Guessing Individual Sequences: Generating Randomized Guesses Using Finite–State Machines

Neri Merhav

The Andrew & Erna Viterbi Faculty of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000, ISRAEL
E–mail: merhav@ee.technion.ac.il

Abstract

Motivated by earlier results on universal randomized guessing, we consider an individual–sequence approach to the guessing problem: in this setting, the goal is to guess a secret, individual (deterministic) vector $x^n = (x_1, \ldots, x_n)$, by using a finite–state machine that sequentially generates randomized guesses from a stream of purely random bits. We define the finite–state guessing exponent as the asymptotic normalized logarithm of the minimum achievable moment of the number of randomized guesses, generated by any finite–state machine, until x^n is guessed successfully. We show that the finite–state guessing exponent of any sequence is intimately related to its finite–state compressibility (due to Lempel and Ziv), and it is asymptotically achieved by the decoder of (a certain modified version of) the 1978 Lempel–Ziv data compression algorithm (a.k.a. the LZ78 algorithm), fed by purely random bits. The results are also extended to the case where the guessing machine has access to a side information sequence, $y^n = (y_1, \ldots, y_n)$, which is also an individual sequence.

Index Terms: guessing exponent, individual sequences, sequence complexity, finite–state machine, Lempel–Ziv algorithm, incremental parsing, side information, randomized guessing.

1 Introduction

Consider the problem of guessing the realization of a finite–alphabet random vector $X^n = (X_1, \ldots, X_n)$ using a series of yes/no questions of the form: "Is $X^n = x^n(1)$?", "Is $X^n = x^n(2)$?", and so on, until a positive response is received. Given a distribution on X^n, a commonly used performance metric for the guessing problem is the expected number of guessing trials required until X^n is guessed correctly, or more generally, a general moment of this number.

The design of *guessing strategies* with the quest of minimizing the moments of the number of guesses has several applications in information theory and related fields. One of them, for example, is sequential decoding, as shown by Arikan [1], who built on the pioneering earlier work of Massey [6] and related the best achievable guessing moment to the Rényi entropy. More modern applications of the guessing problem evolve around aspects of information security, in particular, guessing passwords or decrypting messages protected by random keys. For example, one may submit a sequence of guessing queries in attempt to crack passwords – see, e.g., [10, Introduction] (as well as [11] and other references therein) for a brief, yet quite comprehensive review on guessing and security, as well as for some general historical overview of prior work on the guessing problem in its large plethora of variants and extensions.

One of the main results in [11] is about devising optimal *randomized guessing* strategies, where instead of designing a particular, deterministic guessing list in advance, one randomly draws independent guesses according to a carefully chosen probability distribution of n–vectors, with the motivation that such a randomized strategy saves the need of storing in memory long guessing lists and it also saves the need for synchronization between the guesses of two or more agents who attempt to crack the same password from different IP addresses in parallel. It turns out that with a clever choice of the randomized guessing distribution, the resulting moment of the number of guesses (w.r.t. the randomness of both X^n and the guesses themselves) is exponentially the same as that of the optimal deterministic guessing strategy. In a later work [10], this finding was further strengthened in two different ways: first, the framework was extended from that of a discrete memoryless source (that governs X^n) to the much more general non–unifilar, finite–state source (hidden Markov source model). Secondly, a universal randomized guessing distribution was proposed, which is independent of the unknown parameters of this finite–state source, as well as the moment order

of the number of guesses. The universal random guessing distribution, $P(x^n)$, proposed in [10], was proportional to $2^{-LZ(x^n)}$, where $LZ(x^n)$ designates the length (in bits) of the compressed version of x^n according to the LZ78 data compression algorithm [14]. Moreover, practical algorithms for efficiently implementing this distribution were proposed in [10]. Finally, these results were also extended to account for the availability of side information, Y^n, that is correlated to X^n, under a probabilistic model where the sequence of pairs $\{(X_i, Y_i)\}$ emerges from a non–unifilar finite–state source.

Motivated by those results of [10], in this work, we make an additional step towards generality, by completely dropping the probabilistic assumption concerning the n–vector to be guessed. In particular, we assume that it is an individual (deterministic), finite–alphabet vector, which will be denoted by x^n. We also assume that the independent random guesses are generated sequentially, using a finite–state machine, fed by a sequence of purely random bits.[1] Inspired by the individual–sequence approach to data compression, pioneered by Ziv and Lempel, [14], and followed later in the context of other tasks, like gambling [3], prediction [4], and encryption [8], we define the *finite–state guessing exponent* as the minimum asymptotic exponential rate of the expectation of a given power of the number of guesses, that is achievable by any finite–state machine, and propose a universal randomized guessing scheme that asymptotically achieves the finite–state guessing exponent. Similarly as in [3] and [8] (but in contrast to [4]), we show that the finite–state guessing exponent is very intimately related to the finite–state compressibility of the sequence to be guessed.

The proposed achievability scheme is basically the same as in [10], which was based on the simple idea of feeding the LZ78 decoder by purely random bits. While such a scheme cannot be realized using a finite–state machine, a simple twist can be offered, similary as done in [14] in the context of compression: by employing this randomized guessing scheme repeatedly and resetting its memory at the beginning of every new block (however long), it becomes implementable as a finite–state machine, and it still achieves the finite–state guessing exponent in the limit of an increasing number of states. We therefore prove that the same achievability scheme as in [10] is asymptotically optimal, not only in the probabilistic setting, but also in the individual–sequence setting that we define here. Finally, we outline how these findings extend (with a few twists) to the case where the guessing machine has access to a (deterministic) side information sequence.

[1] The supply of random bits is assumed unlimited.

The outline of the paper is as follows. In Section 2, we formalize the problem setting and spell out the objectives. In Section 3, we assert the converse theorem and prove it. In Section 4, we present the achievability scheme and prove the direct theorem. Finally, in Section 5, we outline the main modifications needed in order to extend the model and the results to the case where side information is available to the guessing machine.

2 Problem Setting

A finite–state guessing machine (FSGM) is defined by a sixtuplet $Q = (\mathcal{U}, \mathcal{X}, \mathcal{Z}, \Delta, f, g)$, where $\mathcal{U} = \{0, 1\}$ is the binary input alphabet, $\mathcal{X}$ is a finite output alphabet of size α, $\mathcal{Z}$ is a finite set of states, $\Delta : \mathcal{Z} \to \{0, 1, 2, \ldots\}$ defines the number of input bits processed at each state, $f : \mathcal{Z} \times \mathcal{U}^* \to \mathcal{X}$ is the output function, and $g : \mathcal{Z} \times \mathcal{U}^* \to \mathcal{Z}$ is the next–state function, where $\mathcal{U}^*$ is a set of variable–length binary strings. When a binary sequence, $\boldsymbol{u} = u_1, u_2, \ldots, u_i \in \mathcal{U}$, $i = 1, 2, \ldots$, drawn from the binary symmetric source (BSS), is fed into an FSGM Q, it produces an output sequence $\hat{\boldsymbol{x}}^n = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \in \mathcal{X}^n$, while passing through a sequence of states, $z_1, z_2, \ldots, z_n$, according to the following recursive equations, implemented for $i = 1, 2, \ldots, n$,

$$t_i = t_{i-1} + \Delta(z_i), \qquad t_0 \stackrel{\Delta}{=} 0 \tag{1}$$

$$v_i = (u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots, u_{t_i}), \tag{2}$$

$$\hat{x}_i = f(z_i, v_i), \tag{3}$$

$$z_{i+1} = g(z_i, v_i), \tag{4}$$

where, without loss of generality, z_1 is assumed to be some fixed member of $\mathcal{Z}$. An FSGM Q with s states, or an s–state guessing machine, is one with $|\mathcal{Z}| = s$.

