{Z \in \mathcal{V}(G_{E}^{(i)})}$ is defined by the weights of the states in $G_{E}^{(i)}$ in the following manner:

$$ (e_{E}^{(i)})_{Z} = |Z| ; \quad Z \in \mathcal{V}(G_{E}^{(i)}) \quad (3.10) $$

By Proposition 3.7 and Equation (3.4), the vector $\widetilde{e}_{E}^{(0)}$ is equal to the eigenvector $\bar{e}_{E}$ associated with the Perron eigenvalue $n$ of the adjacency matrix $A_{G_{E}} = A_{G_{E}^{(0)}}$. The vector $\tilde{e}_{E}^{(A)}$ happens to be the all-one vector, as in $G_{E}^{(A)}$ every state corresponds to a single path in $E$. Since $G_{E}^{(A)}$ is an $(S_{E}, n)$ encoder, the all-one vector, $\bar{I}$, is an eigenvector of $A_{G_{E}^{(A)}}$, associated with the eigenvalue $n$. Thus, in the case of $G_{E}^{(A)}$ we can write:

$$ A_{G_{E}^{(A)}} \cdot \tilde{e}_{E}^{(A)} = n \cdot \bar{I} = n \cdot \tilde{e}_{E}^{(A)} \quad (3.11) $$
Proposition 3.10 The components of \( \mathbf{e}_\mathcal{E}(i) \) are not greater than \( n^{A-i} \), for \( 0 \leq i \leq A \).

Proof: Let \( Z = T_{\mathcal{E}}(\pi, w) \) be a state in \( G_{\mathcal{E}}^{(i)} \). All the paths in \( Z \) share their first \( i \) edges, and then they may separate at the \((i+1)\)st state. The outdegree \( n \) of \( \mathcal{E} \) implies that \( Z \) contains no more than \( n^{A-i} \) paths. Therefore, \( |Z| \leq n^{A-i} \), for every state \( Z \) of \( G_{\mathcal{E}}^{(i)} \). □

Proposition 3.11 The vector \( \mathbf{e}_\mathcal{E}(i) \), which is defined by (3.10), is an eigenvector of the matrix \( A_{G_{\mathcal{E}}^{(i)}} \), associated with the eigenvalue \( n \).

Proof: Since we have already proved the claim for the cases \( i = 0 \) (Proposition 3.2) and \( i = A \), we assume from now on in the proof that \( 0 < i < A \). Consequently, if \( Z = T_{\mathcal{E}}(\pi, w) \) is a state of \( G_{\mathcal{E}}^{(i)} \), then the path \( \pi \) has at least one edge and the word \( w \) is not the empty word.

Let \( Z = T_{\mathcal{E}}(\pi, w) \) be some state in \( G_{\mathcal{E}}^{(i)} \). We denote by \( E_Z \) the set of edges in \( \mathcal{E} \) that can extend the paths of \( Z \) by 1. Recall that all the terminal states of the paths in \( Z \) are distinct due to losslessness. The number of edges in \( E_Z \) is exactly \( n|Z| \), as the outdegree of states in \( \mathcal{E} \) is \( n \). We now classify the paths in \( Z \) and the edges \( E_Z \) according to the edges \( \{e\} \) that follow \( \pi \) in the paths of \( Z \). Consider all the paths in \( Z \) that share a prefix of length \( i+1 \) which is composed of the path \( \pi \) and one additional edge \( e \). We denote by \( E_{l,e} \) the edges of \( E_Z \) that extend those paths and are labeled with \( l \).

Let \( \pi_1 \) be the path obtained by appending the edge \( e \) to \( \pi \) and then truncating the first edge of \( \pi \). The set of \( \mathcal{E} \)-edges \( E_{l,e} \) defines a single edge labeled \( l \) that goes in \( G_{\mathcal{E}}^{(i)} \) from \( Z = T_{\mathcal{E}}(\pi, w) \) to the state \( Z_{l,e} = T_{\mathcal{E}}(\pi_1, w^{l}) \). Hence, \( E_{l,e} \) is the set of terminal \( \mathcal{E} \)-edges of the paths in \( Z_{l,e} \). The number of edges \( E_{l,e} \) is thus equal to the number of paths in \( Z_{l,e} \), namely, \( |Z_{l,e}| \). Note that every neighbor of \( Z \) can be considered as \( Z_{l,e} \), for some label \( l \) and some edge \( e \) which follows \( \pi \). Hence, summing up the weights of the neighbors \( Z_{l,e} \), over all the relevant edges \( e \) and over all the labels \( l \) in the alphabet \( \Sigma \), we get:

\[
(A_{G_{\mathcal{E}}^{(i)}}, \mathbf{e}_\mathcal{E}(i))_Z = \sum_{l,e} |Z_{l,e}| = \sum_{l,e} |E_{l,e}| = |E_Z| = n|Z| = n(\mathbf{e}_\mathcal{E}(i))_Z
\]

(3.12)

Proposition 3.12 The splitting of the graph \( G_{\mathcal{E}}^{(i-1)} \), which yields the graph \( G_{\mathcal{E}}^{(i)} \), is consistent with respect to \( \mathbf{e}_\mathcal{E}(i-1) \), for \( 1 \leq i \leq A \). The resulting eigenvector is \( \mathbf{e}_\mathcal{E}(i) \).

Proof: Let \( Z = T_{\mathcal{E}}(\pi, w) \) be a state in \( G_{\mathcal{E}}^{(i-1)} \), \( 1 \leq i \leq A \). In the \( i \)th round of splitting, \( Z \) is split into the states \( \{Y\} = \{T_{\mathcal{E}}(\pi, w^{1})\} \), for the various \( i \)th edges \( e \) in the paths of \( Z \). Since \( \{Y\} \) is a partition of the set of paths in \( Z \), we have

\[
\sum_{Y \in \text{descendant}(Z)} (\mathbf{e}_\mathcal{E}(i))_Y = \sum_{Y} |Y| = |Z| = (\mathbf{e}_\mathcal{E}(i-1))_Z
\]

(3.13)
As $c_{E}^{(i)}$ is an eigenvector of $A_{G_{E}^{(i)}}$, we get that the splitting of $G_{E}^{(i-1)}$ is consistent with $c_{E}^{(i-1)}$ and it yields the eigenvector $c_{E}^{(i)}$.

We now prove Lemma 3.3, using the six preceding propositions.

**Proof:** By Propositions 3.7, 3.9, 3.11, and 3.12, the determinizing graph $G_{E}$ of $E$ can be split into $A$ rounds in consistency with the eigenvector $c_{E}$. By Proposition 3.10, the components of $c_{E}$ are not greater than $n^{A}$. By Proposition 3.8, at the end of the splitting process, the all-one vector is obtained and the resulting graph, $G_{E}^{(A)}$, is an $(S_{E}, n)$ encoder of anticipation $A$. The original encoder $E$ can be obtained from $G_{E}^{(A)}$ by a reduction of strongly equivalent states. By Proposition 3.9, in the $i$th round of splitting, each state of the intermediate graph $G_{E}^{(i-1)}$ is split into $n$ states at most, and by Proposition 3.10, all the components of the resulting eigenvector, $c_{E}^{(i)}$, are not greater than $n^{A-i}$. 

$\blacksquare$
Chapter 4

First Lower Bound on the Anticipation

4.1 Statement of the Result — Theorem 4.1

We present now the first main theorem of this work, which implies a new lower bound on the anticipation of \((S, n)\) encoders by means of state-splitting.

**Theorem 4.1** Let \(S\) be an irreducible constrained system and let \(n\) be a positive integer where \(C(S) \geq \log n\). Suppose there exists some irreducible \((S, n)\) encoder, \(E\), of a finite anticipation \(A\). Then there exists an irreducible deterministic graph \(G\) which is a presentation of \(S\), not necessarily reduced, and \(G\) satisfies the following:

(i) There is an \((A_G, n)\)-approximate eigenvector, \(\bar{c}\), whose largest component is not greater than \(n^A\).

(ii) The graph \(G\) can be fully split according to \(\bar{c}\) in \(A\) rounds of splitting. An \((S, n)\) encoder, \(E_G\), whose anticipation is \(A\), is obtained after deleting excess edges.

(iii) In each round of splitting, each of the states is split into no more than \(n\) states.

(iv) In the \(i\)th round of splitting, the components of the resulting approximate-eigenvector, \(\bar{c}^{(i)}\), are not greater than \(n^{A-i}\).

(v) The encoder \(E\) can be obtained from \(E_G\) by a reduction of strongly equivalent states.

**Remark:** There is no loss of generality in requiring the existence of an irreducible \((S, n)\) encoder \(E\) of finite anticipation \(A\), since the existence of an \((S, n)\) encoder with finite anticipation \(A\) implies that each of its irreducible sinks is an irreducible \((S, n)\) encoder with anticipation not greater than \(A\). (See the discussion at the end of Section 2.3.)
Theorem 4.1 can be viewed as a new lower bound on the anticipation of \((S, n)\) encoders, using the following corollary. In Section 4.3, we present examples which show that this lower bound indeed improves on the one stated in [MR91].

**Corollary 4.2** Let \(S\) and \(n\) be as in Theorem 4.1. The anticipation of any \((S, n)\) encoder cannot be smaller than the minimal integer \(A\) for which there exists a presentation \(G\) of \(S\) that can be split in \(A\) rounds while satisfying conditions (i)-(v) of Theorem 4.1.

The proof of Theorem 4.1 is given through a sequence of constructions and propositions in Section 4.2, following the next discussion.

Every \((S, n)\) encoder \(E\) can be regarded as an \((S_E, n)\) encoder, where \(S_E = S(E) \subseteq S\). In Section 3.1, we defined \(G_E\) to be a specific irreducible deterministic presentation of \(S_E\). We then showed in Lemma 3.3 that \(G_E\) satisfies the conditions of Theorem 4.1 with respect to the system \(S_E\). Hence, when \(C(S) = \log n\) — in which case \(S = S_E\) (see Section 2.3) — Theorem 4.1 is an immediate corollary of Lemma 3.3.

In order to prove the theorem for \(C(S) \geq \log n\) \((S_E \subseteq S)\), we describe in Section 4.2 a certain construction of a graph \(G\), which is an irreducible deterministic presentation of \(S\). The graph \(G\), which will serve as the graph claimed in Theorem 4.1, contains an unreduced version of \(G_E\) as a subgraph.

Let \(G_S\) be the Shannon cover of \(S\) and let \(G_E\) be the irreducible deterministic presentation of \(S_E\), as defined in Section 3.1. The graphs \(G_E\) and \(G_S\) are both irreducible and deterministic, and the constrained systems they present satisfy the inclusion \(S_E \subseteq S\). Hence, by Proposition 2.2 (4), every state \(v'\) in \(G_E\) is dominated by at least one state \(v\) of \(G_S\), which means that the follower set \(F_{G_E}(v')\) is contained in the follower set \(F_{G_S}(v)\). In particular, the existence of an edge labeled \(l\) from \(v'\) to \(u'\) in \(G_E\) implies the existence of an edge labeled \(l\) in \(G_S\) from \(v\) to some \(u\), where the state \(u'\) of \(G_E\) is dominated by state \(u\) of \(G_S\).

Since the graph \(G\) claimed in Theorem 4.1 is an irreducible deterministic presentation of \(S\), it is obtained from \(G_S\) by an expansion of states. It is tempting to try to take the Shannon cover, \(G_S\), as the graph \(G\) and to require that the splitting of every state \(v\) of \(G_S\) will be the same as that of some state \(v'\) of \(G_E\) which is dominated by \(v\). Every outgoing edge of \(v'\) corresponds to an outgoing edge of \(v\) which carries the same label, so it seems that \(v\) can split the same way as \(v'\), in the sense of partitioning its outgoing edges among its descendant states. This idea does indeed work in the first round of splitting and we discuss it in the proof of Theorem 6.1, which deals with one round of splitting of \(G_S\) and its power graphs.

However, this idea fails when it comes to the second round of splitting. The way in which a state (or more precisely, its descendant states) can be split in the second round depend on the way in which the outgoing neighbors are split in the first round. Hence, to enable a state \(v\) of \(G_S\) to split also in the second round the same way as \(v'\) does in \(G_E\), every outgoing neighbor \(u\) of \(v\) in \(G\), which corresponds to some neighbor \(u'\) of \(v'\) in \(G_E\), should split as \(u'\) does in the first round.
Yet, the state \( u \) of \( G_S \) may dominate another state \( u'' \) of \( G_E \), which is not necessarily equivalent to \( u' \) and does not necessarily split in the first round the same way as \( u' \) splits. Suppose that in the first round the state \( u \) of \( G_S \) chooses to split the same way as state \( u'' \) splits in \( G_E \). Moreover assume that the splitting of \( u'' \) differs from the splitting of \( u' \), in the number and the weights of the descendant states after the first round. Then in the second round of splitting, it may be impossible for the descendant states of \( v \) to split like the descendant states of \( v' \), as the local neighborhoods of the former states may not be isomorphic to those of the latter states.

### 4.2 Proof of Theorem 4.1

The construction of \( G \) described in this section solves the difficulties mentioned in Section 4.1 in the following way: For every pair of states, \( v' \in G_E \) and \( v \in G_S \), where \( v' \) is dominated by \( v \), we dedicate a state \((v', v)\) in \( G \). This state is equivalent to the state \( v \) of \( G_S \), by means of follower sets, but it splits the same way as \( v' \) does in \( G_E \) throughout the \( A \) rounds of splitting. In addition, \( G \) consists of some extra states that are equivalent to those states of \( G_S \) which do not dominate any state of \( G_E \). The formal and exact procedure of constructing \( G \) is given below.

#### 4.2.1 Construction of \( G \) — Step 1

In the first step, we construct a deterministic presentation of \( S_E \) which can be obtained from \( G_E \) by expansion of states. Let \( G_0 \) be the subgraph of the label product \( G_E \times G_S \), in which the states are defined by

\[
V(G_0) = \{(v', v) : v' \in V(G_E), v \in V(G_S), v' \preceq v\} \tag{4.1}
\]

An edge labeled \( l \) goes in \( G_0 \) from \((u', u)\) to \((v', v)\) if and only if an edge labeled \( l \) goes in \( G_E \) from \( u' \) to \( v' \) and an edge labeled \( l \) goes in \( G_S \) from \( u \) to \( v \). Let \( G_1 \) be an irreducible sink of \( G_0 \).