In the guessing game between Alice and Bob, Alice has access to a certain secret n–vector $x^n = (x_1, \ldots, x_n) \in \mathcal{X}^n$, while Bob is unaware of x^n, but is equipped with an FSGM Q. In each guessing round, Bob activates Q by feeding it with a sequence of purely random bits, $u_1, u_2, \ldots$, until an output sequence of length n, $\hat{x}^n = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)$, is obtained, which is then submitted to Alice as a guess. Alice in her turn compares $\hat{x}^n$ to x^n and returns an affirmative response if they match, and a negative one if they do not. In the former case, x^n has been guessed successfully and the guessing process terminates. In the latter case, a new guessing round takes place, using new,

independent random bits, and so on, until x^n is guessed successfully.

Let $G_Q(x^n)$ be the random number of guessing rounds needed for Q until success. For a given $\zeta > 0$, define

$$\gamma_s(x^n) = \min_{Q \in \mathcal{Q}(s)} \frac{\ln \boldsymbol{E}\{[G_Q(x^n)]^\zeta\}}{n}, \tag{5}$$

where $\mathcal{Q}(s)$ is the set of all FSGMs with no more than s states. In order to define asymptotics, consider an infinite sequence $\boldsymbol{x} = (x_1, x_2, \ldots)$, whose components take on values in $\mathcal{X}$. We define the asymptotic s–state guessing exponent as

$$\gamma_s(\boldsymbol{x}) = \limsup_{n \to \infty} \gamma_s(x^n), \tag{6}$$

and finally, the finite–state guessing exponent of $\boldsymbol{x}$ is defined as

$$\gamma(\boldsymbol{x}) = \lim_{s \to \infty} \gamma_s(\boldsymbol{x}). \tag{7}$$

While the minimizing FSGM Q^* of eq. (5) depends, in general, on x^n, our objective is to devise a universal, sequential, randomized guessing scheme that is independent of x^n, and yet it asymptotically achieves $\gamma_s(x^n)$ in the limit of large n, followed by the limit of large s, and therefore it achieves also $\gamma(\boldsymbol{x})$. Moreover, this universal guessing scheme will not depend on the moment order ζ either.

Two observations regarding the above defined model of the FSGM are in order.

1. As can be seen in eqs. (1)–(4), at each cycle i, the FSGM processes $\Delta(z_i)$ new input bits, in other words, a number of bits that depends solely on the current state, z_i. We could have defined the model to be seemingly more general, where the number of input bits depends, not only on z_i, but also on a certain number of the next incoming input bits, $u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots$, in the following manner. Consider a situation where for each state $z \in \mathcal{Z}$, one defines a binary tree, $T(z)$, and define $\Delta(z_i, u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots)$ to be the number of branches of the path from the root of $T(z)$ down to the leaf pertaining to the trail associated with $u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots$. On the other hand, if one does not care about "wasting" input bits, this model can be formalized in the above defined framework by re–defining $\Delta(z)$ to be the length of the path from the root to the deepest leaf and then extending $T(z)$ to be a full binary tree of depth $\Delta(z)$, where f and g are defined to be the

same for all descendants of every given leaf of the original tree, $T(z)$. For example, if $T(z)$ originally contains the three leaves, corresponding to the binary paths '0', '10' and '11', then we extend the path '0' to its two children, '00' and '01' to obtain a full binary tree of four leaves and depth $\Delta(z) = 2$, but we let $f(z, 00) = f(z, 01)$ and $g(z, 00) = g(z, 01)$ (see also the example in comment no. 2 below), so that the second bit after '0' is not really used and hence is immaterial. Therefore, the only difference between the FSGM with the extended tree and the original FSGM is that the second bit of '00' and '01' is "wasted", but since we are assuming, in this paper, that resources of randomness are unlimited, this is inconsequential. The reason we prefer the model of $\Delta(z_i)$ over the model of $\Delta(z_i, u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots)$ is just its relative simplicity.

2. The above defined model of the FSGM is general enough to operate as any (state–dependent) mapping from variable–length binary input strings to variable–length strings of symbols, such as decoders corresponding to variable–to–variable length encoders for data compression. This is the case when we allow $\Delta(z) = 0$ at some states, as it enables the finite–state machine to idle between successive readings of chunks of input bits. To implement such a variable–to–variable length mapping in the framework of our model, the system works as follows. Upon receiving a variable–length binary input string v_i, the system passes along a certain sequence of states, where at each state it outputs one output symbol without reading any new input bits ($\Delta(z_j) = 0$), until the entire output word is produced. As a simple example, consider the variable–to–variable length mapping

$$0 \to \text{ab} \quad 10 \to \text{bac} \quad 11 \to \text{ca}. \tag{8}$$

The state transition diagram of the associated FSM is depicted in Fig. 1, as a graph whose vertices represent the states and whose edges designate the state transitions.

Each state transition is defined by a pair (z, v), $z \in \mathcal{Z}$, $v \in \mathcal{U}^{\Delta(z)}$, and its edge is labeled in Fig. 1 by the corresponding output $\hat{x} = f(z, v)$, followed by a slash, and followed in turn by the contents of v. For instance, states and A and B represent the assignment $0 \to$ ab or equivalently, the pair of assignments $00 \to$ ab and $01 \to$ ab (see comment no. 1 above). The transition from state A to state B is labeled by "a/null", which means $\Delta(\text{A}) = 0$ and v is null (no new bits are read by the system), and the output is $\text{a} = f(\text{A}, \text{null})$. This means that after visiting at state A, the machine must pass to state B (as $\text{B} = g(\text{A}, \text{null})$), without processing input bits. From state B, all outgoing

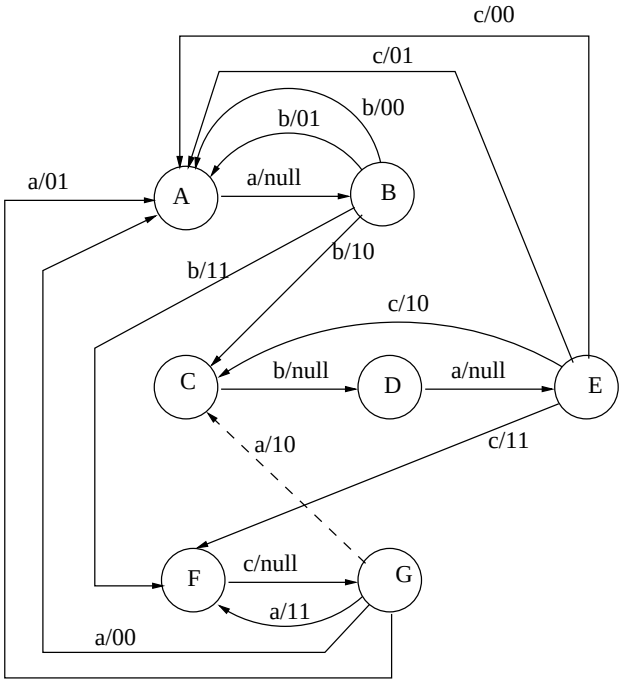

Figure 1: State transition diagram of the variable–to–variable length mapping (8).

edges are associated with the output "b", and $\Delta(\mathrm{B}) = 2$ new bits are read for the next round of the mapping (8): if the first bit is '0', then we are back to state A, otherwise, we pass to state C or state F, to map '10' or '11', respectively, and we proceed similarly as before.