**Proposition 4.3** The graph \( G_1 \) is a presentation of \( S_E \) that can be obtained from \( G_E \) by an operation of expansion.

**Proof:** Every state \((u', u)\) in \( G_1 \) is a copy of the state \( u' \) in \( G_E \): the set of labels on the edges going out of \((u', u)\) in \( G_1 \) is the same as the set of labels on the edges going out of \( u' \) in \( G_E \). The target state of an edge labeled \( l \) going out of \((u', u)\) in \( G_1 \) is \((v', v)\), for some \( v \in V(G_S) \), if and only if the target state of the corresponding edge of \( u' \) in \( G_E \) is \( v' \).

On the other hand, for every \( v' \in V(G_E) \) there is a state \( u \in V(G_S) \) such that \((u', u)\) is a state in \( G_1 \): By Proposition 2.2 and by the definition of \( G_0 \) and \( G_1 \), the set \( V(G_1) \) is not empty, so it contains some state \((v', v)\). As \( G_E \) is irreducible, there is a path from \( v' \) to \( u' \) in \( G_E \). A respective path must exist in \( G_0 \) from \((v', v)\) to \((u', u)\), for some \( u \in V(G_S) \), and this path is also a path of \( G_1 \), as \( G_1 \) is an irreducible sink of \( G_0 \) that contains the state \((v', v)\). ■

40
Let $\tilde{e}$ be the eigenvector of $G_\mathcal{E}$ associated with the eigenvalue $n$, as defined by Equation (3.4). We define $\tilde{e}_1 = \{(e_1(v',v))_{(v',v)\in V(G_1)}\}$ to be the eigenvector of $A_{G_1}$ associated with the eigenvalue $n$, where $\tilde{e}_1$ is an unreduced version of $\tilde{e}$ (see Section 2.7):

\[(e_1(v',v)) = (e_\mathcal{E})_{v'} \] (4.2)

**Proposition 4.4** The graph $G_1$ and the (true) eigenvector $\tilde{e}_1$ of $A_{G_1}$, associated with the eigenvalue $n$, satisfy conditions (i)-(v) of Theorem 4.1 with respect to the system $S_\mathcal{E}$.

**Proof:**

(i) The maximal component of $\tilde{e}_1$ equals the maximal component of $\tilde{e}$, and therefore it is not greater than $n^A$.

(ii) Using Corollary 2.12, we get that $G_1$ can be split according to $\tilde{e}_1$ in at most $A$ rounds, until the all-one vector is obtained. (The all-one vector obtained by splitting $G_1$ is an unreduced version of the all-one vector obtained by splitting $G_\mathcal{E}$.)

The anticipation of $E_1$ is not greater than $A$, since $E_1$ is obtained from a deterministic presentation of $S_\mathcal{E}$ by $A$ rounds of splitting. According to Proposition 2.9, the anticipation of $E_1$ cannot be smaller than the anticipation of its reduced version $E'_1$. By Lemma 3.3, the anticipation of $E'_1$ is $A$. Hence, the anticipation of $E_1$ is exactly $A$.

(iii) By the discussion following Observation 2.11, each state of $G_1$ splits the same way as the respective equivalent state splits in $G_\mathcal{E}$, and thus in every round of splitting of $G_1$, each state is split into no more than $n$ states.

(iv) By Corollary 2.12, the components of $\tilde{e}_1^{(i)}$ are equal to the respective components of $\tilde{e}_\mathcal{E}^{(i)}$, and therefore, the largest component of $\tilde{e}_1^{(i)}$ cannot exceed $n^{A-i}$.

(v) The graph obtained by splitting $G_1$ is an $(S_\mathcal{E}, n)$ encoder, since the all one-vector is its (true) eigenvector. Denote this encoder by $E_1$. By Corollary 2.12, the graph $E_1$ can be obtained by an operation of expansion from the encoder graph $E'$ which is obtained after splitting $G_\mathcal{E}$. The graph $E'$, in turn, can be obtained from the original encoder $E$ by an operation of expansion and therefore, the graph $E_1$ can be obtained directly from $E$ by an expansion. Thus, the graph $E_1$ can turn into $E$ by a reduction of strongly equivalent states, using Observation 2.10.

\[\square\]

### 4.2.2 Construction of $G$ — Step 2

We now turn the graph presentation $G_1$ of $S_\mathcal{E}$ into a presentation $G_2$ of $S$ by adding the following set of states to $V(G_1)$:

\[\{(\phi, v) : v \in V(G_\mathcal{E})\}\] (4.3)

We refer to the above states as the $\phi$-states. The graph $G_2$ contains all the edges of $G_1$ and additional edges which are of two types, depending on their initial state, as follows:
1. \((u', u) \in V(G_2)\), where \(u' \neq \phi\): If there is an edge labeled \(l\) in \(G_S\) from \(u\) to some state \(v\) but there is no edge labeled \(l\) in \(G_2\) going out of \(u'\), we draw an edge labeled \(l\) from \((u', u)\) to \((\phi, v)\) in \(G_2\).

2. \((\phi, u) \in V(G_2)\): We draw an edge labeled \(l\) going out of \((\phi, u)\) in \(G_2\) for every edge \((u \xrightarrow{l} v)\) in \(G_S\). In case \(G_2\) contains some states \(\{ (v', v) \}\) in which \(v' \neq \phi\), the edge \(e\) is directed into one of these states. Otherwise, \(e\) is directed into \((\phi, v)\).

### 4.2.3 Construction of \(G\) — Step 3

Finally, to obtain the graph \(G\) from the graph \(G_2\), we delete repeatedly all the states \((\phi, v)\) in \(G_2\) that have no incoming edges. Notice that \(G\) contains the graph \(G_1\) as a subgraph and \(G_1\), in turn, is an unreduced version of \(G_S\).

**Proposition 4.5** *The graph \(G\) that was constructed from \(G_2\) is an irreducible deterministic presentation of \(S\).*

**Proof:** Presentation of \(S\): We show that \(S(G) = S(G_S)\).

**(P1)** First, we prove that for every \((u', u) \in V(G)\), we have \(F_G((u', u)) = F_{G_S}(u)\). The proof is by induction on the length of the words in the follower sets, where the set of words of length \(i\) in a follower set \(F_H(v)\) is denoted by \(F^i_H(v)\). The words in \(F^i_G((u', u))\) are the labels on the edges going out of \((u', u)\) in \(G\). By the construction, these labels also appear on the edges going out of \(u\) in \(G_S\), so \(F^i_G((u', u)) = F^i_{G_S}(u)\). Now assume that \(F^i_G((u', u)) = F^i_{G_S}(u)\). A word of length \(i + 1\), going out of a state \((u', u)\), is composed of a label which appears on an edge from \((u', u)\) to some neighbor \((v', v)\), concatenated with a word of length \(i\) that is generated from \((v', v)\). By the construction, the existence of an edge labeled \(l\) from \((u', u)\) to \((v', v)\) in \(G\) implies the existence of an edge labeled \(l\) from \(u\) to \(v\) in \(G_S\). According to the induction assumption, \(F^i_G((v', v)) = F^i_{G_S}(v)\). Thus, concatenating the label on the edge entering a neighbor to words of length \(i\), that go out of that neighbor, yields the same words of length \(i + 1\) in both \(F^{i+1}_G((u', u))\) and \(F^{i+1}_{G_S}(u)\). Therefore, \(F_G((u', u)) = F_{G_S}(u)\) for every \((u', u) \in V(G)\).

**(P2)** For every state \(u \in V(G_S)\) there exists at least one state \((u', u)\) in \(V(G)\): The set \(V(G)\) is not empty, so it contains some state \((v', v)\). Since \(G_S\) is irreducible, \(u\) is accessible from \(v\) in \(G_S\) and, hence, there must be a path in \(G\) from \((v', v)\) to some \((u', u)\), where \(u' \in V(G_2) \cup \{ \phi \}\).

Observations \((P1)\) and \((P2)\) imply that \(S(G) = S\).

**Deterministic:** The set of labels on the outgoing edges of a state \((u', u)\) in \(G\) is identical to the set of labels on the outgoing edges of \(u\) in \(G_S\). The Shannon cover \(G_S\) is a deterministic graph, and so is \(G\).

**Irreducible:**
(I1) As $G_1$ is irreducible, every state $(v', v)$ in $G$ that belongs to $V(G_1)$ (i.e., $v' \neq \phi$) is accessible from every other state $(u', u) \in V(G_1)$.

(I2) Every state $(\phi, v)$ is accessible in $G$ from every state $(u', u)$ of $V(G_1)$:

If $(\phi, v)$ is left after Step 2 (see Section 4.2.2), it means that it has some incoming edge. If the initial state of this edge belongs to $V(G_1)$, then, using (I1), we are done. Otherwise, the incoming edge starts at some other $\phi$-state, which means that there is no other state $(v', v)$ in $G$ except $(\phi, v)$. As every state $u$ in $G_S$ has a path to state $v$, we get that every state $(u', u)$ in $G$ has a corresponding path that must reach $(\phi, v)$.

(I3) A state $(\phi, u)$ has an outgoing path in $G$ to every state $(v', v) \in V(G_1)$:

Due to (I1), it is enough to show that $(\phi, u)$ has a path to some state of $G_1$ in $G$. As $V(G_1)$ is not empty, it contains some state $(v', v)$. There must be a path from $u$ to $v$ in $G_S$ and a corresponding path in $G$ which goes out of $(\phi, u)$ and has the form

$$(\phi, u) \rightarrow (u_1', u_1) \rightarrow \ldots \rightarrow (u_k', u_k) = (u_k', v)$$

If $u_i' \neq \phi$ for some $i < k$, we are done. Otherwise, If $u_i = \phi$ for every $i < k$, then $u_k'$ must be $v'$ or some other state $v''$ of $G_1$ which is dominated by $v$.

(I4) Observations (I2) and (I3) also imply that every $\phi$-state in $G$ is connected by a path to any other $\phi$-state.

These properties imply the irreducibility of $G$. ■

The graph $G$ is therefore an irreducible deterministic presentation of $S$, which means that it is an unreduced version of the Shannon cover, $G_S$, and can be obtained from $G_S$ by an operation of expansion.

**Proposition 4.6** The graph $G$ that was constructed in this section satisfies conditions (i)-(v) of Theorem 4.1.

**Proof:** Let $\tilde{c}_u$ be the eigenvector of $G_1$ associated with the eigenvalue $u$, as defined by Equation (3.4), and let $\tilde{c}_1$ be the eigenvector of $A_{G_1}$ associated with $u$, where $\tilde{c}_1$ is an unreduced version of $\tilde{c}_u$. The $(A_G, n)$-approximate eigenvector, $\tilde{c} = [\tilde{c}(v', v)]_{(v', v) \in V(G)}$ is defined as follows:

$$c(v', v) = \begin{cases} 
(\tilde{c}_u)_{v'} & \text{if } v' \neq \phi \\
0 & \text{if } v' = \phi 
\end{cases} \quad (4.4)$$

or, equivalently:

$$c(v', v) = \begin{cases} 
(\tilde{c}_1)_{(v', v)} & \text{if } (v', v) \in V(G_1) \\
0 & \text{otherwise} 
\end{cases} \quad (4.5)$$

To see that $\tilde{c}$ is an $(A_G, n)$-approximate eigenvector, let $(u', u)$ be a state of $G_1$ in $G$. The neighborhood of $(u', u)$ in $G$ contains all the states which are neighbors of $(u', u)$ in $G_1$,
appearing with the same weights as in $G_1$. The neighborhood of $(u', u)$ in $G$ may also include some extra $\phi$-states, i.e., states with zero weight. Thus, for every state $(u', u) \in V(G_1)$ in $G$ we get the equality

$$
(A_G \bar{\epsilon})_{(u', u)} = (A_G \bar{\epsilon})_{(u', u)} + 0 = nc_1_{(u', u)} = nc_{(u', u)}
$$

For a state $(\phi, u)$, we get:

$$
(A_G \bar{\epsilon})_{(\phi, u)} \geq 0 = nc_{(\phi, u)}
$$

Hence, $\bar{\epsilon}$ is an $(A_G, n)$-approximate eigenvector.

We now show that the graph $G$ and the $(A_G, n)$-approximate eigenvector $\bar{\epsilon}$ satisfy conditions (i)-(v) of Theorem 4.1, with respect to the constrained system $S$. The components of the vector $\bar{\epsilon}$ are either components of $\bar{\epsilon}_1$ or are equal to zero. Thus, each of the components of $\bar{\epsilon}$ does not exceed $n$.

The definition of $\bar{\epsilon}$ by Equation (4.5) emphasizes the fact that in addition to the states and edges of $G_1$, the graph $G$ contains only zero-weight states and their outgoing and incoming edges. Hence, after omitting all the zero-weight states from $G$ and removing the edges attached to them, we obtain the graph $G_1$, which can be split in $A$ rounds while satisfying the requirements of Theorem 4.1.

Using Proposition 4.4, in any round of the splitting, each state is split into no more than $n$ states, and the components in the approximate eigenvector $\bar{\epsilon}^{(i)}$, obtained after the $i$th round, cannot be greater than $n^{A-i}$.

At the end of the splitting, we get the $(S, n)$ encoder $E_G = E_1$. We have already proved that the anticipation of $E_1$ is $A$ and that it can be turned into the original encoder, $E$, by a reduction of strongly equivalent states.

This concludes the proof that there exists some irreducible deterministic presentation $G$ of $S$, which satisfies conditions (i)-(v) of Theorem 4.1.

4.3 Application of Theorem 4.1 — Examples

4.3.1 Preliminaries

In this section, we show how Theorem 4.1 can be applied to obtain lower bounds on the anticipation of encoders for a given constrained system. We demonstrate this for two specific constrained systems which widely appear in the literature and in practice. For those constrained systems, our bounds are tight and improve on the bounds which can be computed from [MR91].

At first sight, Theorem 4.1 may not look too useful for getting lower bounds on the anticipation of encoders for given systems: Suppose we wish to prove that every encoder at rate $p : q$ for a system $S$ has anticipation at least $A$. One way to do so is to show that there cannot exist an $(S^p, 2^p)$ encoder whose anticipation is $A - 1$ or less. To obtain that from Theorem 4.1, we should prove that none of the irreducible deterministic presentations
of $S$ can be split in no more than $A - 1$ rounds, while satisfying conditions (i)-(v) of the theorem. However, it is impractical to examine all the presentations of $S$ and show that none of them fulfills the conditions of the theorem, as the number of such presentations may be very big. (We show in Chapter 5 that it is sufficient to consider only finitely many such presentations, and we also bound their number of states from above.)

Nevertheless, in this section we show how Theorem 4.1 implies that any encoder at rate $2:3$ for the $(1,7)$-RLL constraint has anticipation at least 2 and any encoder at rate $2:5$ for the $(2,18,2)$-RLL constraint has anticipation at least 3. Rate $2:3$ for the $(1,7)$-RLL constraint and rate $2:5$ for the $(2,18,2)$-RLL constraint are the common rates for encoding in these cases, as they are very close to the respective capacities of the constrained systems. We point out that the old lower bound of [MR91] yields anticipation 1 for the $(1,7)$-RLL constraint at rate $2:3$ and 2 for the $(2,18,2)$-RLL constraint at rate $2:5$. Later in this work, we prove these two results in a simpler way, using Theorem 6.1, but here we present the lower bounds through a rather simple application of Theorem 4.1.

Suppose we are given a system $S$ and integers $A, q, n$ (where typically $n = 2^p$ for an integer $p$), and suppose we want to prove that there is no $(S^q, n)$ encoder with anticipation $A$ or less. We do so by showing that for every irreducible and deterministic presentation $G$ of $S$, and for every $(A_G, n)$-approximate eigenvector $x$ of $S$, it is impossible to have the following:

- the maximal component in $x$ is not greater than $n^A$,
- $G$ can be split in $A$ rounds, consistently with $x$, where in every round each state is split into $n$ states at most, and
- after the $i$th round, the components of the resulting approximate eigenvector are not greater than $n^{A-i}$.

In order to show this for every irreducible deterministic presentation of $S$, we use Proposition 4.7 below and its two corollaries, which establish a connection between the set of approximate eigenvectors $\mathcal{X}(A_G, n)$ and the set $\mathcal{X}(A_G, n)$, where $G_S$ is the Shannon cover of $S$ and $G$ is any other irreducible deterministic presentation of $S$. Recall that by Proposition 2.2, for every state $u$ in $G$ there exists a unique state $u'$ in $G_S$ so that $f_G(u) = f_{G_S}(u')$.

**Proposition 4.7** Let $S$ be an irreducible constrained system and let $n$ be a positive integer and $u \in V(G)$ to the (unique) $u' \in V(G_S)$ which is equivalent to $u$, i.e., $f_G(u) = f_{G_S}(u')$. Suppose there exists some $(A_G, n)$-approximate eigenvector $x$ whose maximal component equals $L$. Then there also exists some $(A_G, n)$-approximate eigenvector $x'$ whose maximal component is $L$ and $x_u \leq x'_u$ for every $u \in V(G)$.

**Proof:** Given the vector $x = [x_u]_{u \in V(G)}$, define the vector $x' = [x'_u]_{u \in V(G_S)}$ by

$$x'_u = \max_{u \in V(G), f(u) = u'} x_u \quad u' \in V(G_S)$$

(4.8)
The definition of $\bar{z}'$ implies $x_u \leq x'_{f(u)}$ for every $u \in V(G)$. Let $u'$ be a state in $V(G_S)$ and let $u$ be a state in $V(G)$ such that $u' = f(u)$ and $x'_{u'} = x_u = \max_{v \in V(G) : f(v) = u'} x_v$. Every neighbor $v$ of $u$ in $G$ is mapped by $f$ to a neighbor $v'$ of $u'$ in $G_S$, and $x'_{v'} \geq x_v$. Consequently, we get:

$$
(A_{G_S} \bar{z}')_u \geq (A_G \bar{z})_u \geq n x_u = n x'_{u'}
$$

and, therefore, $\bar{z}'$ is an $(A_{G_S}, n)$-approximate eigenvector with maximal component $L$. The maximal component of $\bar{z}'$ equals $L$, as $\max_{u' \in V(G_S)} x'_{u'} = \max_{u \in V(G)} x_u = L$.

**Corollary 4.8** Let $S$, $G_S$, and $G$ be as in Proposition 4.7 and let $L$ be a positive integer. If no $(A_{G_S}, n)$-approximate eigenvector with maximal component $L$ exists, then an $(A_G, n)$-approximate eigenvector with maximal component $L$ does not exist either.

**Corollary 4.9** Let $S$, $G_S$, and $f$ be as in Proposition 4.7. Let $L$ and $M$ be positive integers and $u'$ be a state of $G_S$. If every $(A_{G_S}, n)$-approximate eigenvector $\bar{z}'$ with maximal component $L$ satisfies $x'_{u'} < M$, then no $(A_G, n)$-approximate eigenvector $\bar{z}$ with maximal component $L$ satisfies $x_u \geq M$, where $f(u) = u'$.

### 4.3.2 (1,7)-RLL Constraint

We prove here that every encoder at rate 2:3 for the (1,7)-RLL constraint has anticipation not smaller than 2. Rate 2:3 is the common rate for encoding in the case of the (1,7)-RLL...
constraint since the capacity in this case is 0.6793. This result is an improvement of the old lower bound from [MR91] which implies $A \geq 1$. The new lower bound, 2, is achieved by some known encoders, such as the one presented in [WW91].

Let us denote by $S_{(1,7)}$ the system of all the finite words which satisfy the (1,7)-RLL constraint. Since the rate is $2:3$, we take $S = S_{(1,7)}^3$ and $n = 2^2 = 4$. We assume to the contrary the existence of an $(S,4)$ encoder whose anticipation is $A = 1$. Suppose that there exists some presentation $G$ of $S$, which is irreducible and deterministic, and there is an $(A_G,4)$-approximate eigenvector $\bar{x}$ whose maximal component is not greater than $4(= n^A)$. We then show that it is impossible to split $G$ in an $\bar{x}$-consistent single round, so that every state of $G$ is split into no more than $4(= n)$ states and the components in the resulting approximate vector are not greater than $1(= n^{A-1})$.

Consider the Shannon Cover, $G_S$, of $S_{(1,7)}^3$ which consists of eight states, $v_0, \ldots, v_7$. The local neighborhood (picture) of each state $v_i \in V(G_S)$, is illustrated in Figure 4.1. By running Franaszek's algorithm, it turns out that there are only two $(A_G,4)$-approximate eigenvectors with maximal component not greater than 4: $\bar{y} = (2,3,3,3,2,2,2,1)$ and $\bar{y}' = (2,3,3,3,2,2,2,0)$. The maximal component in both vectors is 3, and the states whose respective components equal 3 are $v_1$, $v_2$, and $v_3$. These three states all have outdegree 5 in $G_S$.

Applying Corollary 4.8 on $S_{(1,7)}^3$ with $L = 1,2,4$, we get that the maximal component of the $(A_G,4)$-approximate eigenvector $\bar{x}$ must also be equal to 3. We now apply Corollary 4.9 to each of the states $v_0$, $v_4$, $v_5$, $v_6$, and $v_7$ in $G_S$ with $L = 3$ and $M = 3$. It follows that $x_u = 3$ for a state $u$ in $G$ only if $u$ is equivalent to one of $v_1$, $v_2$, or $v_3$. Such a state $u$ must therefore have exactly five outgoing edges.

Now assume to the contrary that in one round the graph $G$ can be fully split consistently with $\bar{x}$. Splitting $u$ in this round means generating three descendant states, $u^1$, $u^2$ and $u^3$, each of weight 1, and distributing the five outgoing edges of $u$ among those states. It is clear that at least one of the descendant states, say $u^1$, inherits only a single edge. As the target state of this edge has respective component in $\bar{x}$ which is not greater than 3, we get that the sum of weights in the neighborhood of $u^1$ in the split graph is smaller than $(n =)4$ times the weight of $u^1$, and therefore the splitting is not $\bar{x}$-consistent.

We conclude that none of the presentations $G$ of $S$ can be split in one round while satisfying the conditions of Theorem 4.1, and therefore no $(S,4)$ encoder exists with anticipation smaller than 2.

4.3.3 (2,18,2)-RLL Constraint

In this example, we prove that any encoder at rate $2:5$ for the system of the (2,18,2)-RLL constraint has anticipation at least 3. Rate $2:5$ is the common rate for encoding in this case, as the capacity of the (2,18,2)-RLL constraint is 0.40403. This result is an improvement of the result $A \geq 2$, which is implied by the bound in [MR91]. The new lower bound, 3, is attained by some known encoders, such as those described in [Holl95] and [Weig90]. Let $S_{(2,18,2)}$ be the set of binary words satisfying the (2,18,2)-RLL constraint. Since the rate is
2:5, we take $S = S^5_{(2,18,2)}$ and $n = 2^2 = 4$. The Shannon cover $G_S$ of this system consists of nineteen states, $v_0, v_1, \ldots, v_{18}$, each having outdegree at most 5. The local neighborhoods of the states in $G_S$ are illustrated in Figure 4.2.

Using Franaszek’s algorithm, we obtain that the minimal value for the maximal component of any $(A_{G_S},4)$-approximate eigenvector is 12. According to Corollary 4.8, for every deterministic presentation $G$ of $S$, every $(A_{G},4)$-approximate eigenvector has maximal component 12 or more. As 12 is greater than $n^1 (= 4)$, it is clear that anticipation 1 is impossible for any $(S,4)$ encoder. Although 12 is not greater than $n^2 (= 16)$, we prove here that the anticipation of any $(S,4)$ encoder can neither be $A = 2$. We show that for every presentation $G$ of $S$, and for every $(A_{G},4)$-approximate eigenvector $\vec{x}$, it is impossible to have the following:

(A) The maximal component of $\vec{x}$ is not greater than $n^A = 4^2 = 16$.

(B) $G$ can be fully split in $2 (= A)$ rounds.

(C) In each round, every state is split into $4 (= n)$ states at most.

(D) After the first round, the components of the resulting approximate eigenvector are not greater than $4 (= n^{A-1})$, and after the second round, they are not greater than $1 (= n^{A-2})$.

By Theorem 4.1, this will imply the nonexistence of an $(S,4)$ encoder with anticipation 2.

Let $G$ be some irreducible deterministic presentation of $S = S^5_{(2,18,2)}$ and assume the existence of an $(A_{G},4)$-approximate eigenvector $\vec{x} = [x_{u}]_{u \in V(G)}$ with maximal component
not greater than 16. We now examine all the possible ways to split $G$ according to $x$ while satisfying (A)-(D), and show that none of these splittings is $x$-consistent. Since we assume that $\max_{u \in V(G)} x_u \leq 16$, we distinguish between the following two cases:

**Case 1:** $12 \leq \max_{u \in V(G)} x_u < 16$

Let $u$ be a state in $G$ with $x_u \geq 12$. The sum of weights of the descendant states of $u$ after the first round should be equal to $x_u$, and therefore this sum cannot be less than 12. By (D), the weights of the states after the first round cannot be greater than 4, and therefore $u$ must be split in the first round into three states or more. It is impossible for $u$ to be split into exactly three states, for the following reasons:

- Splitting $u$ into three states implies that each of them has weight exactly 4. Now, the outdegree of $u$ in $G$ is at most 5, as the outdegree of every state in $G_S$, and therefore in $G$, is at most 5. Hence, when we split $u$ into three states we have to distribute at most five edges among three descendant states.

At least one of the three descendant states inherits only a single edge, whose target state has weight smaller than 16. As the weight of the descendant state is 4, we get that the weight of its neighborhood is smaller than $u$ times its own weight. Thus, we conclude that there is no $x$-consistent splitting of $u$ into three states (or less).

On the other hand, according to (C), $u$ cannot be split in the first round into more than four states. Hence, the only remaining option is to split $u$ in the first round into exactly four states, which means distributing the (at most five) outgoing edges of $u$ among the four descendant states. At most one descendant state then gets more than one edge. Each of the other descendant states gets a single edge, and therefore each such state should have weight not greater than 3, or else the splitting will be inconsistent with respect to $x$. Taking into account that the sum of weights of the four descendant states of $u$ must be at least 12, we get that at least two of the descendant states that inherit a single edge must have weight exactly 3.

Let $u^1$ be such a descendant state of $u$ of weight 3 that inherits a single edge of $u$ in the first round. Suppose that this original edge goes in $G$ from $u$ to $v$. The weight of $v$ is equal to, or greater than, $u$ times the weight of $u^1$, i.e., $x_v \geq 12$. Therefore, the ways $v$ could be split in the first round must be similar to the way $u$ did, namely, splitting into four states, at most one of which has weight 4 and at least two of which have weight 3. The local neighborhood of $u^1$ thus looks like one of the three alternatives, (a), (b), or (c), in Figure 4.3 (the weights are written inside the circles).

Now, if $G$ is the graph guaranteed by Theorem 4.1, then $u^1$ has to be split in the second round into three states, $u^{1,1}$, $u^{1,2}$, $u^{1,3}$, each of weight 1. As $u^1$ has outdegree 4, two of its descendant states, say $u^{1,1}$ and $u^{1,2}$, each inherits only a single edge of $u^1$. One of these states, say $u^{1,1}$, inherits an outgoing edge of $u^1$ that terminates in a state $v'$ whose weight is smaller than 4. So the sum of weights in the neighborhood of $u^{1,1}$ is smaller than $(u \times 4)$ times its weight. Thus, this splitting process turns out to be inconsistent with respect to $x$.

**Case 2:** $\max_{u \in V(G)} x_u = 16$

By the Franaszek algorithm, the $(A_{G_S}, n)$-approximate eigenvector which is the largest (componentwise) among those with maximal component 16 is:

$$ \bar{y} = (9, 12, 16, 12, 16, 12, 16, 11, 15, 11, 15, 11, 14, 13, 8, 11, 5, 7) $$
Figure 4.3: Possible outgoing pictures from state $u^1$ after the first round
Looking at the graph $G_S$ in Figure 4.2, we observe that according to $\bar{y}$, any state of $G_S$ can have no more than two neighbors with weight 16. The maximality of $\bar{y}$ implies that this is also the case with respect to any other $(A_{G_S}, n)$-approximate eigenvector. We also claim that a state in the graph $G$ cannot have more than two neighbors whose respective components in $\bar{x}$ are equal to 16. To see this, consider some state $u$ of $G$ and recall the function $f$, which is defined in Proposition 4.7. Every neighbor $v$ of $u$ is mapped by $f$ to a neighbor $f(v)$ of $f(u)$ in $G_S$.