3 Converse Theorem

3.1 Preliminaries

Before presenting the converse theorem and its proof, we need to recall the notions of finite–state compressibility and string parsing from [14] (which inspired the above definition of the FSGM). To make the presentation self–contained, we provide the definitions of these terms here, but with a slightly different notation, not to confuse with the above notation of the FSGM.

A K–state encoder E is defined by a quintuple $(\mathcal{X}, \mathcal{B}, \Sigma, f, g)$, where $\mathcal{X}$ is the alphabet of size α of the source sequence to be compressed, $\mathcal{B}$ is a finite set of binary words (possibly of different lengths, including the null word for idling), Σ is a finite set of K states, $p : \Sigma \times \mathcal{X} \to \mathcal{B}$ is the

encoder output function, and $q : \Sigma \times \mathcal{X} \to \Sigma$ is the next–state function. When an input sequence $(x_1, x_2, ...)$ is fed sequentially into E, the encoder outputs a sequence of binary words $(b_1, b_2, \ldots)$, $b_i \in \mathcal{B}$, while going through a sequence of states $(\sigma_1, \sigma_2, \ldots)$, $\sigma_i \in \Sigma$, according to

$$b_i = p(\sigma_i, x_i), \quad \sigma_{i+1} = q(\sigma_i, x_i), \quad i = 1, 2, ... \tag{9}$$

where σ_i is the state of E at time instant i and where the initial state, σ_1, is assumed a fixed member of Σ. The decoder receives the compressed bit–stream, $b_1, b_2, \ldots$, and reconstructs $x_1, x_2, \ldots$. A finite–state encoder E is said to be *information lossless* (IL) if for all $\sigma_1 \in \Sigma$ and all $x^n \in \mathcal{X}^n$, $n \geq 1$, the triple $(\sigma_1, \sigma_{n+1}, b^n)$ (with $b^n \stackrel{\Delta}{=} (b_1, \ldots, b_n)$) uniquely determines x^n, where σ_{n+1} and b^n are obtained by iterating eq. (9) with σ_1 and x^n as inputs. The length function associated with E is defined as $L_E(b^n) = \sum_{i=1}^{n} l(b_i)$, where $l(b_i)$ is the length of the binary string $b_i \in \mathcal{B}$. The compression ratio of x^n w.r.t. E is defined as[2]

$$\rho_E(x^n) = \frac{L_E(b^n)}{n}. \tag{10}$$

Next, we define

$$\rho_K(x^n) = \min_{E \in \mathcal{E}(K)} \rho_E(x^n), \tag{11}$$

where $\mathcal{E}(K)$ is the class of all IL encoders with no more than K states. Furthermore, for an infinite sequence $\boldsymbol{x} = (x_1, x_2, \ldots)$, let

$$\rho_K(\boldsymbol{x}) = \limsup_{n \to \infty} \rho_K(x^n), \tag{12}$$

and finally,

$$\rho(\boldsymbol{x}) = \lim_{K \to \infty} \rho_K(\boldsymbol{x}). \tag{13}$$

The following converse theorem was asserted and proved in [14] in the context of data compression.

Theorem 1 *[14, Theorem 1] For every $x^n \in \mathcal{X}^n$,*

$$\rho_K(x^n) \geq \frac{[c(x^n) + K^2]}{n} \log \left[\frac{c(x^n) + K^2}{4K^2} \right] + \frac{2K^2}{n}, \tag{14}$$

where $c(x^n)$ is the largest number of distinct strings (or phrases) whose concatenation forms x^n.

[2]Note that here, unlike in [14], we define the compression ratio without normalization by $\log \alpha$, where α is the source alphabet size.

3.2 Main Result

Returning to our guessing problem, our converse theorem relates the finite–state guessing exponent of x^n to its finite–state compressibility.

Theorem 2 *Let Q be an arbitrary FSGM with s states and let x^n be a sequence generated from Q with probability less than $1/2$. Then,*

$$\gamma_s(x^n) \geq \zeta \cdot \max_{\{\ell \ \mathit{divides}\ n\}} \left[\rho_{K(\ell)}(x^n) - \frac{\log(2s^3e)}{\ell}\right] - \frac{\log(e^2 2^\zeta)}{n}; \tag{15}$$

$$\gamma_s(x^n) \geq \zeta\Big[c(x^n)\ln c(x^n) - n\delta_n(s)\Big], \tag{16}$$

where $K(\ell) \stackrel{\Delta}{=} (\alpha^{\ell+1}-1)/(\alpha-1)$ and $\delta_n(s)$ tends to zero uniformly as $n \to \infty$ for any fixed s.

Discussion. The condition that x^n is generated from Q with probability less than $1/2$ is purely technical and it is almost trivially always met. It means that at least one input bit is utilized for the generation of x^n. Theorem 2 provides two different lower bounds on $\gamma_s(x^n)$. The first one relates the s–state guessing exponent with the K–state compressibility of x^n, and it implies also an asymptotic version of our converse theorem, which is $\gamma(\boldsymbol{x}) \geq \zeta \cdot \rho(\boldsymbol{x})$. This lower bound will be shown to be achieved asymptotically by a certain sequence of FSGMs whose number of states goes to infinity *after* $n \to \infty$, thus establishing also the reversed inequality $\gamma(\boldsymbol{x}) \leq \zeta \cdot \rho(\boldsymbol{x})$, and hence an equality,

$$\gamma(\boldsymbol{x}) = \zeta \cdot \rho(\boldsymbol{x}). \tag{17}$$

The second lower bound, which is in fact a corollary of the first one, and hence is weaker, has the advantage that it is computable. It can be achieved by a randomized guessing machine whose number of states is unlimited. The details are deferred to the next section, where the achievability results will be asserted and proved.

Proof of Theorem 2. Let Q be an arbitrary FSGM with s states. First, observe that since the input bits are i.i.d., the output of Q is a non–unifilar[3] finite–state source, a.k.a. a hidden Markov source.

[3]A non–unifilar finite–state source is a finite–state source where the underlying state sequence is hidden, namely, it cannot be recovered from the source sequence alone. A unifilar source is obtained as a special case, where eq. (4) is replaced by the equation $z_{i+1} = g(z_i, \hat{x}_i)$. Note that since $\hat{x}_i = f(z_i, v_i)$, then eventually, here too z_{i+1} is a function of (z_i, v_i), but not every function of (z_i, v_i) can be represented as a composition of f and g in the form, $z_{i+1} = g(z_i, f(z_i, v_i))$. Thus, our model is more general than the model of unifilar finite–state sources.