Using Corollary 4.9, if the respective component of $f(v)$ in every $(A_{G_S}, 4)$-approximate eigenvector is smaller than 16, then $x_v$ is also smaller than 16. Since for every $(A_{G_S}, 4)$-approximate eigenvector $\bar{x}'$, the state $f(u)$ has at most two neighbors $f(v)$ with respective components 16 in $\bar{x}'$, it follows that the state $u$ of $G$ cannot have more than two neighbors $v$ whose respective components in $\bar{x}$ equal to 16. Otherwise, we could obtain from Equation (4.8) an $(A_{G_S}, 4)$-approximate eigenvector $\bar{x}'$ in which the components corresponding to three neighbors of $f(u)$ would be equal to 16.

Now, let $u$ be a state in $G$ with $x_u = 16$. There is only one way to split $u$ into no more than four states, each with weight not greater than 4, and this is by splitting $u$ into exactly four states, each of weight exactly 4. At least three of these four descendant states inherit only a single edge of $u$, as there are no more than five edges outgoing from $u$. As $u$ has at most two neighbors of weight 16 according to $\bar{x}$, at least one of the descendant states of $u$, say $u^1$, gets a single edge that leads to a state of weight smaller than 16. As the weight of state $u^1$ is 4, it implies that the weight of its neighborhood is smaller than 4($= n$) times its own weight. Hence, the splitting in this case is not $\bar{x}$-consistent.

We can conclude that there is no graph presentation $G$ that satisfies the requirement of Theorem 4.1 for $\mathcal{A} = 2$, and therefore no $(S, 4)$ encoder exists with anticipation 2.
Chapter 5

Upper Bounds on the Number of States

5.1 Introduction

By Theorem 4.1, the existence of an \((S, n)\) encoder with anticipation \(\mathcal{A}\) implies the existence of some irreducible deterministic presentation \(G\) of \(S\) which can be split in \(\mathcal{A}\) rounds, satisfying conditions \((i)-(v)\) of the theorem. In Section 4.2, we gave an intuitive argument why there might be cases in which the Shannon cover of \(S\) cannot satisfy the conditions of Theorem 4.1. In such cases, the graph \(G\), whose existence is guaranteed by Theorem 4.1, is obtained from the Shannon cover by a (proper) expansion of states.

In Section 5.2, we present a specific example in which the Shannon cover of a system cannot be split according to conditions \((i)-(v)\) of Theorem 4.1. Yet, another graph presentation of the system, which is obtained from the Shannon cover by expansion through duplicating a single state, does split following those conditions. Thus, in some cases, the number of states in any graph \(G\) that satisfies conditions \((i)-(v)\) of Theorem 4.1 must be strictly greater than the number of states in the Shannon cover of the system. There are no known a priori upper bounds on the smallest number of states in any graph \(G\) that satisfies conditions \((i)-(v)\) of Theorem 4.1 (provided there exists such a graph). Still, in Section 5.3 we show how the graph \(G\) which was constructed in Section 4.2 can be reduced to other deterministic presentations \(G'\) and \(G''\) of \(S\), for which:

- The first four conditions of Theorem 4.1 are satisfied, i.e., each of the graphs \(G'\) and \(G''\) can be split in \(\mathcal{A}\) rounds while satisfying conditions \((i)-(iv)\) until an \((S, n)\) encoder is obtained.

- The number of states in \(G'\) and \(G''\) can be bounded from above by numbers which can be effectively computed from \(G_S\), \(n\), and \(\mathcal{A}\).

By condition \((v)\) of Theorem 4.1, the graph obtained from splitting \(G\) can turn into the original encoder by a reduction of strongly equivalent states. However, the \((S, n)\) encoders
which are obtained from splitting $G'$ and $G''$ do not necessarily have this property. In the $(S, n)$ encoder obtained after splitting $G''$, we can only guarantee that every state is equivalent (but not necessarily strongly equivalent) to some state of the original encoder, while in the graph $G'$, even this property is not guaranteed.

Nevertheless, it follows that the existence of $(S, n)$ encoders with anticipation $\mathcal{A}$ implies that at least one of those encoders can be obtained using the state splitting algorithm, starting with a deterministic presentation of $S$ whose number of states can be bounded from above by a number that can be effectively computed from $G_S$, $n$, and $\mathcal{A}$. Hence, the problem of verifying whether an $(S, n)$ encoder with anticipation $\mathcal{A}$ exists is a decidable problem.

5.2 Example Where Splitting Requires Expansion of $G_S$

In the following example, we consider a constraint $S$ and an integer $n$ such that the Shannon cover, $G_S$, has a unique $(A_{G_S}, n)$-approximate eigenvector, $\bar{x}$, whose components do not exceed $n^4$. Actually, $\bar{x}$ is a true eigenvector of $A_{G_S}$, associated with the eigenvalue $n$. We prove that $G_S$ cannot be split according to $\bar{x}$ in a way that satisfies the conditions of Theorem 4.1 for $\mathcal{A} = 4$. On the other hand, there is another deterministic presentation of $S$, $\mathcal{G}$, that is obtained by expansion through duplicating one state of $G_S$, and $\mathcal{G}$ can be split according to the corresponding expanded version of $\bar{x}$. In the splitting process of $\mathcal{G}$, the components of the resulting eigenvector after the $i$th round, $i = 1, 2, 3, 4$, do not exceed $n^{4-i}$. In particular, after the fourth round, the all-one vector is obtained. In each splitting, a state is split into no more than $n$ descendants.

The constraint $S$ is presented by the graph $G_S$ in Figure 5.1, which consists of the eleven states, $A$, $B$, $C$, $E$, $F$, $G$, $H$, $I$, $J$, $K$, and $L$. The subscripts that appear in the figure form an eigenvector, as will be explained right below.

We assume that all the edges of $G_S$ are distinctly labeled, so $G_S$ is deterministic and reduced. It is easy to verify that $G_S$ is also an irreducible graph. (In fact, most of our interest is in states $A$, $B$, and $C$, while the rest of the states were added mainly in order to make the graph irreducible and to enable its splitting after the expansion operation.) Hence, the graph $G_S$ in Figure 5.1 is the Shannon cover of $S$.

The Perron eigenvalue of the matrix $A_{G_S}$ is 2, and the weights that appear as subscripts in Figure 5.1 form an eigenvector $\bar{x}$ of $A_{G_S}$, associated with the eigenvalue 2. The maximal component of $\bar{x}$ is $2^4 = 16$. As the vector $\bar{x}$ contains a component equaling 1, $\bar{x}$ is the smallest (componentwise) among the positive integer eigenvectors of $A_{G_S}$ that are associated with the eigenvalue 2. Any other integer eigenvector is a multiple of $\bar{x}$ by some integer and therefore, $\bar{x}$ is the only $(A_{G_S}, 2)$-approximate eigenvector whose components do not exceed $2^4$. (See Section 2.2.)

We now explain why the splitting of $G_S$ according to the conditions of Theorem 4.1 is impossible. Under those conditions, a state should be split in the first round into no more than two states, and after the first round, all the components must not be greater than $2^3 = 8$. To satisfy these requirements, states $A$, $B$, and $C$, whose weights are greater than
Figure 5.1: Graph $G_S$
Figure 5.2: Splitting of states A and B in the first round

8, must be split in the first round. Looking at Figure 5.1, the only possibility for state A to split in the first round is by having two descendants of weight 8, \(A_8^1\) and \(A_8^2\). The dashed edges in Figure 5.2 (a) illustrate the only possible distribution of the outgoing edges of A between its two descendant states. The only possibility for state B to split in the first round is to have two descendants of weight 8, \(B_8^1\) and \(B_8^2\), with a distribution of the outgoing edges between them as shown in Figure 5.2 (b).

State C has two possible options how to split in the first round: According to the first option, C can split into \(C_{10}^1\) and \(C_{10}^2\), as in Figure 5.3 (a). The second option for C is to split into \(C_{5}^1\) and \(C_{5}^2\), as in Figure 5.3 (b). Note that the states \(E_3\) and \(F_5\) cannot be split in the first round, as each of them has a single outgoing edge.

The local outgoing pictures of states \(A_8^2\) and \(B_8^2\) after the first round thus look like those in Figure 5.4 (a) or like those in Figure 5.4 (b), depending on the way C is split. If we choose the first option, then, in the second round, it is impossible to split state \(A_8^2\) into two states with weights not greater than 4 = \(2^2\). If the second option is chosen for the first round, then, in the second round, state \(A_8^2\) can be split as required, but state \(B_8^2\) cannot.

Thus, \(G_S\) cannot be split in four rounds that satisfy the conditions of Theorem 4.1. In order to have a splitting which does satisfy the conditions, \(A_8^2\) would “need” the splitting of C into \(C_{5}^1\) and \(C_{5}^2\), so that \(A_8^2\) could split in the second round, while \(B_8^2\) would “need” C to be split into \(C_{10}^1\) and \(C_{10}^2\). This leads us to duplicate state C and get a new presentation \(\hat{G}\) of S. The difference between \(\hat{G}\) and \(G_S\) is as follows:

1. \(\hat{G}\) has an extra state – \(D_{10}\) – whose local outgoing picture is identical to that of C, including the labels on the edges.

2. The edge that goes in \(G_S\) from \(B_{16}\) to \(C_{10}\), is replaced in \(\hat{G}\) by an edge from \(B_{16}\) to \(D_{10}\).

The graph \(\hat{G}\) is therefore a deterministic and irreducible presentation of S, but, unlike \(G_S\), the graph \(\hat{G}\) can be split according to the requirements in the first two rounds: In the
Figure 5.3: Splitting options for state $C$ in the first round

Figure 5.4: Possible outgoing pictures of descendants after the first round
Figure 5.5: Outgoing picture after the first round of the splitting of $\mathcal{G}$

Figure 5.6: Second round of the splitting of $\mathcal{G}$
The first round of the splitting of $G$, $C$ is split into $C_1^1$ and $C_2^2$, and $D$ is split into $D_1^1$ and $D_2^2$. The outgoing pictures of states $A_{s}^2$ and $B_{s}^2$ after the first round then look like in Figure 5.5. Thus, in the second round, the states $A_{s}^2$ and $B_{s}^2$ can both be split into states of weight 4. The way in which $A_{s}^2$ and $B_{s}^2$ distribute their edges among their descendants is illustrated in Figure 5.6.

According to a check we have done, $G$ can be fully split in four rounds, while the weights of the states in each round decrease from 16 to 8, 4, 2, and 1, and in each splitting a state is split into two descendants at most. It follows that an $(S, n)$ encoder does exist whose anticipation is not greater than 4, even though $G_S$ cannot serve as the graph $G$ in Theorem 4.1.

(We point out that in the our example, $G_S$ cannot replace $G$ in Theorem 4.1 even if condition (iii) of the theorem were omitted.)

5.3 Upper Bounds on the Number of States

5.3.1 First Upper Bound – The Graph $G'$

In this section, we define the graph $G'$, which is obtained from the graph $G$ of Section 4.2 by a reduction of states. The splitting properties of $G'$ are similar to those of $G$, but the number of states in $G'$ can be bounded from above. The properties of $G'$ are summarized in the following theorem.

**Theorem 5.1** Let $S$ be an irreducible constrained system and let $n$ be a positive integer where $C(S) \geq \log n$. Let $G_S$ be the Shannon cover of $S$ and $N$ be the number of states in $G_S$. Suppose that there exists some irreducible $(S, n)$ encoder whose anticipation is $A$. Then there exists an irreducible deterministic presentation $G'$ of $S$, whose number of states does not exceed $N(1 + n^{A+(A-1)n^A})$, and $G'$ satisfies the following:

(i) There is an $(A_G', n)$-approximate eigenvector, $\tilde{c}'$, whose largest component is not greater than $n^A$.

(ii) The graph $G'$ can be fully split according to $\tilde{c}'$ in $A$ rounds of splitting. An $(S, n)$ encoder, $E_{G'}$, whose anticipation is not greater than $A$, is obtained after deleting excess edges.

(iii) In each round of splitting, each of the states is split into no more than $n$ states.

(iv) In the $i$th round of splitting, the components of the resulting approximate-eigenvector, $\tilde{c}'^{(i)}$, are not greater than $n^{A-i}$.

**Remark:** Unlike what we had in condition (v) of Theorem 4.1, in this case we do not guarantee that the resulting encoder, $E_{G'}$, can be reduced to the original encoder, $E$. In other words, it is not guaranteed that every $(S, n)$ encoder $E$ of anticipation $A$ can be obtained
by state splittings and state reductions, starting with an irreducible deterministic presentation of $S$ in which the number of states is bounded from above by $N(1 + n^{A+|A-1|n^A})$. Nevertheless, it is guaranteed that some $(S, n)$ encoder, $E_{G'}$, can be obtained by $A$ rounds of splitting from the graph presentation $G'$ of $S$ whose number of states is bounded from above by $N(1 + n^{A+|A-1|n^A})$.

In addition, it is not guaranteed that the anticipation of $E_{G'}$ is equal to the anticipation $A$ of $E$. At any rate, the anticipation of $E_{G'}$ cannot be greater than $A$, since only $A$ rounds of splitting are needed to get $E_{G'}$ from the deterministic graph $G'$.

The proof of Theorem 5.1 which is given below involves the following definitions. Let $G$ be the graph defined in Section 4.2, and let $\bar{e}$ be the $(A_G, n)$-approximate eigenvector which is defined by Equation (4.5). Due to Theorem 4.1, $G$ can be fully split in $A$ rounds, so we denote by $G^{(i)}$ the graph obtained in the $i$th round of the splitting of $G$. For every state $v \in V(G_S)$, all the states of the form $(v', v)$ in $G$ are equivalent states that satisfy $F_G((v', v)) = F_G(v)$, even though the follower sets in $G_G$ of the various states $v'$ are not necessarily equal.