In particular,

$$P(\hat{x}^n|z_1) = \sum_{z_2,z_3,\ldots,z_{n+1}} \prod_{i=1}^{n} P(\hat{x}_i, z_{i+1}|z_i), \tag{18}$$

where $P(\hat{x}, z'|z) = m(\hat{x}, z'|z) \cdot 2^{-\Delta(z)}$, $m(\hat{x}, z'|z)$ being the number of binary strings $\{v\}$ of length $\Delta(z)$ such that $f(z, v) = \hat{x}$ and $g(z, v) = z'$, as each such binary string has probability $2^{-\Delta(z)}$. Since z_1 is assumed fixed, the conditioning on z_1 will henceforth be dropped. Now, since the guesses are statistically mutually independent,

$$\boldsymbol{E}\{[G_Q(x^n)]^\zeta\} = \sum_{k=1}^{\infty} k^\zeta [1 - P(x^n)]^{k-1} P(x^n). \tag{19}$$

Consider now the following chain of inequalities, holding for any $q \in (0, 1]$:

$$\begin{aligned}
\sum_{k=1}^{\infty} k^\zeta (1-q)^{k-1} \cdot q &\geq q \cdot \sum_{k=\lfloor 1/q \rfloor}^{\infty} k^\zeta (1-q)^{k-1} \\
&\geq q \cdot \sum_{k=\lfloor 1/q \rfloor}^{\infty} \left(\left\lfloor \frac{1}{q} \right\rfloor \right)^\zeta (1-q)^k \\
&\geq q \cdot \left(\frac{1}{q} - 1 \right)^\zeta \cdot \frac{(1-q)^{\lfloor 1/q \rfloor}}{1-(1-q)} \\
&\geq \left(\frac{1}{q} - 1 \right)^\zeta \cdot (1-q)^{1/q} \\
&= \left(\frac{1}{q} - 1 \right)^\zeta \cdot \exp\left\{ \frac{1}{q} \ln(1-q) \right\} \\
&\geq \left(\frac{1-q}{q} \right)^\zeta \cdot \exp\left\{ -\frac{1}{1-q} \right\},
\end{aligned} \tag{20}$$

where in the last step, we have used the inequality $\ln(1+t) \geq \frac{t}{1+t}$, which holds for all $t > -1$. Thus, with the assignment $q = P(x^n)$, and since it is assumed that $P(x^n) \leq 1/2$, we have

$$\begin{aligned}
\boldsymbol{E}\{[G_Q(x^n)]^\zeta\} &\geq \frac{2^{-\zeta}}{e^2} \cdot [P(x^n)]^{-\zeta} \\
&= \frac{2^{-\zeta}}{e^2} \cdot \exp_2\{-\zeta \log P(x^n)\}.
\end{aligned} \tag{21}$$

Assume that ℓ divides n and consider the partition of $\hat{x}^n$ into $m = n/\ell$ non–overlapping blocks of length ℓ, $\hat{x}^n = (\hat{x}_1^\ell, \hat{x}_{\ell+1}^{2\ell}, \ldots, \hat{x}_{n-\ell+1}^n)$, where here and throughout the sequel, the notation $\hat{x}_i^j$ $(i < j)$ denotes the string segment $(\hat{x}_i, \hat{x}_{i+1}, \ldots, \hat{x}_j)$ (and a similiar notation applies also to other vectors). We also define the diluted sequence of states $z^m \stackrel{\Delta}{=} (z_1, z_{\ell+1}, z_{2\ell+1}, \ldots, z_{n-\ell+1})$. Then, it follows from eq. (18) that

$$P(\hat{x}^n, z^m) = \prod_{i=0}^{m-1} P(\hat{x}_{i\ell+1}^{i\ell+\ell}, z_{i\ell+\ell+1}|z_{il+1}). \tag{22}$$

Let $\mathcal{T}(\hat{x}^n|z^m)$ denote the set of vectors in $\mathcal{X}^n$ that are obtained by permuting different ℓ–blocks that begin at the same state, z, and end at the same state, z'. Obviously, all members of $\mathcal{T}(\hat{x}^n|z^m)$ have the same joint probability with z^m. A simple combinatorial argument (see, e.g., [7, eq. (A.11)]) yields that

$$\log|\mathcal{T}(\hat{x}^n|z^m)| \geq m[\hat{H}_\ell(x^n) - \log(s^2 e)], \tag{23}$$

where $\hat{H}_\ell(x^n)$ is the entropy associated with the empirical distribution of the non–overlapping ℓ–blocks of $\hat{x}^n$, that is,

$$\hat{H}_\ell(\hat{x}^n) = -\sum_{a^\ell \in \mathcal{X}^\ell} \hat{P}(a^\ell) \log \hat{P}(a^\ell), \tag{24}$$

where

$$\hat{P}(a^\ell) = \frac{1}{m}\sum_{i=0}^{m-1} \mathcal{I}\{\hat{x}_{i\ell+1}^{i\ell+\ell} = a^\ell\}, \quad a^\ell \in \mathcal{X}^\ell \tag{25}$$

$\mathcal{I}\{\hat{x}_{i\ell+1}^{i\ell+\ell} = a^\ell\}$ being the indicator function for the event $\{\hat{x}_{i\ell+1}^{i\ell+\ell} = a^\ell\}$. Now, since

$$1 \geq \sum_{\tilde{x}^n \in \mathcal{T}(\hat{x}^n|z^m)} P(\tilde{x}^n, z^m) = |\mathcal{T}(\hat{x}^n|z^m)| \cdot P(\hat{x}^n, z^m), \tag{26}$$

it follows that

$$\begin{aligned} P(\hat{x}^n) &= \sum_{z^m} P(\hat{x}^n, z^m) \\ &\leq \sum_{z^m} \frac{1}{|\mathcal{T}(\hat{x}^n|z^m)|} \\ &\leq s^m \cdot 2^{-m[\hat{H}_\ell(x^n) - \log(s^2 e)]} \\ &= 2^{-m[\hat{H}_\ell(x^n) - \log(s^3 e)]}, \end{aligned} \tag{27}$$

or, equivalently,

$$-\log P(\hat{x}^n) \geq m[\hat{H}_\ell(x^n) - \log(s^3 e)]. \tag{28}$$

It remains to lower bound the r.h.s. in terms of the finite–state compressibility. Consider a Shannon code w.r.t. an arbitrary probability distribution F of ℓ–vectors, that is, a code that assigns $\lceil -\log F(x^\ell) \rceil$ bits to the lossless compression of x^ℓ. Similarly as argued in [9, p. 2245, right column], such a code can be implemented by an IL finite–state encoder with $\sum_{j=0}^{\ell} \alpha^j = (\alpha^{\ell+1} - 1)/(\alpha - 1) =$

$K(\ell)$ states, and so, by the definition of the K–state compressibility,

$$\begin{aligned} n\rho_{K(\ell)}(\hat{x}^n) &\leq \sum_{i=0}^{m-1} \lceil -\log F(\hat{x}_{i\ell+1}^{i\ell+\ell}) \rceil \\ &\leq -\sum_{i=0}^{m-1} \log F(\hat{x}_{i\ell+1}^{i\ell+\ell}) + m, \end{aligned} \tag{29}$$

and since this is true for every probability distribution F on $\mathcal{X}^\ell$, the r.h.s. may be minimized w.r.t. F, to obtain

$$n\rho_{K(\ell)}(\hat{x}^n) \leq m\hat{H}_\ell(\hat{x}^n) + m. \tag{30}$$

Combining this with eq. (28), we obtain

$$-\log P(\hat{x}^n) \geq n\rho_{K(\ell)}(\hat{x}^n) - m\log(2s^3e), \tag{31}$$

which, together with eq. (21), proves the first inequality of the converse theorem, since ℓ is an arbitrary divisor of n.

Finally, the second lower bound of the converse theorem is obtained from [14, Theorem 1], as follows.

$$\begin{aligned} n\rho_{K(\ell)}(\hat{x}^n) &\geq [c(\hat{x}^n) + K^2(\ell)] \log \frac{c(\hat{x}^n) + K^2(\ell)}{4K^2(\ell)} \\ &\geq c(\hat{x}^n)\log c(\hat{x}^n) - [c(\hat{x}^n) + K^2(\ell)]\log[4K^2(\ell)] \\ &\geq c(\hat{x}^n)\log c(\hat{x}^n) - \frac{n\log[4K^2(\ell)]\log\alpha}{(1-\epsilon_n)\log n} - K^2(\ell)\log[4K^2(\ell)], \end{aligned} \tag{32}$$

where $\epsilon_n \to 0$, and the last step follows from the inequality $c(x^n) \leq n\log\alpha/[(1-\epsilon_n)\log n]$ [5, Theorem 2], [2, Lemma 13.5.3, p. 450]. It follows that the second lower bound of Theorem 2 holds with

$$\delta_n(s) = \min_{\{\ell \text{ divides } n\}} \left[\frac{\log[4K^2(\ell)]\log\alpha}{(1-\epsilon_n)\log n} + \frac{K^2(\ell)\log[4K^2(\ell)]}{n} + \frac{\log(2s^3e)}{\ell} \right] + \frac{\log(e^2 2^\zeta)}{n}. \tag{33}$$

This completes the proof of Theorem 2.

4 Direct Theorem

4.1 Preliminaries

Before presenting our direct theorem (achievability), we need to recall a few more terms and facts from [14].

The incremental parsing procedure of the LZ78 algorithm is a procedure of sequentially parsing a vector x^n such that each new phrase is the shortest string that has not been encountered before as a parsed phrase, with the possible exception of the last phrase, which might be incomplete. For example, the incremental parsing of the vector $x^{15} =$ abbabaabbaaabaa is a,b,ba,baa,bb,aa,ab,aa. Let $c_{\rm LZ}(x^n)$ denote the number of phrases in x^n resulting from the incremental parsing procedure. Obviously, $c_{\rm LZ}(x^n) \le c(x^n) + 1$ [5, Theorem 2], [14, eq. (6)], as $c(x^n)$ was defined above as the maximum number of distinct phrases. Let $LZ(x^n)$ denote the length of the LZ78 binary compressed code for x^n. According to [14, Theorem 2],

$$\begin{aligned}
LZ(x^n) &\le [c(x^n)+1]\log\{2\alpha[c(x^n)+1]\} \\
&= c(x^n)\log[c(x^n)+1] + c(x^n)\log(2\alpha) + \log\{2\alpha[c(x^n)+1]\} \\
&= c(x^n)\log c(x^n) + c(x^n)\log\left[1+\frac{1}{c(x^n)}\right] + c(x^n)\log(2\alpha) + \log\{2\alpha[c(x^n)+1]\} \\
&\le c(x^n)\log c(x^n) + \log e + \frac{n(\log\alpha)\log(2\alpha)}{(1-\epsilon_n)\log n} + \log[2\alpha(n+1)] \\
&\stackrel{\Delta}{=} c(x^n)\log c(x^n) + n\cdot\epsilon(n), \qquad (34)
\end{aligned}$$

where $\epsilon(n)$ clearly tends to zero as $n\to\infty$, at the rate of $1/\log n$.

4.2 Main Result

Returning to the guessing problem, our direct theorem is as follows.

Theorem 3 *(a) Consider the random guessing distribution*

$$P(\hat{x}^n) = \frac{2^{-LZ(\hat{x}^n)}}{\sum_{x^n\in\mathcal{X}^n} 2^{-LZ(x^n)}}. \qquad (35)$$

Then,

$$\log \boldsymbol{E}\{[G(x^n)]^\zeta\} \le \zeta\cdot c(x^n)\log c(x^n) + n\cdot O\left(\frac{1}{\log n}\right). \qquad (36)$$

(b) Let ℓ divide n and consider the product form random guessing distribution,

$$P(\hat{x}^n) = \prod_{i=0}^{n/\ell-1}\left[\frac{2^{-LZ(\hat{x}_{i\ell+1}^{i\ell+l})}}{\sum_{x^\ell\in\mathcal{X}^\ell} 2^{-LZ(x^\ell)}}\right]. \qquad (37)$$

Then, for every positive integer K,

$$\log \boldsymbol{E}\{[G(x^n)]^\zeta\} \le \zeta n\cdot\left[\rho_K(\hat{x}^n) + \frac{\log(4K^2)}{(1-\epsilon_\ell)\log\ell} + \frac{K^2\log(4K^2)}{\ell} + \epsilon(\ell)\right]. \qquad (38)$$

Discussion. Part (a) of Theorem 3 is an achievability result that is matching the second lower bound in Theorem 2. However, this is incompatible with the framework of finite–stage machines since this random guessing distribution cannot be implemented with a finite–state machine as n grows without bound. The random guessing distribution of part(b), on the other hand, can be implemented using an FSGM with no more than $\ell \cdot \alpha^{\ell}$ states, and it is a matching achievability result to the first lower bound of Theorem 2. It should be pointed that even the random guessing distribution of part (a) can be implemented efficiently using practical algorithms, as described in [10]. The upper bound of part (b) is meaningful when $K^2 \ll \ell$.

Theorems 2 and 3 together tell us that essentially the best achievable guessing moment $\boldsymbol{E}\{[G(x^n)]^{\eta}\}$ is of the exponential order of $2^{\zeta c(x^n)\log c(x^n)}$. In general, when the η-th moment of a random variable behaves like A^{η} for some positive constant A that is independent of η, and every $\eta > 0$, it indicates that this random variable is (nearly) degenerate. In other words, it concentrates very rapidly around its mean. Indeed, it can easily be shown very similarly[4] as in [10, eq. (17)], that the probability of the event $\{G(x^n) \geq 2^{c(x^n)\log c(x^n)+n\epsilon}\}$ decays double exponentially rapidly for the optimal guessing distribution, provided that $\epsilon > 0$.

Proof of Theorem 3. Part (a) is almost a restatement of [10, Theorem 3]. The proof is therefore almost identical to the proof of that result. The only difference is that here, since x^n is a given deterministic vector, the final step in [10, proof of Theorem 3], of taking the expectation w.r.t. the randomness of x^n, is now omitted, and the expectation of $[G_Q(x^n)]^{\zeta}$ is taken only w.r.t. the randomness of the guesses, which as is shown in [10, Lemma 1], behaves like $[P(x^n)]^{-\zeta}$, where $P(\cdot)$ is the random guessing distribution.