Let $v$ be a state of $G_S$ and let $\{(v'_1, v), \ldots, (v'_k, v)\}$ be a maximal set of states in $G$ such that the following holds:

- The respective weights of $(v'_1, v), \ldots, (v'_k, v)$ in $\bar{e}$ are all equal.
- Throughout the $A$ rounds of the splitting of $G$, the states $(v'_1, v), \ldots, (v'_k, v)$ and their respective descendants in the graphs $G^{(i)}$ are split into identical number of states, with the same weights in the resulting approximate eigenvectors.

(The maximality of the set $\{(v'_1, v), \ldots, (v'_k, v)\}$ means that it is impossible to add another $v_i$ so that the two previous conditions hold.) By reducing every such maximal set of equivalent states in $G$ into a single representative of the set, we get a new deterministic presentation of $S$. By taking an irreducible sink of the resulting graph, we obtain a new irreducible deterministic presentation of $S$, which we denote by $G'$. More formally, for every pair of states $(v', v)$ and $(v'', v)$ in $G$, where $(v'', v)$ is reduced to $(v', v)$ while constructing $G'$, we require that:

- The components of $(v'', v)$ and $(v', v)$ in $\bar{e}$ are equal.
- For every integer $i$ such that $0 \leq i \leq A$, the number of descendant states of $(v'', v)$ in $G^{(i)}$ is equal to the number of descendant states of $(v', v)$ in $G^{(i)}$. A function $\phi^{(i)}_{(v'', v)}$, which is one to one and onto, can be defined from the set of descendant states of $(v'', v)$ in $G^{(i)}$ to the set of descendant states of $(v', v)$ in $G^{(i)}$, where two states are mapped one to the other by $\phi^{(i)}_{(v'', v)}$ only if they have equal components in $\bar{e}^{(i)}$.
- Let $(v'', v)^{(i:z)}$ be a descendant state of $(v'', v)$ in $G^{(i)}$ and suppose that $\phi^{(i)}_{(v'', v)}((v'', v)^{(i:z)}) = (v', v)^{(i:r)}$. Then the descendant states of $(v'', v)^{(i:z)}$ in $G^{(i+1)}$ are mapped by $\phi^{(i+1)}_{(v'', v)}$ to the descendant states of $(v', v)^{(i:r)}$ in $G^{(i+1)}$. 59
A state $(u', u) \in G$ which remains also in $G'$, will be denoted in $G'$ by $(U', U) \in V(G')$. The $(A_{G'}, u)$-approximate eigenvector, $\varphi'$, is defined so that the component of a state $(U', U)$ in $G'$ is equal to the respective component of $(u', u)$ according to $\varphi$. It is easy to verify that $\varphi'$ is indeed an $(A_{G'}, u)$-approximate eigenvector. We claim that $G'$ can be split in $A$ rounds, consistently with $\varphi'$, while satisfying the conditions of Theorem 5.1. The way in which a state $(U', U)$ is split in $G'$ is similar to the way in which $(u', u)$ is split in $G$, as we state in Lemma 5.2 below. The chart in Figure 5.7 illustrates the connections between the descendant states of $(U', U)$ in $G^{(i)}$ and those of $(u', u)$ in $G^{(6)}$, according to Lemma 5.2.

**Lemma 5.2** The graph $G'$ can be split in $A$ rounds, consistently with the vector $\varphi'$, in such a way that after the $i$th round of splitting, $0 \leq i \leq A$, the resulting graph, $G^{(i)}$, and its corresponding approximate eigenvector, $\varphi^{(i)}$, satisfy the following:

1. For every state $(U', U)$ of $G'$, the number of descendant states in $G^{(6)}$ is equal to the number of descendant states of $(u', u)$ in $G^{(6)}$.

2. A function $\Psi^{(i)}_{(u', u)}$, which is one to one and onto, can be defined from the descendant states of $(U', U)$ in $G^{(i)}$ to the descendant states of $(u', u)$ in $G^{(6)}$, so that a state $(U', U)^{(i)}$ in $G^{(i)}$ and the state $(u', u)^{(6)} = \Psi^{(6)}_{(u', u)}((U', U)^{(i)})$ in $G^{(6)}$ have equal weights and equal outdegrees.

3. A function $\psi^{(i)}_{(u', u)}$, which is one to one and onto, can be defined from the outgoing edges of the descendants of $(U', U)$ in $G^{(i)}$ to the outgoing edges of the descendants of $(u', u)$ in $G^{(6)}$, in such a way that:

   a) If $e_1, \ldots, e_m$ are the outgoing edges of $(U', U)^{(i)}$ in $G^{(i)}$, then the outgoing edges of $(u', u)^{(6)}$ in $G^{(6)}$ are $\psi^{(6)}_{(u', u)}(e_1), \ldots, \psi^{(6)}_{(u', u)}(e_m)$.

   b) Two edges which are mapped one to the other by $\psi^{(i)}_{(u', u)}$ carry identical labels.
(e) Let \( e \) be an outgoing edge of \( (U', U)^{(i+1)} \) in \( G^{(i)} \) and let \( (V', V)^{(i+1)} \) be the target state of \( e \). The outgoing edge \( \psi^{(i)}_{v(w', u)}(e) \) of \( (w', u)^{(i+1)} \) in \( G^{(i+1)} \) has a target state of the form \( (v'', v)^{(i+1)} \), where the state \( (v'', v) \) in \( G \) is reduced to \( (v', v) \) while constructing \( G' \). Moreover, the function \( \psi^{(i)}_{v(w', u)} \) maps state \( (v'', v)^{(i+1)} \) of \( G^{(i)} \) to state \( (v', v)^{(i+1)} \) in \( G^{(i)} \), where \( (v', v)^{(i+1)} = \Psi^{(i)}_{v(w', u)}(V', V)^{(i+1)} \).

(Due to the definition of \( \phi^{(i)}_{v(w', u)} \) and due to the assumption that \( (V', V)^{(i+1)} \) of \( G^{(i)} \) and \( (v', v)^{(i+1)} \) of \( G^{(i+1)} \) have equal weights, we get that the target states of two edges which are mapped one to the other by \( \psi^{(i)}_{v(w', u)} \) have equal weights according to \( \epsilon^{(i)} \) and \( \sigma^{(i)} \).)

**Proof:** The proof of Lemma 5.2 is carried out by induction on the number of rounds, \( i \), where \( 0 \leq i \leq A \). By the definition of the vector \( \epsilon' \), the component of a state \( (U', U) \) in \( \epsilon' \) is equal to the component of \( (w', u) \) in \( \epsilon \). By the construction of \( G' \), every state \( (U', U) \) appears in \( G' \) with the same number of outgoing edges and the same set of labels as \( (w', u) \) does in \( G \). Consequently, the function \( \Psi^{(0)}_{v(w', u)} \) maps state \( (U', U) \) in \( G' \) to state \( (w', u) \) in \( G \).

The outgoing edges of \( (U', U) \) in \( G' \) can be mapped by a function \( \psi^{(0)}_{v(w', u)} \) to the outgoing edges of \( (w', u) \) in \( G \). The function \( \psi^{(0)}_{v(w', u)} \) is then one to one and onto and it preserves the labeling of the edges. The construction of \( G' \) also implies that if the target state of an edge \( e \) in \( G' \) is \( (V', V) \), then the target state of the edge \( \psi^{(0)}_{v(w', u)}(e) \) in \( G \) is a state \( (v'', v) \), which is reduced to \( (v', v) \) while constructing \( G' \). Hence, Lemma 5.2 holds for \( i = 0 \).

We now assume the validity of the lemma for the first \( i \) rounds, where \( 0 \leq i \leq A - 1 \), and prove it for the \((i + 1)\)st round. Let \((U', U)\) be a state of \( G' \) and let \((U', U)^{(i+1)}\) be any of the descendant states of \((U', U)\) in \( G^{(i+1)} \). By the induction hypothesis, the component of \((U', U)^{(i+1)}\) in \( \epsilon^{(i+1)} \) is equal to the respective component of \((w', u)^{(i+1)}\) in \( \epsilon^{(i+1)} \). The outgoing neighborhood of \((U', U)^{(i+1)}\) in \( G^{(i+1)} \) is identical to the one of \((w', u)^{(i+1)}\) in \( G^{(i+1)} \), in terms of the number of neighbors and their respective components in the approximate eigenvectors.

Suppose that the state \((w', u)^{(i+1)}\) of \( G^{(i+1)} \) is split in the \((i+1)\)st round into \( k \) states, where the \( j \)th descendant state inherits the edges \( \psi^{(i)}_{v(w', u)}(e_1), \ldots, \psi^{(i)}_{v(w', u)}(e_m) \). Then \((U', U)^{(i+1)}\) of \( G^{(i+1)} \) can also be split into \( k \) states, where the \( j \)th descendant state inherits the edges \( e_1, \ldots, e_m \). The labels on the edges inherited by the \( j \)th descendant state of \((U', U)^{(i+1)}\) are the same labels which appear on the edges inherited by the \( j \)th descendant of \((w', u)^{(i+1)}\).

The weight of the state \((U', U)^{(i+1)}\) in \( G^{(i+1)} \) can be distributed among its \( k \) descendants, so that the component of the \( j \)th descendant in \( \epsilon^{(i+1)} \) is equal to the component of the \( j \)th descendant of \((w', u)^{(i+1)}\) in \( \epsilon^{(i+1)} \). The components in \( \epsilon^{(i+1)} \) which correspond to the target states of \( e_1, \ldots, e_m \), are equal to the components in \( \epsilon^{(i+1)} \) which correspond to the target states of \( \psi^{(i)}_{v(w', u)}(e_1), \ldots, \psi^{(i)}_{v(w', u)}(e_m) \). Therefore, if the distribution of weights among the descendants of \((w', u)^{(i+1)}\) in \( G^{(i+1)} \) is consistent with \( \epsilon^{(i+1)} \), then the distribution of weights among the descendants of \((U', U)^{(i+1)}\) in \( G^{(i+1)} \) is consistent with \( \epsilon^{(i+1)} \). The function \( \Psi^{(i+1)}_{v(w', u)} \) maps the \( j \)th descendant state of \((U', U)^{(i+1)}\) in \( G^{(i+1)} \) to the \( j \)th descendant state of \((w', u)^{(i+1)}\) in \( G^{(i+1)} \).
It is left to prove that for every descendant state of \((U', U)^{(i+1)}\), the neighborhood in 
\(G'^{(i+1)}\) is identical to the neighborhood of the corresponding descendant of \((u', u)^{(i)}\) in 
\(G^{(i+1)}\), in the number of outgoing edges and the weights of their target states. To prove 
that, it is sufficient to show that the target states of the edges \(e_1, \ldots, e_m\) are split the same 
way as the target states of \(\psi_{(u', u)}^{(i)}(e_1), \ldots, \psi_{(u', u)}^{(i)}(e_m)\) are, i.e., into the same numbers of 
descendant states, with equal multisets of weights.

Suppose that a state \((V', V)^{(i+1)}\) is the target state of the edge \(e_r\), which is an outgoing 
edge of \((U', U)^{(i+1)}\) in \(G^{(i+1)}\). Then, by the induction hypothesis, the target state of \(\psi_{(u', u)}^{(i)}(e_r)\) 
in \(G^{(i)}\) is a state of the form \((v'', v)^{(i)}\), which is mapped by \(\phi_{(v'' ,v)}^{(i)}\) to the state \((v', v)^{(i+1)}\) 
of \(G^{(i+1)}\).

By the definition of \(\phi_{(v'', v)}^{(i)}\), the number of descendant states of \((v'', v)^{(i)}\) in \(G^{(i+1)}\) is 
equal to the number of descendant states of \((v', v)^{(i+1)}\) in \(G^{(i+1)}\). The descendant states of 
\((v'', v)^{(i)}\) in \(G^{(i+1)}\) are mapped by \(\phi_{(v'', v)}^{(i+1)}\) to the descendant states of \((v', v)^{(i+1)}\) in \(G^{(i+1)}\), 
and the two sets of states have identical multisets of weights.

Using the same arguments that were presented in the case of \((U', U)^{(i+1)}\), we can prove 
that the state \((V', V)^{(i+1)}\) in \(G^{(i+1)}\) is split in the \((i+1)\)st round to the same number of 
states, with the same multiset of weights, as \((v', v)^{(i+1)}\) is split in \(G^{(i+1)}\). As a result, the state 
\((V', V)^{(i+1)}\) in \(G^{(i+1)}\) splits as the state \((v'', v)^{(i)}\) does in \(G^{(i+1)}\). The function \(\psi_{(v'', v)}^{(i+1)}\) then maps 
every descendant edge of the edge \(e_r\) in \(G^{(i+1)}\) to a descendant edge of \(\psi_{(v'' ,v)}^{(i)}(e_r)\) in \(G^{(i+1)}\). 
The target state of the latter edge is mapped by \(\phi_{(v'' ,v)}^{(i+1)}\) to a state \((v', v)^{(i+1)}\) of \(G^{(i+1)}\), 
where the state \((V', V)^{(i+1)}\) appears in \(G^{(i+1)}\) as the target state of the former edge. This 
concludes the proof of Lemma 5.2.

We now prove Theorem 5.1.

**Proof:** Properties (i)-(iv) in Theorem 5.1 are implied by Lemma 5.2 and by the properties 
of \(G\) according to Theorem 4.1. To complete the proof of Theorem 5.1, it remains to 
show that the number of states in \(G'\) is bounded from above by \(N(1 + n^{A^1(A-1)n^A})\). The 
computation of this bound is based on the reduction procedure which is used to convert \(G\) 
into \(G'\).

Given a specific state \(u \in G_S\) and a specific weight \(w\), where \(0 \leq w \leq n^A\), there cannot 
be two distinct states \((U', U)\) and \((U'', U)\) of weight \(w\) in \(G'\) which are split throughout the 
\(A\) rounds to the same numbers of descendant states with the same multisets of weights. 
Therefore, the number of states \((U', U)\) of weight \(w\) in \(G'\) cannot be greater than the number 
of different ways to distribute the weight \(w\) among the descendents of \((U', U)\) throughout 
the \(A\) rounds of splitting, where in each round, each state is split into no more than \(n\) 
states. In addition, the weights of the descendents after the \(i\)th round are not greater than 
\(n^{A-i}\), for \(1 \leq i \leq A\).