As for part(b) of Theorem 3, since $LZ(\cdot)$ is a uniquely decipherable code, it satisfies Kraft's

[4]In eq. (17) of [10], this it is shown for the empirical entropy (instead of $c(x^n)\log c(x^n)$, but the proof for $c(x^n)\log c(x^n)$ is exactly the same.

inequality, and so,

$$\begin{aligned} P(\hat{x}^n) &= \prod_{i=0}^{n/\ell-1} \frac{2^{-LZ(\hat{x}_{\ell i+1}^{\ell i+\ell})}}{\sum_{\tilde{x}^\ell \in \mathcal{X}^\ell} 2^{-LZ(\tilde{x}^\ell)}} \\ &\geq \prod_{i=0}^{n/\ell-1} 2^{-LZ(\hat{x}_{\ell i+1}^{\ell i+\ell})} \\ &= \exp_2\left\{ -\sum_{i=0}^{n/\ell-1} LZ(\hat{x}_{\ell i+1}^{\ell i+\ell}) \right\} \\ &\geq \exp_2\left\{ -\sum_{i=0}^{n/\ell-1} c(\hat{x}_{\ell i+1}^{\ell i+\ell}) \log c(\hat{x}_{\ell i+1}^{\ell i+\ell}) - n\cdot\epsilon(\ell) \right\}, \end{aligned} \tag{39}$$

where the last step follows from eq. (34) applied to ℓ–vectors. It remains to show that $\sum_{i=0}^{n/\ell-1} c(\hat{x}_{\ell i+1}^{\ell i+\ell}) \log c(\hat{x}_{\ell i+1}^{\ell i+\ell})$ is essentially no larger than $\rho_K(\hat{x}^n)$ for some K that can be chosen arbitrarily large, provided that $n \gg \ell$ and ℓ is large enough. Consider the following chain of inequalities for a given positive integer K:

$$\begin{aligned} & \sum_{i=0}^{n/\ell-1} \{c(\hat{x}_{\ell i+1}^{\ell i+\ell}) \log c(\hat{x}_{\ell i+1}^{\ell i+\ell}) - [c(\hat{x}_{\ell i+1}^{\ell i+\ell}) + K^2] \log(4K^2)\} \\ \leq\ & \ell \sum_{i=0}^{n/\ell-1} \rho_K(\hat{x}_{\ell i+1}^{\ell i+\ell}) \\ =\ & \ell \cdot \sum_{i=0}^{n/\ell-1} \min_{E\in\mathcal{E}(K)} \rho_E(\hat{x}_{\ell i+1}^{\ell i+\ell}) \\ =\ & \sum_{i=0}^{n/\ell-1} \min_{E\in\mathcal{E}(K)} \sum_{j=1}^{\ell} l[p(\sigma_{\ell i+j}, \hat{x}_{\ell i+j})] \\ \leq\ & \min_{E\in\mathcal{E}(K)} \sum_{i=0}^{n/\ell-1} \sum_{j=1}^{\ell} l[p(\sigma_{\ell i+j}, \hat{x}_{\ell i+j})] \\ =\ & \min_{E\in\mathcal{E}(K)} \sum_{i=1}^{n} l[p(\sigma_i, \hat{x}_i)] \\ =\ & n\cdot\rho_K(\hat{x}^n), \end{aligned} \tag{40}$$

where the first inequality follows from [14, Theorem 1] applied to ℓ–vectors. Thus,

$$\begin{aligned}\sum_{i=0}^{n/\ell-1} c(\hat{x}_{\ell i+1}^{\ell i+\ell}) \log c(\hat{x}_{\ell i+1}^{\ell i+\ell}) &\leq n \cdot \rho_K(\hat{x}^n) + \sum_{i=0}^{n/\ell-1} [c(\hat{x}_{\ell i+1}^{\ell i+\ell}) + K^2] \log(4K^2) \\ &\leq n \cdot \rho_K(\hat{x}^n) + \frac{n}{\ell} \cdot \left[\frac{\ell \log(4K^2)}{(1-\epsilon_\ell) \log \ell} + K^2 \log(4K^2) \right] \\ &= n \cdot \left[\rho_K(\hat{x}^n) + \frac{\log(4K^2)}{(1-\epsilon_\ell) \log \ell} + \frac{K^2 \log(4K^2)}{\ell} \right], \end{aligned} \tag{41}$$

and so,

$$\begin{aligned}\sum_{i=0}^{n/\ell-1} LZ(\hat{x}_{\ell i+1}^{\ell i+\ell}) &\leq \sum_{i=0}^{n/\ell-1} c(\hat{x}_{\ell i+1}^{\ell i+\ell}) \log c(\hat{x}_{\ell i+1}^{\ell i+\ell}) + n\epsilon(\ell) \\ &\leq n \cdot \left[\rho_K(\hat{x}^n) + \frac{\log(4K^2)}{(1-\epsilon_\ell) \log \ell} + \frac{K^2 \log(4K^2)}{\ell} + \epsilon(\ell) \right]. \end{aligned} \tag{42}$$

This completes the proof of Theorem 3.

5 Side Information

We now consider the extended setting where a deterministic side information vector, y^n, is available to the randomized guessing machine. Since most of the ideas and techniques extend quite straightforwardly, we only outline the differences compared to the case without side information.

We now define the model as follows. An FSGM is defined by a set $Q = (\mathcal{U}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}, \ell, f, g, \Delta)$, where $\mathcal{U}$, $\mathcal{X}$, and $\mathcal{Z}$ are as before, $\mathcal{Y}$ is the finite alphabet of size β associated with the side information, ℓ is a positive integer, $f : \mathcal{Z} \times \mathcal{Y}^\ell \times \mathcal{U}^* \to \mathcal{X}^\ell$ is the output function, $g : \mathcal{Z} \times \mathcal{Y}^\ell \times \mathcal{U}^* \to \mathcal{Z}$ is the next–state function, and $\Delta : \mathcal{Z} \times \mathcal{Y}^\ell \to \{0, 1, 2, \ldots\}$. When $\boldsymbol{u} = u_1, u_2, \ldots$ and the side information sequence, $\boldsymbol{y} = y_1, y_2, \ldots$, $y_t \in \mathcal{Y}$, $t = 1, 2, \ldots$, are fed into Q, it produces $\hat{\boldsymbol{x}}^n$, according to

$$t_i = t_{i-1} + \Delta(z_i, y_{(i-1)\ell+1}^{i\ell}), \qquad t_0 \stackrel{\Delta}{=} 0 \tag{43}$$

$$v_i = (u_{t_{i-1}+1}, u_{t_{i-1}+2}, \ldots, u_{t_i}), \tag{44}$$

$$\hat{x}_{(i-1)\ell+1}^{i\ell} = f(z_i, y_{(i-1)\ell+1}^{i\ell}, v_i), \tag{45}$$

$$z_{i+1} = g(z_i, y_{(i-1)\ell+1}^{i\ell}, v_i). \tag{46}$$

Note that here, we have somewhat generalized the model in the sense that the system is now fed by ℓ–tuples of $\boldsymbol{y}$ and it produces ℓ–tuples of $\hat{\boldsymbol{x}}$. The reason is that in the context of systems with

a side information input, input–output mechanisms that work on a symbol–by–symbol basis (i.e., $\ell = 1$) are too limited. It is reasonable to allow dependencies between side information symbols and their corresponding output symbols with some delay and anticipation, and indeed, such a delay and anticipation will be needed in the achievability scheme. We could have allowed a general ℓ also in the earlier case, where no side information was available.

Let $G_Q(x^n|y^n)$ denote the random number of guessing rounds needed for Q until success. Next, for a given $\zeta > 0$, define

$$\gamma_{s,\ell}(x^n|y^n) = \min_{Q\in\mathcal{Q}(s,\ell)} \frac{\ln \boldsymbol{E}\{[G_Q(x^n|y^n)]^\zeta\}}{n}, \tag{47}$$

where $\mathcal{Q}(s,\ell)$ is the set of all FSGMs with block length less than or equal to ℓ and no more than s states. For two given infinite sequences, $\boldsymbol{x} = (x_1, x_2, \ldots)$ and $\boldsymbol{y} = (y_1, y_2, \ldots)$, we define

$$\gamma_{s,\ell}(\boldsymbol{x}|\boldsymbol{y}) = \limsup_{n\to\infty} \gamma_{s,\ell}(x^n|y^n), \tag{48}$$

and finally,

$$\gamma(\boldsymbol{x}|\boldsymbol{y}) = \lim_{s\to\infty} \lim_{\ell\to\infty} \gamma_{s,\ell}(\boldsymbol{x}|\boldsymbol{y}). \tag{49}$$

To present the results, we need a few more definitions. Consider the joint parsing of the sequence of pairs, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, let $c(x^n, y^n)$ denote the number of phrases, $c(y^n)$ – the number of distinct y-phrases, $\boldsymbol{y}(j)$ – the j-th distinct y-phrase, $1 \le j \le c(y^n)$, and finally, let $c_j(x^n|y^n)$ denote the number of times $\boldsymbol{y}(j)$ appears as a phrase, or, equivalently, the number of distinct x-phrases that appear jointly with $\boldsymbol{y}(j)$, so that $\sum_{j=1}^{c(y^n)} c_j(x^n|y^n) = c(x^n, y^n)$. Then, we define the conditional LZ complexity [13] as

$$u(x^n, y^n) = \sum_{j=1}^{c(y^n)} c_j(x^n|y^n) \log c_j(x^n|y^n). \tag{50}$$

Let the conditional K–state compressibility of x^n given y^n, denoted $\rho_K(x^n|y^n)$, be defined as in [9, pp. 2245]: A K-state encoder E with side information is defined by a set of six objects $(\Sigma, \mathcal{B}, \mathcal{X}, \mathcal{Y}, p, q)$, where Σ is a finite set of K states, $\mathcal{B}$ is a finite set of binary words (possibly of different lengths, including the null word for idling), $\mathcal{X}$ is the finite alphabet of the source to be compressed, $\mathcal{Y}$ is a finite alphabet of side information, $p : \Sigma \times \mathcal{X} \times \mathcal{Y} \to \mathcal{B}$ is the encoder output function, and $q : \Sigma \times \mathcal{X} \times \mathcal{Y} \to \Sigma$ is the next–state function. When an input sequence $x_1, x_2, \ldots$ and

a side information sequence $y_1, y_2, \ldots$ are fed together, sequentially into E, the encoder outputs a sequence of binary words $b_1, b_2, \ldots, b_i \in \mathcal{B}$, according to

$$b_i = p(\sigma_i, x_i, y_i), \quad \sigma_{i+1} = q(\sigma_i, x_i, y_i), \quad i = 1, 2, ... \tag{51}$$

where σ_i is the state of E at time instant i. The decoder, on the other hand, receives the pair sequence $(b_1, y_1), (b_2, y_2), \ldots$ and reconstructs the source sequence $x_1, x_2, \ldots$. A finite–state encoder E with side information is said to be *information lossless* (IL) if for all $\sigma_1 \in \Sigma$ and all $(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n$, $n \geq 1$, the quadruple $(\sigma_1, \sigma_{n+1}, b^n, y^n)$ uniquely determines x^n, The length function associated with E is defined as $L_E(x^n|y^n) = \sum_{i=1}^n l(b_i)$, where $l(b_i)$ is the length of the binary string $b_i \in \mathcal{B}$. We now define

$$\rho_K(x^n|y^n) = \min_{E \in \mathcal{E}(K)} \frac{L_E(x^n|y^n)}{n} \tag{52}$$

where $\mathcal{E}(K)$ is the class of IL encoders with no more than K states. As shown in [9, eq. (13)],

$$\begin{aligned} \rho_K(x^n|y^n) &\geq \sum_{j=1}^{c(y^n)} [c_j(x^n|y^n) + K^2] \log \frac{c_j(x^n|y^n) + K^2}{4K^2} \\ &\geq u(x^n, y^n) - [c(x^n, y^n) + K^2] \log(4K^2) \\ &\geq u(x^n, y^n) - \frac{n \log(4K^2)}{(1 - \epsilon_n) \log n} - K^2 \log(4K^2). \end{aligned} \tag{53}$$

5.1 Converse Bounds

The converse bounds are as follows.

$$\gamma_{s,\ell}(x^n|y^n) \geq \zeta \cdot \left[\rho_{K(\ell))}(x^n|y^n) - \frac{\log(2s^3 e)}{\ell} \right] - \frac{\log(e^2 2^\zeta)}{n} \tag{54}$$

$$\gamma_{s,\ell}(x^n|y^n) \geq \zeta \Big[u(x^n, y^n) - n\delta_n(s, \ell) \Big], \tag{55}$$

where now $K(\ell)$ is redefined as $([\alpha\beta]^{k+1} - 1)/(\alpha\beta - 1)$ and $\delta_n(s, \ell)$ tends to zero uniformly as $n \to \infty$ for any fixed (s, ℓ). Note that now there is no maximization over all values of ℓ that are divisors of n because now ℓ is a parameter of the model and not an auxiliary parameter as before (indeed, $\gamma_{s,\ell}(x^n|y^n)\gamma_{s,\ell}(x^n|y^n)$ at the l.h.s. depends on ℓ too).

The proof follows the same lines as before with a just few twists. Any FSGM Q with s states incudes a channel from y^n to x^n with the following structure,

$$P(\hat{x}^n|y^n) = \sum_{z_2, z_3, \ldots, z_{n+1}} \prod_{i=1}^{n/\ell} P(\hat{x}_{(i-1)\ell+1}^{i\ell}, z_{i+1}|z_i, y_{(i-1)\ell+1}^{i\ell}). \tag{56}$$

As in the earlier derivation, we have

$$\boldsymbol{E}\{[G_Q(x^n|y^n)]^\zeta\} \geq \frac{2^{-\zeta}}{e^2} \cdot \exp\{-\zeta \ln P(x^n|y^n)\}. \tag{57}$$

Consider now the partitioning $\hat{x}^n$ and y^n into $m = n/\ell$ non–overlapping segments of length ℓ. Then, once again,

$$-\log P(\hat{x}^n|y^n) \geq m[\hat{H}_\ell(x^n|y^n) - \log(s^3 e)], \tag{58}$$

where $\hat{H}_\ell(x^n|y^n)$ is the conditional empirical entropy of ℓ–blocks. Similarly as in the earlier derivation, we can further lower bound the r.h.s. in terms of the conditional compressibility.

$$n\rho_{K(\ell)}(\hat{x}^n|y^n) \leq m\hat{H}_\ell(\hat{x}^n|y^n) + m, \tag{59}$$

and combining this with eq. (58), we obtain

$$-\log P(\hat{x}^n|y^n) \geq n\rho_{K(\ell)}(\hat{x}^n|y^n) - m\log(2s^3 e), \tag{60}$$

which, together with eq. (57), proves the first converse bound, and the second lower bound follows from [9]

$$\begin{aligned} n\rho_{K(\ell)}(\hat{x}^n|y^n) &\geq \sum_{j=1}^{c(y^n)} [c_j(\hat{x}^n|y^n) + K^2(\ell)] \log \frac{c_j(\hat{x}^n|y^n) + K^2(\ell)}{4K^2(\ell)} \\ &\geq \sum_{j=1}^{c(y^n)} c_j(\hat{x}^n|y^n) \log c_j(\hat{x}^n|y^n) - [c(\hat{x}^n, y^n) + K^2(\ell)] \log[4K^2(\ell)] \\ &\geq u(\hat{x}^n, y^n) - \frac{n\log[4K^2(\ell)]}{(1-\epsilon_n)\log n} - K^2(\ell)\log[4K^2(\ell)]. \end{aligned} \tag{61}$$