The number of states in \(G'\) is therefore bounded from above by the solution to the 
following combinatorial problem: In how many different ways can we divide \(w\) identical 
elements among \(n\) identical buckets and then divide the elements of each bucket into \(n\)
smaller buckets, etc., where in the \(i\)th step, every bucket contains no more than \(n^{A-i}\) elements? In our calculations, we will allow repetitions of the same element in several buckets, which may only increase the upper bound. In addition, we ignore the fact that the buckets are identical and that so are the elements. These simplifications will yield a crude, but easy to compute bound.

Under those simplifications, the number of ways to distribute the contents of a bucket from the \(i\)th step is not greater than \(n^{A-i}\). Here we ignored the fact that every bucket in the \((i+1)\)st step should have at most \(n^{A-(i+1)}\) elements. The number of ways to simultaneously distribute the weights of all the buckets from the \(i\)th step is not greater than \((n^{A-i})^{n^i} = n^{n^A}\). In the last round, there is only one way to distribute the contents of the buckets, since only one element should get into each bucket. Consequently, the number of ways to distribute the \(w\) elements among the buckets throughout the \(A\) steps is not greater than \((n^{A})^{A-1} = n^{(A-1)n^A}\). This is also an upper bound on the number of ways to distribute the weight \(w\) of \((U', U)\) among its descendants throughout the \(A\) rounds.

The graph \(G'\) may contain states of the form \((U', U)\) for all possible weight values \(w\) between zero and \(n^{A}\). Notice that a state of weight zero has only one way in which it can be split throughout the \(A\) rounds. Thus, given a specific \(v \in G_S\), the maximum number of states of the form \((U', U)\) in the graph \(G'\) is not greater than \(1 + n^A \cdot n^{(A-1)n^A} = 1 + n^{A+(A-1)n^A}\).

Given that the number of states in \(G_S\) is \(N\), we conclude that the number of states in the irreducible deterministic presentation \(G'\) of \(S\) is not greater than \(N(1 + n^{A+(A-1)n^A})\). This completes the proof of Theorem 5.1.

### 5.3.2 Second Upper Bound – The Graph \(G''\)

Given an \((S, n)\) encoder \(E\) with anticipation \(A\), we now define a second graph \(G''\), which is also obtained from \(G\) by a reduction of equivalent states. The graph \(G''\) has splitting properties which are also similar to those of \(G\), but it contains more states than \(G'\) does. Still, the number of states in \(G''\) can be bounded from above. At the end of the splitting of \(G''\), an \((S, n)\) encoder \(E_{G''}\) is obtained in which every state is equivalent to a state in the original encoder, \(E\) (but not necessarily strongly-equivalent). In fact, an \((S_E, n)\) encoder is obtained after splitting \(G''\), where \(S_E = S\{E\}\).

**Theorem 5.3** Let \(S\) be an irreducible constrained system and let \(n\) be a positive integer where \(C(S) \geq \log n\). Let \(G_S\) be the Shannon cover of \(S\), let \(N\) be the number of states in \(G_S\), and let \(d\) be the maximum outdegree of a state in \(G_S\). Suppose that there exists some irreducible \((S, n)\) encoder, \(E\), whose anticipation is \(A\). Then there exists an irreducible deterministic presentation \(G''\) of \(S\), whose number of states is bounded from above by

\[
N \cdot \left(\left(n^A + 1\right)^{(d+1)A} \cdot n^{((d+1)^A-1)} \cdot \frac{A+1}{(n-1)!}\right)
\]

and the graph \(G''\) satisfies the following:

(i) There is an \((A_{G''}, n)\)-approximate eigenvector, \(\bar{v}'\), whose largest component is not greater than \(n^A\).
(ii) The graph $G''$ can be fully split according to $\mathcal{E}$ in $A$ rounds of splitting. After deleting excess edges, we obtain an $(S, n)$ encoder, $\mathcal{E}_{G''}$, in which every state is equivalent to a state of $E$. The anticipation of $\mathcal{E}_{G''}$ is not greater than $A$.

(iii) In each round of the splitting of $G''$, each of the states is split into no more than $n$ states.

(iv) In the $i$th round of the splitting of $G''$, the components of the resulting approximate-eigenvector, $\mathcal{E}^{(i)}$, are not greater than $n^{A-1}$.

The construction of $G''$ and the proof of Theorem 5.3 are similar to the construction of $G'$ out of $G$ and the proof of Theorem 5.1. Let $G$ be the graph defined in Section 4.2 and let $\mathcal{E}$ be the $(A_G, n)$-approximate eigenvector, defined by Equation (4.5). While the graph $G'$ is obtained from $G$ by reducing all the states $(v', v)$ whose respective descendants have equal weights, the graph $G''$ is obtained from $G$ by reducing all the states $(v', v)$ whose respective descendants are equivalent states.

More formally, let $v$ be a state of $G_S$ and let $\{(v'_1, v), \ldots, (v'_k, v)\}$ be a maximal set of states in $G$ such that:

- The components of $(v'_1, v), \ldots, (v'_k, v)$ in $\mathcal{E}$ are all equal.
- Throughout the $A$ rounds of the splitting of $G$, the states $(v'_1, v), \ldots, (v'_k, v)$ and their respective descendants in the graphs $G^{(i)}$ are split into identical number of states, with the same weights and the same sets of follower sets.

By reducing every such maximal set of equivalent states in $G$ into a single representative of the set and then taking an irreducible sink of the resulting graph, we obtain an irreducible deterministic presentation of $S$, which we denote by $G''$.

A pair of states $(v', v)$ and $(v'', v)$ in $G$ are said to split equivalently if $(v'', v)$ is reduced to $(v', v)$ while constructing $G''$. Such a pair of states has the following properties:

1. The components of $(v'', v)$ and $(v', v)$ in $\mathcal{E}$ are equal.

2. For every integer $i$ such that $0 \leq i \leq A$, the number of descendant states of $(v'', v)$ in $G^{(i)}$ is equal to the number of descendant states of $(v', v)$ in $G^{(i)}$. A function $\phi^{(i)}_{(v'', v)}$, which is one to one and onto, can be defined from the descendant states of $(v'', v)$ in $G^{(i)}$ to the descendant states of $(v', v)$ in $G^{(i)}$, where two states are mapped one to the other by $\phi^{(i)}_{(v'', v)}$ only if they have equal components in $\mathcal{E}^{(i)}$.

3. In addition, every two states which are mapped one to the other by $\phi^{(i)}_{(v'', v)}$ are equivalent states, i.e., have the same follower set.

4. Let $(v'', v)^{(i+1)}$ be a descendant state of $(v'', v)$ in $G^{(i)}$ and suppose that $\phi^{(i)}_{(v'', v)}((v'', v)^{(i+1)}) = (v', v)^{(i+1)}$. Then the descendant states of $(v'', v)^{(i+1)}$ in $G^{(i+1)}$ are mapped by $\phi^{(i+1)}_{(v'', v)}$ to the descendant states of $(v', v)^{(i+1)}$ in $G^{(i+1)}$. 

64
We denote by \((U', U)\) the appearance of a \(G\)-state \((u', u)\) in the graph \(G'\). An \((A_{G'}, u)\)-approximate eigenvector \(\tilde{v}'\) is defined in which the component corresponding to a state \((U', U)\) of \(G'\) is equal to the respective component of \((u', u)\) in \(\tilde{v}\). As in the case of \(G'\), a state \((U', U)\) of \(G'\) splits as \((u', u)\) does in \(G\), as stated in the following lemma:

**Lemma 5.4** The graph \(G'\) can be split in \(A\) rounds, consistently with the vector \(\tilde{v}'\), in such a way that after the \(i\)th round of splitting, \(0 \leq i \leq A\), the resulting graph, \(G'^{(i)}\), and its corresponding approximate eigenvector, \(\tilde{v}'^{(i)}\), satisfy the following:

1. For every state \((U', U)\) of \(G'\), the number of descendant states in \(G'^{(i)}\) is equal to the number of descendant states of \((u', u)\) in \(G^{(i)}\).

2. A function \(\Psi_{(u', u)}^{(i)}\), which is one to one and onto, can be defined from the descendant states of \((U', U)\) in \(G'^{(i)}\) to the descendant states of \((u', u)\) in \(G^{(i)}\), so that a state \((U', U)^{(i)}\) in \(G'^{(i)}\) and the state \((u', u)^{(i)}\) = \(\Psi_{(u', u)}^{(i)}\)\((U', U)^{(i)}\) in \(G^{(i)}\) have equal weights, equal outdegrees, and equal follower sets.

3. A function \(\psi_{(u', u)}^{(i)}\), which is one to one and onto, can be defined from the outgoing edges of the descendants of \((U', U)\) in \(G'^{(i)}\) to the outgoing edges of the descendants of \((u', u)\) in \(G^{(i)}\), in such a way that:

   a) If \(e_1, \ldots, e_m\) are the outgoing edges of \((U', U)^{(i)}\) in \(G'^{(i)}\), then the outgoing edges of \((u', u)^{(i)}\) in \(G^{(i)}\) are \(\psi_{(u', u)}^{(i)}(e_1), \ldots, \psi_{(u', u)}^{(i)}(e_m)\).

   b) Two edges which are mapped to the other by \(\psi_{(u', u)}^{(i)}\) carry identical labels.

   c) Let \(e\) be an outgoing edge of \((U', U)^{(i)}\) in \(G'^{(i)}\) and let \((V', V)^{(i)}\) be the target state of \(e\). The outgoing edge \(\psi_{(u', u)}^{(i)}(e)\) of \((u', u)^{(i)}\) in \(G^{(i)}\) has a target state of the form \((v', v)^{(i)}\), where the state \((v', v)\) in \(G\) is reduced to \((v', v)\) while constructing \(G'\). Moreover, the function \(\phi_{(u', u)}^{(i)}\) maps state \((v', v)^{(i)}\) of \(G^{(i)}\) to state \((v', v)^{(i)}\) of \(G^{(i)}\) (By the definition of \(\phi_{(u', u)}^{(i)}\) and by using the induction assumption for the state \((V', V)^{(i)}\), we get that the target states of two edges which are mapped one to the other by \(\psi_{(u', u)}^{(i)}\) have identical follower sets in \(G^{(i)}\) and \(G'^{(i)}\). They also have equal components in \(\tilde{v}^{(i)}\) and \(\tilde{v}'^{(i)}\)).

**Proof:** The proof of Lemma 5.4 is almost the same as the proof of Lemma 5.2, with the following supplements:

Suppose that a state \((u', u)^{(i)}\) of \(G^{(i)}\) is split in the \((i + 1)\)st round into \(k\) states, where the \(j\)th descendant state inherits the edges \(\psi_{(u', u)}^{(i)}(e_1), \ldots, \psi_{(u', u)}^{(i)}(e_m)\), and suppose that state \((U', U)^{(i)}\) of \(G'^{(i)}\) is split into \(k\) states, where the \(j\)th descendant state inherits the edges \(e_1, \ldots, e_m\). Since the edges inherited by the \(j\)th descendants have the same labels and have target states with the same follower sets in both \(G^{(i)}\) and \(G'^{(i)}\), the \(j\)th descendant
state of \((U', U)^{(i)}\) in \(G^{i+1}\) and the \(j\)th descendant state of \((u', u)^{(i)}\) in \(G^{i+1}\) have equal follower sets.

Similarly to \((U', U)^{(i)}\), a state \((V', V)^{(i+1)}\), which is the target state of the edge \(e_r \in \{e_1, \ldots, e_m\}\), splits in the \((i + 1)\)st round of \(G^i\) to the same number of states, with the same sets of follower sets, as \((v', v)^{(i+1)}\) does in the \((i + 1)\)st round of \(G\). By the definition of the function \(\phi^{(i)}_{(v', v)}\), the state \((v'', v)^{(i+1)}\), which is the target state of \(\psi^{(i)}_{(v', v)}(e_r)\) in \(G^{i+1}\), splits the same way as state \((v', v)^{(i)}\) does in \(G^{i}\). Hence, the target state of the edge \(e_r\) in \(G^{i+1}\) is split to the same number of states, with the same set of follower sets, as the target state of \(\psi^{(i)}_{(v', v)}(e_r)\) does in \(G^{i}\). The function \(\psi^{(i+1)}_{(v', v)}\) can therefore be defined so that two edges are mapped one to the other only if they carry the same label and their target states are of the same weight and have the same follower set. Recalling the definition of \(i\)-strong equivalence from Section 2.7, we point out that the states \((U', U)^{(i)}\) and \((u', u)^{(i)}\) are 1-strongly equivalent.

We now prove Theorem 5.3.

**Proof:** We need to prove that every state of the encoder \(E_{G^i}\), which is obtained after splitting \(G^i\), is equivalent to some state of the original encoder \(E\) and vice versa. Every state of \(E_{G^i}\) is a state \((U', U)^{(i)}\) of \(G^{i+1}\) which is equivalent to the state \((u, u)^{(i)}\) of \(G^{i+1}\) which, in turn, is equivalent to some state of the original encoder \(E\). On the other hand, every state of \(E\) is equivalent to some \((v', v)^{(i)}\) in \(G^{i+1}\), and \((v', v)^{(i)}\) is equivalent to some state \((v', v)^{(i)}\) in \(G^{i+1}\), where state \((v', v)^{(i)}\) of \(G\) is reduced to \((v', v)^{(i)}\) while constructing \(G^i\). Finally, state \((v', v)^{(i)}\) is equivalent to \((v', v)^{(i+1)}\) in \(G^{i+1}\). Therefore, \(S(E_{G^i}) = S(E) = S_{G^i}\) and \(E_{G^i}\) is an \((S_{E}, n)\) encoder.

The rest of properties (i)-(iv) in Theorem 5.3 are implied by Lemma 5.4, and it is left to prove the upper bound on the number of states in \(G^i\).

**Definition 5.1** A pair of states \((v', v)\) and \((v'', v)\) in \(G\) are said to split equivalently in the first \(t\) rounds, if the following holds:

- For every integer \(i\) such that \(0 \leq i \leq t\), the number of descendant states of \((v'', v)\) in \(G^{(i)}\) is equal to the number of descendant states of \((v', v)\) in \(G^{(i)}\).

- A function \(\phi^{(i)}_{(v', v)}\), which is one to one and onto, can be defined from the descendant states of \((v'', v)\) in \(G^{(i)}\) to the descendant states of \((v', v)\) in \(G^{(i)}\), where two states are mapped one to the other by \(\phi^{(i)}_{(v', v)}\) only if they are equivalent states and the stay also have equal components in \(e^{(i)}\).