It follows that the second converse bound holds with

$$\delta_n(s,\ell) = \frac{\log[4K^2(\ell)]}{(1-\epsilon_n)\log n} + \frac{K^2(\ell)\log[4K^2(\ell)]}{n} + \frac{\log(2s^3 e)}{\ell} + \frac{\log(e^2 2^\zeta)}{n}. \tag{62}$$

5.2 Achievability

Following the same steps as in Section 4 and in [10], consider randomly drawing guesses according to the distribution

$$P(\hat{x}^n|y^n) = \frac{2^{-LZ(\hat{x}^n|y^n)}}{\sum_{\tilde{x}^n} 2^{-LZ(\hat{x}^n|y^n)}}, \tag{63}$$

where $LZ(\hat{x}^n|y^n)$ is the length of compressed version of $\hat{x}^n$ given y^n using the conditional version of the LZ78 algorithm [13, p. 460] (see also [12]). It is easy to see that this randomized guessing distribution asymptotically achieves the second lower bound, since

$$LZ(\hat{x}^n|y^n) \leq u(x^n, y^n) + n\epsilon_1(n), \tag{64}$$

where $\epsilon_1(n)$ is of the order of $\log(\log n)/(\log n)$, as shown in [13, p. 460]. Once again, to devise a matching direct in the framework of finite–state machines, we can restart every ℓ–block and apply the random guessing distribution

$$\begin{aligned} P(\hat{x}^n|y^n) &= \prod_{i=0}^{n/\ell-1} \left[\frac{2^{-LZ(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell})}}{\sum_{\tilde{x}^\ell \in \mathcal{X}^\ell} 2^{-LZ(\tilde{x}^\ell|y_{\ell i+1}^{\ell i+\ell})}} \right] \\ &\geq \exp_2 \left\{ - \sum_{i=0}^{n/\ell-1} u(\hat{x}_{\ell i+1}^{\ell i+\ell}, y_{\ell i+1}^{\ell i+\ell}) - n\epsilon_1(\ell) \right\}. \end{aligned} \tag{65}$$

We now need to show that $\sum_{i=0}^{n/\ell-1} u(\hat{x}_{\ell i+1}^{\ell i+\ell}, y_{\ell i+1}^{\ell i+\ell})$ is essentially no larger than $\rho_K(\hat{x}^n|y^n)$ for some K that can be chosen arbitrarily large, provided that n is large enough. Once again, consider the following chain of inequalities for a given positive integer K:

$$\begin{aligned} & \sum_{i=0}^{n/\ell-1} \sum_{j=1}^{c(y_{\ell i+1}^{\ell i+\ell})} [c_j(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) + K^2] \log \left[\frac{c_j(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) + K^2}{4K^2} \right] \\ \leq\ & \ell \sum_{i=0}^{n/\ell-1} \rho_K(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) \\ =\ & \ell \cdot \sum_{i=0}^{n/\ell-1} \min_{E \in \mathcal{E}(K)} \rho_E(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell})) \\ =\ & \sum_{i=0}^{n/\ell-1} \min_{E \in \mathcal{E}(s)} \sum_{j=1}^{\ell} l[p(\sigma_{\ell i+j}, \hat{x}_{\ell i+j}, y_{\ell i+j})] \\ \leq\ & \min_{E \in \mathcal{E}(K)} \sum_{i=0}^{n/\ell-1} \sum_{j=1}^{\ell} l[p(\sigma_{\ell i+j}, \hat{x}_{\ell i+j}, y_{\ell i+j})] \\ =\ & \min_{E \in \mathcal{E}(K)} \sum_{i=1}^{n} l[p(\sigma_i, \hat{x}_i, y_i)] \\ =\ & n \cdot \rho_K(\hat{x}^n|y^n). \end{aligned} \tag{66}$$

Thus,

$$\begin{aligned}
\sum_{i=0}^{n/\ell-1}\sum_{j=1}^{c(y_{\ell i+1}^{\ell i+\ell})} c_j(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) \log c_j(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) &\leq n\cdot\rho_K(\hat{x}^n|y^n) + \sum_{i=0}^{n/\ell-1}\sum_{j=1}^{c(y_{\ell i+1}^{\ell i+\ell})} c_j(\hat{x}_{\ell i+1}^{\ell i+\ell}|y_{\ell i+1}^{\ell i+\ell}) \log(4K^2) \\
&\leq n\cdot\rho_K(\hat{x}^n|y^n) + \frac{n}{\ell}\cdot\frac{\ell\log(\alpha\beta)\cdot\log(4K^2)}{(1-\epsilon_\ell)\log\ell} \\
&= n\cdot\left[\rho_K(\hat{x}^n|y^n) + \frac{\log(\alpha\beta)\log(4K^2)}{(1-\epsilon_\ell)\log\ell}\right] \qquad (67)
\end{aligned}$$

and the remaining steps are similarly as before.

References

[1] E. Arikan, "An inequality on guessing and its application to sequential decoding," *IEEE Trans. Inform. Theory*, vol. IT–42, no. 1, pp. 99–105, January 1996.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Second Edition, John Wiley & Sons, Hoboken, New Jersey, U.S.A., 2006.

[3] M. Feder, "Gambling using a finite–state machine," *IEEE Trans. on Inform. Theory*, vol. 37, no. 5, pp. 1459–1465, September 1991.

[4] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1258–1270, July 1992.

[5] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. Inform. Theory*, vol. IT–22, no. 1, pp. 75–81, January 1976.

[6] J. L. Massey, "Guessing and entropy," *Proc. IEEE International Symposium on Information Theory (ISIT '94)*, p. 204, 1994.

[7] N. Merhav, "Universal coding with minimum probability of code word length overflow," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 556–563, May 1991.

[8] N. Merhav, "Perfectly secure encryption of individual sequences," *IEEE Trans. Inform. Theory*, vol. 59, no. 3, pp. 1302–1310, March 2013.

[9] N. Merhav, "Universal detection of messages via finite–state channels," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2242–2246, September 2000.

[10] N. Merhav and A. Cohen, "Universal randomized guessing with application to asynchronous decentralized brute–force attacks," to appear in *IEEE Trans. Inform. Theory*, 2019.

[11] S. Salamatian, W. Huleihel, A. Beirami, A. Cohen, and M. Médard, "Why botnets work: distributed brute–force attacks need no synchronization," to appear in *IEEE Trans. Inform. Forensics and Security*, September 2019.

[12] T. Uyematsu and S. Kuzuoka, "Conditional Lempel-Ziv complexity and its application to source coding theorem with side information," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 86, no. 10, pp. 2615–2617, October 2003.

[13] J. Ziv, "Universal decoding for finite–state channels," *IEEE Trans. Inform. Theory*, vol. IT–31, no. 4, pp. 453–460, July 1985.

[14] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inform. Theory*, vol. IT–24, no. 5, pp. 530–536, September 1978.