- Let \((v'', v)^{(i+1)}\) be a descendant state of \((v'', v)\) in \(G^{(i)}\) and suppose that \(\phi^{(i)}_{(v'', v)}((v'', v)^{(i+1)}) = (v', v)^{(i+1)}\). In case that \(i \leq (t - 1)\), the descendant states of \((v'', v)^{(i+1)}\) in \(G^{(i+1)}\) are mapped by \(\phi^{(i+1)}_{(v'', v)}\) to the descendant states of \((v', v)^{(i+1)}\) in \(G^{(i+1)}\).
We examine the conditions under which two states of \( G \) split equivalently in the first \( t \) rounds. Two states \((v', v)\) and \((v'', v)\) in \( G \) split equivalently in the zero round simply if they have the same weight, i.e., equal components in \( \bar{e} \). The two states split equivalently also in the first round if:

- Their respective neighbors in \( G \) have the same weights according to \( \bar{e} \), i.e., the respective neighbors split equivalently in the zero round, and
- \((v', v)\) and \((v'', v)\) are split to the same number of descendant states in the first round of \( G \), and they distribute their respective edges among their descendants in the same manner.

Recall that for every state \((v', v)\) of \( G \), which is not a \( \emptyset \)-state (i.e., its weight is greater than zero), the sum of weights in its neighborhood is equal to \( v \) times its weight. Therefore, the way in which the edges of \((v', v)\) are distributed among its descendants also implies the distribution of the weight among its descendants. Also recall that we remove from \( G \) all the zero-weight states, including their edges, before starting the splitting process, so we do not have to take into account the distribution of these edge among the descendant states.

In general, the following conditions are sufficient to guarantee that \((v', v)\) and \((v'', v)\) split equivalently in the first \( t \) rounds:

1. \((v', v)\) and \((v'', v)\) split equivalently in the first \((t - 1)\) rounds.
2. Their respective neighbors in \( G \) split equivalently in the first \((t - 1)\) rounds.
3. Descendants \((v'', v)^{(t-1);s}\) in \( G^{(t-1)} \), which are mapped by \( \phi_{(v'';v)}^{(t-1)} \) to the respective descendants \((v', v)^{(t-1);r}\), have the same number of outgoing edges as the states \((v', v)^{(t-1);r}\) do. Respective edges carry the same labels and go into states which have identical weights and identical follower sets.
4. In round \( t \), state \((v', v)^{(t-1);r}\) and state \((v'', v)^{(t-1);s}\) of \( G^{(t-1)} \) are split to the same number of states, where two edges of \((v', v)^{(t-1);r}\) are inherited by one descendant state if and only if the respective edges of \((v'', v)^{(t-1);s}\) are also inherited by the same descendant state.

Given a state \( v \in V(G_S) \), consider the largest set of states in \( G \) of the form \((v', v)\), where no two states split equivalently in the zero round. The number of such states is not greater than the number of distinct weights in \( G \), i.e., \((1 + n^4) \). Notice that the maximum outdegree of a state in \( G \) is equal to the maximum outdegree in \( G_S \) which is \( d \). Given \( v \in V(G_S) \), let \( f(t) \) represent the maximum number of states \((v', v)\) in \( G \) such that no two split equivalently in the first \( t \) rounds. By the previous analysis, \( f(t) \) is not greater than \( f(t - 1) \) multiplied by \( (f(t - 1))^d \), multiplied by the number of different ways to distribute the edges of the descendants in \( G^{(t-1)} \) among the descendants in \( G^{(t)} \).

After deleting the edges which enter zero-weight states in \( G \), a state \((v', v)\) has a set of edges which corresponds to that of \( v' \) in \( G_S \). The edges of \( v' \) in \( G_S \) are defined by the \( E \)-edges that go out of the terminal \( E \)-states of the paths in \( v' \). The number of terminal \( E \)-states
in a state of \( G^{(t-1)} \) is not greater than \( n^{A-(t-1)} \) and the number of \( E \)-edges going out of them is not greater than \( n \cdot n^{A-(t-1)} = n^{A+1-(t-1)} \). Hence, the number of outgoing edges of a state in \( G^{(t-1)} \) or in \( G^{(t-1)}(v) \) cannot be greater than \( n^{A+1-(t-1)} \). These edges should be distributed among no more than \( n \) descendant states. Given the local neighborhood of a state in \( G^{(t-1)} \), the number of different ways to distribute its outgoing edges among its \( n \) descendants in round \( t \) does not exceed \( n^{n^{A+1-(t-1)}} \). The number of descendant states of \((v', v)\) in \( G^{(t-1)} \) is not greater than \( n^{t-1} \), so the total number of different ways for the descendants of \((v', v)\) to be split in round \( t \) is not greater than \( (n^{n^{A+1-(t-1)}})^{n^{t-1}} = n^{n^{A+1}} \).

We thus get the following recursive equation:

\[
    f(t) = f(t-1) \cdot (f(t-1))^{d} \cdot n^{A+1}
\]

(5.2)

\[
    f(0) = n^{A} + 1
\]

(5.3)

The solution of this equation is:

\[
    f(t) = (n^{A} + 1)^{(d+1)^t} \cdot n^{((d+1)^{t-1}) \cdot n^{A+1} / (d-1)}
\]

(5.4)

Given \( v \in V(G_S) \), the maximum number of states in \( G'' \) of the form \((v', v)\) is equal to the maximum number of states \((v', v)\) in \( G \) such that no two split equivalently in the \( A \) rounds of splitting. Hence, given \( v \in V(G_S) \), the maximum number of states \((v', v)\) in \( G'' \) is not greater than

\[
    f(A) = (n^{A} + 1)^{(d+1)^A} \cdot n^{((d+1)^{A-1}) \cdot n^{A+1} / (d-1)}
\]

(5.5)

The total number of states in \( G'' \) is not greater than \( N \) times \( f(A) \), where \( N \) is the number of states \( v' \) in \( G_S \).
Chapter 6

Second Lower Bound on the Anticipation

6.1 Statement and Proof of Theorem 6.1

We repeat the statement of the theorem that appears in Section 2.6 and then we give its proof in detail. We recall that by Lemma 3.4, every \((A_G, n)\)-approximate eigenvector for a given graph \(G\) is also an \((A_{G_k}, n^k)\)-approximate eigenvector, for every positive integer \(k\).

**Theorem 6.1** Let \(S\) be an irreducible constrained system, let \(n\) be a positive integer where \(C(S) \geq \log n\), and let \(G\) be any irreducible deterministic presentation of \(S\). Suppose that there exists some irreducible \((S, n)\) encoder whose anticipation is \(A\). Then there exists an \((A_G, n)\)-approximate eigenvector \(\bar{x}\), (which is also an \((A_{G_k}, n^k)\)-approximate eigenvector), such that the following holds:

1. The largest component of \(\bar{x}\) is not greater than \(n^A\).
2. For every integer \(k\) in the range \(1 \leq k \leq A\), the states of \(G^k\) can be split in one round consistently with the \((A_{G_k}, n^k)\)-approximate eigenvector \(\bar{x}\), such that the components of the resulting approximate eigenvector are not greater than \(n^{A-k}\) and each of the states in \(G^k\) is split into no more than \(n^k\) states.

We point out two important differences between Theorem 6.1 and Theorem 4.1: While Theorem 4.1 gives a necessary and sufficient condition on the existence of \((S, n)\) encoders, Theorem 6.1 gives only a necessary condition on the existence of encoders. However, the condition of Theorem 6.1 is given by means of every irreducible deterministic presentation of \(S\), particularly the Shannon cover, while the necessary condition of Theorem 4.1 holds only for some irreducible deterministic presentation of \(S\). The proof of Theorem 6.1 is as follows:

**Proof:** Let \(E\) be an \((S, n)\) encoder with anticipation \(A\). Let \(G_E\) be the graph defined in Section 3.1 and let \(\bar{\varepsilon}_E = [(\varepsilon_E)_Z]_{Z \in G_E}\) be the eigenvector of \(A_{G_E}\), associated with the
eigenvalue \( n \), as defined by Equation (3.4). By Lemma 3.3, the maximal component of the vector \( \tilde{c}_k \) is not greater than \( n^d \). According to Lemma 3.5, for each integer \( k \), where \( 1 \leq k \leq A \), the graph \( G_k^k \) can be split in one round consistently with \( \tilde{c}_k \), yielding a corresponding eigenvector whose components are not greater than \( n^{d-k} \). In this single round, each state of \( G_k^k \) is split into no more than \( n^k \) states.

Let \( G \) be any of the irreducible deterministic presentations of \( S \). The idea of the proof of the claimed splitting capability of \( G_k \) is showing that every state of \( G_k \) can mimic the splitting of some state in \( G_k^k \). We define a function \( f : V(G_k) \rightarrow V(G_k^k) \cup \{ \phi \} \) which maps each state \( u \) of \( G_k \) to a single state of \( G_k^k \) whose splitting determines the splitting of \( u \).

The graphs \( G_k^k \) and \( G \) are irreducible deterministic presentations of the constrained systems \( S_k \) and \( S \), respectively, where \( S_k \subseteq S \). By Proposition 2.2, every state of \( G_k^k \) is dominated by at least one state of \( G \). As a result, every state of \( G_k^k \) is dominated by at least one state of \( G \), for every positive integer \( k \). Given a state \( u \) of \( G_k \), it may happen that no state of \( G_k^k \) is dominated by \( u \). In this case, we define \( f(u) \) to be \( \phi \). Otherwise, there might be several states in \( G_k^k \) which are all dominated by \( u \), and \( f(u) \) is then defined to be one particular state \( Z \) of \( G_k^k \) such that:

\[
(c_k)_Z \geq (c_k)_Y \quad \forall \ Y \in G_k^k: Y \preceq u \tag{6.1}
\]

Using \( \tilde{c}_k \) and \( f \), we define an integer vector \( \tilde{x} = [x_u]_{u \in V(G)} \) as follows:

\[
x_u = \begin{cases} 
(c_k)_{f(u)} & \text{if } f(u) \neq \phi \\
0 & \text{if } f(u) = \phi 
\end{cases}
\tag{6.2}
\]

We now prove that \( \tilde{x} \) is an \((A_{G_k}, n^k)\)-approximate eigenvector. Let \( u \) be a state in \( G_k \) with \( f(u) = Z \neq \phi \). State \( Z \) of \( G_k^k \) is then dominated by state \( u \) of \( G_k \). For every edge \( e \) which goes out of \( Z \) in \( G_k^k \) and carries the label \( \tilde{l} = l_1l_2...l_k \), there exists a corresponding edge \( e' \) in \( G_k \), which goes out of \( u \) and carries the label \( \tilde{l} \). The target state \( Y \) of \( e \) in \( G_k^k \) is dominated by the target state \( v \) of \( e' \) in \( G_k \), and therefore:

\[
x_v = (c_k)_{f(u)} \geq (c_k)_V 
\tag{6.3}
\]

However, there might also exist outgoing edges \( \{e'\} \) of \( u \) that do not correspond to any outgoing edge of \( Z = f(u) \). Consequently, we get:

\[
(A_{G_k} \tilde{x})_u \geq (A_{G_k} \tilde{c}_k)_{f(u)} = n^k \cdot (c_k)_{f(u)} = n^k \cdot x_u 
\tag{6.4}
\]

In case where \( f(u) = \phi \), we get:

\[
(A_{G_k} \tilde{x})_u \geq 0 = n^k x_u 
\tag{6.5}
\]

Thus, \( \tilde{x} \) is indeed an \((A_{G_k}, n^k)\)-approximate eigenvector, for every positive integer \( k \).

It is left to show how the states of \( G_k \) can mimic the splitting of their corresponding states in \( G_k^k \) (the splitting which satisfies the conditions of Lemma 3.5). Let us denote by \( H_k \) the graph which is obtained by the single round of splitting \( G_k^k \), and let \( \tilde{c}_k \) be the resulting eigenvector. The graph which is obtained by splitting \( G_k \) will be denoted by \( H \),
and its corresponding approximate eigenvector will be denoted by $\vec{x}'$. Notice that the states of $G^k$ which are mapped by $f$ to $\phi$ have a zero weight according to $\vec{x}$ and thus should not be split at all. A state $u$ in $G^k$, for which $f(u) \neq \phi$, can mimic the splitting of $f(u)$ in $G^k_{\vec{c}}$ in the following manner:

Suppose that $Z = f(u)$ is split into $Z_1, \ldots, Z_m$, whose respective weights, i.e., respective components in $\vec{c}$, are $w_1, \ldots, w_m$. Then $u$ can be split into $u_1, \ldots, u_m$, whose respective weights, i.e., respective components in $\vec{x}'$, are also $w_1, \ldots, w_m$: The descendant state $u_r$ inherits the outgoing edges which correspond to those inherited by $Z_r$. The rest of the outgoing edges of $u$, that do not correspond to any outgoing edge of $Z$, are spread in an arbitrary way among $u_1, \ldots, u_m$. The sum of weights in the outgoing neighborhood of a descendant state $u_r$ in $H$ cannot be smaller than the sum of weights in the neighborhood of $Z_r$ in $H_{\vec{c}}$. Consequently,

$$ (A_H \vec{x}')_{u_r} \geq (A_{H,\vec{c}} \vec{c})_{Z_r} = n^k(\vec{c})_{Z_r} = n^k u_r = n^k (\vec{x})_{u_r} \quad (6.6) $$

The splitting of $G^k_{\vec{c}}$ is therefore $\vec{x}$-consistent. It is also clear that every state $u$ of $G^k$ splits in this way into no more than $n^k$ states, with weights not greater than $n^{A-k}$, as does the state $Z = f(u)$ in $G^k_{\vec{c}}$.

Observe that all the states of $G^k$ can be split in this way simultaneously in one round. Notice that every (outgoing) neighbor $Y$ of $Z$ is dominated by some neighbor $v$ of $u$, but it is not always true that $Y = f(v)$. As a result, $v$ may be split differently from $Y$, but that should not affect the capability of $u$ to be split in one round like $Z$. It would have affected the splitting capability of $u$ if we had required $u$ to be split like $Z$ in one more round of splitting. (See the discussion in Section 4.1 following Corollary 4.2.)

### 6.2 Application of Theorem 6.1 – Examples

By choosing the Shannon cover $G_S$ of $S$ to be the graph $G$ in Theorem 6.1, we get the following corollary:

**Corollary 6.2** Let $S$ be an irreducible constrained system, let $n$ be a positive integer where $C(S) \geq \log n$, and let $G_S$ be the Shannon cover of $S$. Let $A$ be a positive integer and let $X_A$ be the set of all $(A_{G_S}, n)$-approximate eigenvectors whose components are not greater than $n^A$. Suppose that for some integer $k$, such that $1 \leq k \leq A$, none of the vectors in $X_A$ enables a splitting of $G^k_{\vec{c}}$ while satisfying condition 2 of Theorem 6.1. Then the anticipation of any $(S, n)$ encoder is at least $A + 1$.

Using the notations of Corollary 6.2, we can state the lower bound of [MR91] on the anticipation as follows (see Theorem 2.7): If $X_A = \emptyset$, then the anticipation of any $(S, n)$ encoder is at least $A + 1$. Therefore, Corollary 6.2 yields a lower bound on the anticipation which is at least as strong as the bound of [MR91]. In fact, in the examples we consider in Sections 6.2.1 - 6.2.3, the lower bounds obtained by Corollary 6.2 turn out to be strictly stronger than those obtained from [MR91]. Corollary 6.2 is at least as strong as the lower bound of [AMR95b], because the latter is obtained by checking the splitting capabilities.
only of the graph $G^A_S$; furthermore, that graph needs to be checked with respect to any $(A_{G_S}, n)$-approximate eigenvector — i.e., not just those in $X_A$.

In the following examples, we use Corollary 6.2 in order to derive tight lower bounds on the anticipation of encoders for three well-known systems. Two of these bounds were already proved in Section 4.3 by applying Theorem 4.1, but the proofs in this section are shorter and more elegant.

6.2.1 (2,7)-RLL Constraint

We prove that every encoder for the (2,7)-RLL constraint at rate 1:2 has anticipation at least 3. Rate 1:2 is the common rate for encoding in this case, as the capacity of the (2,7)-RLL constrained system is 0.5174. This result is an improvement of the old lower bound from [MR91] which implies $A \geq 2$. The new lower bound, 3, is achieved by some known encoders, such as the one presented in Figure 2.2. This encoder is due to Franaszek (see [How98]).

Let $S_{2,7}$ be the set of binary words that satisfy the (2,7)-RLL constraint. As we are interested in encoders that operate at rate 1:2, we define $S = S^2_{2,7}$ and $n = 2^1 = 2$. The Shannon cover of $S_{2,7}$ consists of the eight states, $v_0, \ldots, v_7$, whose local neighborhoods are represented by the graph in Figure 6.1. By Franaszek’s algorithm, the maximal component of each vector in $X(A_{G_S}, 2)$ is at least 4, so according to the lower bound of [MR91], the anticipation of any $(S, n)$ encoder satisfies $A \geq \lceil \log_2 4 \rceil = 2$.

In order to prove that the anticipation of every $(S, 2)$ encoder is at least 3, we assume to the contrary the existence of an $(S, 2)$ encoder with anticipation $A = 2$ and we take $k = 1$. We now show that none of the $(A_{G_S}, 2)$-approximate eigenvectors, with maximal component $n^A = 4$, enables the states of $G^k = G^I_S = G_S$ to be split simultaneously so that:

- the components of the resulting eigenvector are not greater than $n^{A-k} = 2^{2-1} = 2$,
- each of the states is split into no more than $n^k = 2^1 = 2$ states.
The \((A_{G_S}, 2)\)-approximate eigenvector which is the largest (componentwise) among those with maximal component 4, is \(\tilde{y} = (2, 3, 4, 3, 3, 2, 1)\). Every \((A_{G_S}, 2)\)-approximate eigenvector \(\tilde{x} = [x_v]_{v \in V(G_S)}\) with maximal component 4 thus satisfies \((x)_{v_2} = 4\), or \((x)_{v_3} = 4\), or both. The local neighborhoods of states \(v_2\) and \(v_3\) are shown in Figure 6.2, where the weights inside the circles are the values of the respective components in the approximate eigenvector \(\tilde{y}\).

For both states, \(v_2\) and \(v_3\), the sum of weights in the neighborhood, according to the vector \(\tilde{y}\), is equal to \(n\) times the weight of the state. Now let \(\tilde{x}\) be some \((A_{G_S}, 2)\)-approximate eigenvector with maximal component 4. We distinguish between two cases:

**Case 1:** \(x_{v_2} = 4\). Since \(x_{v_2} = y_{v_2} = 4\), the sum of weights in the neighborhood of \(v_2\) according to \(\tilde{x}\) cannot be smaller than that sum according to \(\tilde{y}\). On the other hand, we have \(x_u \leq y_u\) for every neighbor \(u\) of \(v_2\) and, consequently, \(x_u = y_u\) for every such neighbor \(u\). Thus, the neighborhood of \(v_2\) according to \(\tilde{x}\) looks exactly like its neighborhood according to \(\tilde{y}\).

Suppose we want to split \(v_2\), consistently with \(\tilde{x}\), into no more than two states, each of weight not greater than 2. The only way to do so is to split \(v_2\), whose weight is 4, into two descendant states, each of weight 2. As \(v_2\) has only three outgoing edges, one of the two descendant states must inherit a single edge of \(v_2\). Now, the target states of the edges going out of \(v_2\) all have weights smaller than 4. Therefore, the size of the neighborhood of the descendant state which inherits a single edge is smaller than \(n\) times its weight \((= 2 \cdot 2)\).

In other words, any splitting of \(G_S\), according to the requirements of Theorem 6.1 for \(k = 1\), is not \(\tilde{x}\)-consistent whenever \(x_{v_2} = 4\).

**Case 2:** \(x_{v_2} \neq 4\) and so \(x_{v_3} = 4\). The neighborhood of \(v_3\), including the weights of the states according to \(\tilde{y}\), looks exactly like the one of \(v_2\), except that state \(v_5\) replaces \(v_4\). Therefore, we can use the same arguments, as in Case 1, in order to conclude that no \(\tilde{x}\)-consistent splitting of \(G_S\) satisfies the requirements of Theorem 6.1 for \(k = 1\) in this case as well.
Hence, by Corollary 6.2, there is no \((S^2_{(2,7),2})\) encoder with anticipation smaller than 3.

### 6.2.2 \((1,7)\)-RLL Constraint

In Example 4.3.2, we proved that every encoder for the \((1,7)\)-RLL system at rate 2:3 has anticipation at least 2. We showed there that for every presentation \(G\) of \(S^3_{(1,7)}\), which is irreducible and deterministic, if there is an \((A_G,4)\)-approximate eigenvector with maximal component not greater than 4, then it is impossible to fully split \(G\) in one round consistently with this vector. The basis for the proof was the incapability of \(G_S\) to be split in such a way.

It was shown that there are only two \((A_{G_S},4)\)-approximate eigenvectors with components not greater than 4, and none of them allows the states of \(G_S\) to be split in one round so that the resulting weights are not greater than 1. The fact that the Shannon cover of \(S\) does not satisfy the requirements of Theorem 6.1 for the parameters \(n = 4\), \(A = 1\), and \(k = 1\), is sufficient in order to conclude, by Corollary 6.2, that every encoder for the \((1,7)\)-RLL system, at rate 2:3, has anticipation greater than 1.

### 6.2.3 \((2,18,2)\)-RLL Constraint

In this example, we re-iterate the result that any encoder at rate 2:5 for the system of the \((2,18,2)\)-RLL constraint has anticipation at least 3. However, we show this here by an application of Corollary 6.2. Let us denote by \(S_{(2,18,2)}\) the \((2,18,2)\)-RLL constrained system. Since the rate is 2:5, we define \(S = S^5_{(2,18,2)}\) and \(n = 2^5 = 4\). The Shannon cover \(G_S\) consists of nineteen states whose local neighborhood (pictures) are represented by the graph in Figure 4.2. By Franaszek's algorithm, the minimal value of a maximal component in an \((A_{G_S},4)\)-approximate eigenvectors is 12. Applying the lower bound of [MR91], we get that \(A \geq \lceil \log_4 12 \rceil = 2\).

For \(k = 1\), the states of the graph \(G_k^2 = G_S\) can be split simultaneously, each into no more than four states, yielding a vector with components not greater than \(4 = n^{A-k}\). This was shown in the previous proof for the \((2,18,2)\)-RLL constraint, in Section 4.3. Therefore, we take \(k = A = 2\). Calculating the values that are needed for the proof, we get: \(n^k = 4^2 = 16\), \(n^{A-k} = 4^{2-2} = 1\), and \(G^k = G_S^2\).

Using Figure 4.2, we obtain that the maximum outdegree of a state in \(G_S^2\) is 21. Let \(\vec{z}\) be any \((A_{G_S},4)\)-approximate eigenvectors with components not greater than 16. We distinguish between two cases.

**Case 1:** \(12 \leq \max_{v \in V(G)} x_v < 16\)

There exist several states in \((G_S)^2\) with weights at least 12 and outdegrees at most 21. Full splitting of each such state means distributing at most 21 edges among at least 12 states, each of weight 1, so there must be at least three \((2 \cdot 12 - 21)\) descendant states, each inheriting only a single edge. As the weight at the target state of the edge is smaller than 16, the size of the neighborhood of each such a descendant state is smaller than \(n^k\) times its own weight. The splitting is thus inconsistent with the vector \(\vec{z}\).
**Case 2:** \( \max_{u \in V(G)} x_u = 16 \).

Recall that the largest \((A_{G,4}, 4)\)-approximate eigenvector among those with maximal component 16 is

\[
\tilde{y} = (9, 12, 16, 12, 16, 12, 16, 11, 15, 11, 15, 11, 14, 10, 13, 8, 11, 5, 7)
\]

It turns out that none of the states of \((G_S)^2\) has more than six neighbors of weight 16, according to \(\tilde{y}\), and therefore according to any other approximate eigenvector \(\tilde{x}\) with largest component 16. When a state of weight 16 and outdegree 21 is split into 16 states, at least 11 (= 2 \cdot 16 - 21) descendant states inherit only a single edge. No more than six out of these eleven states inherit an edge whose target state is of weight 16. Therefore, there are at least five descendant states for which the sum of weights in their neighborhood is smaller than \(n^k\) times their own weight. The splitting is therefore inconsistent with \(\tilde{x}\) also in this case.

We conclude that the Shannon cover of \(S = S^5_{2,18,2}\) does not satisfy the requirements of Theorem 6.1 for \(A = 2\) and \(k = 2\) and hence, an \((S, 4)\) encoder with anticipation 2 does not exist.
Chapter 7

Conclusion

The contribution of this work to the area of encoding for input-constrained channels concentrates in the following two main aspects. First, in providing lower bounds on the anticipation of encoders for input-constrained channels, in the general case and in three particular cases of practical value. Second, in presenting new properties of the state-splitting algorithm, which is the most common algorithm for constructing encoders for input-constrained channels.

Theorem 4.1 provides a necessary and sufficient condition for the existence of \((S, n)\) encoders with anticipation \(A\), for given \(S\), \(n\), and \(A\). The necessary condition is used in Chapter 4 to derive a lower bound on the anticipation of \((S, n)\) encoders. The condition may also be considered as a constructive criterion for verifying the existence of \((S, n)\) encoders with anticipation \(A\): In order to check whether such encoders exist, a finite number of deterministic presentations of \(S\) should be considered. The upper bounds given in Chapter 5 imply that the number of states in each of the graphs that should be checked is bounded from above by a number that can be effectively computed from the parameters of the constraint.

For each of the relevant graph presentations, a finite number of respective approximate eigenvectors have to be considered, as the components of the vectors should not be greater than \(n^A\). Given an irreducible deterministic graph presentation \(G\) of \(S\) and an \((A_G, n)\)-approximate eigenvector, it should be checked whether it is possible to fully split \(G\) in \(A\) rounds consistently with \(\bar{r}\). Not all the splitting options of \(G\) should be checked, but only those in which every state is split in each round into no more than \(n\) states and the components of the resulting approximate eigenvectors after the \(i\)th round are not greater than \(n^{A-i}\). If the check leads to the conclusion that an \((S, n)\) encoder with anticipation \(A\) exists, then it also provides such an \((S, n)\) encoder.

Yet, the check procedure described above may be impractical in many cases. The upper bounds on the number of states in the graphs \(G'\) and \(G''\) from Chapter 5 are very poor. Those bounds are doubly-exponential in the anticipation \(A\) and become enormous numbers for relatively small values of \(n, A\), and number of states in the Shannon cover of \(S\). In addition, there are still many decisions that should be made while applying the state-splitting algorithm, concerning the initial approximate eigenvector and the splittings
that should be executed. Finding a compact and efficient procedure for generating encoders using the state-splitting algorithm requires further research.

Theorem 6.1 gives a necessary condition on the existence of \((S, n)\) encoders with anticipation \(A\) (for given \(S\), \(n\), and \(A\)). Contrary to the necessary condition of Theorem 4.1, the splitting properties given in Theorem 6.1 should be satisfied by every deterministic presentation of \(S\). Therefore, proving lower bounds on the anticipation of \((S, n)\) encoders becomes easier using Theorem 6.1: It is enough to indicate a single deterministic presentation of \(S\) that does not satisfy the requirements of Theorem 4.1 with respect to a given \(A\) in order to conclude that encoders with anticipation \(A\) or less do not exist.

It is clear that the necessary condition of Theorem 4.1 implies the necessary condition of Theorem 6.1: The former condition is also a sufficient condition for the existence of \((S, n)\) encoders with anticipation \(A\) which, in turn, implies the necessary condition of Theorem 6.1. It is still an open problem though whether the condition of Theorem 6.1 implies the necessary condition of Theorem 4.1. The proof of the validity or invalidity of the last implication may be connected to another open problem: Suppose that for every irreducible deterministic presentation \(G\) of \(S\), each of the power graphs \(G^1, \ldots, G^t\) can be split in one round according to the same approximate eigenvector \(\bar{x}\). What are the conditions, if there exist any, which indicate that one of those presentations \(G\) can be split in \(t\) rounds according to the vector \(\bar{x}\)?
References


