Pushdown Dimension

David Doty* Jared Nichols†

Abstract

Resource-bounded dimension is a notion of computational information density of infinite sequences based on computationally bounded gamblers. This paper develops the theory of pushdown dimension and explores its relationship with finite-state dimension. The pushdown dimension of any sequence is trivially bounded above by its finite-state dimension, since a pushdown gambler can simulate any finite-state gambler. We show that for every rational $0 < d < 1$, there exists a sequence with finite-state dimension d whose pushdown dimension is at most $d/2$. This provides a stronger quantitative analogue of the well-known fact that pushdown automata decide strictly more languages than finite-state automata.

Keywords: resource-bounded dimension, martingale, finite-state, pushdown

1 Introduction

The dimension of a set of points was first explored by Hausdorff [9, 16], who showed that there exist sets of points with fractional dimension, now termed *fractals*. Infinite sequences over a finite alphabet can be viewed as points on the unit interval. Lutz [28] showed that the Hausdorff dimension of a set of infinite sequences could be characterized by the rate at which money could be taken away from a gambler that is trying to make unbounded money by betting on all the sequences in the set. In other words, the higher the dimension of a set, the more random and unpredictable are its elements, and so the more difficult it is to make money betting on its elements (a precise definition follows in later sections).

Though all singleton sets of sequences – i.e., all individual points – have Hausdorff dimension 0, by restricting the computational power of the gambler, individual sequences can be assigned a non-zero dimension. The theory of *resource-bounded dimension* has shed new and unexpected light on the connections between fractal dimensions – such as Hausdorff dimension [16,29] and packing dimension [2,40,41] – and algorithmic compression [5–7,27,29,30,42], prediction [11, 18], and computational complexity [1, 12–15, 17, 19–23, 28, 32–35]. Resource-bounded dimension is a measure of the density of information or randomness in a sequence that is not exploitable by a gambler whose computational power is limited by the resource bound. For example, the *finite-state dimension* of a sequence [5] is the degree to which the

*Corresponding author. Department of Computer Science, Iowa State University, Ames, IA, 50011, USA. ddoty@iastate.edu. This research was funded in part by grant number 9972653 from the National Science Foundation as part of their Integrative Graduate Education and Research Traineeship (IGERT) program.

†Department of Computer Science, Iowa State University, Ames, IA, 50011, USA. This research was supported in part by National Science Foundation Grants 9988483 and 0344187.

sequence appears random to finite-state machines. This paper concerns the *pushdown dimension* of a sequence [36], the degree to which the sequence appears random to pushdown machines (finite-state machines equipped with an infinite stack memory).

For a sequence S and a computational resource bound Δ (such as finite-state or polynomial time), let $\dim_\Delta(S)$ denote the Δ-*dimension* of S, the information density of S as perceived by Δ-bounded machines; see [28] for a full definition. If Δ is more powerful than Δ', then

$$0 \leq \dim_\Delta(S) \leq \dim_{\Delta'}(S) \leq 1.$$

Intuitively, a more powerful gambler can make at least as much money as a less powerful gambler, and hence can tolerate a bigger loss of its winnings on each bet and still make unbounded money.

A finite-state gambler is a finite-state machine that bets money on the next character according to its current state. Weighted finite automata, of which finite-state gamblers are a special case, have also been studied in other contexts [25, 39]. A pushdown gambler is a finite-state gambler augmented with an infinite stack memory, and it is allowed to vary its state transition and its bet at each state depending on the character appearing at the top of the stack. Since any finite-state gambler can be simulated exactly by a pushdown gambler that makes no use of its stack, pushdown gamblers are at least as powerful as finite-state gamblers, and hence $\dim_{\mathrm{PD}}(S) \leq \dim_{\mathrm{FS}}(S)$ for all sequences S.

Since pushdown machines are known to decide strictly more languages than finite-state machines [24], it seems natural to conjecture that there exist sequences with pushdown dimension strictly less than their finite-state dimension. We show this conjecture to be true. Specifically, for every rational $0 < d < 1$, there exists a sequence S with $\dim_{\mathrm{FS}}(S) = d$ such that $\dim_{\mathrm{PD}}(S) \leq \frac{1}{2}\dim_{\mathrm{FS}}(S)$. Thus, using the theory of resource-bounded dimension, we achieve a stronger *quantitative* separation of the relative computational power of pushdown machines and finite-state machines.

Our proof technique also gives a new method by which to construct sequences of arbitrary rational finite-state dimension. Given the binary alphabet $\{0,1\}$, we construct sequences over the alphabet $\Sigma \subseteq \{0,1\}^l$, for a positive integer l. That is, binary strings of length l are interpreted as individual characters of Σ. We show that if a sequence $S \in \Sigma^\infty$ is simultaneously interpreted as a sequence $T \in \{0,1\}^\infty$, then the finite-state dimension of T is "scaled down" by the appropriate amount from the finite-state dimension of S. In particular, if $S \in \Sigma^\infty$ is a sequence with finite-state dimension 1, and $|\Sigma| = 2^k \leq 2^l$, then the finite-state dimension of the sequence $T \in \{0,1\}^\infty$ is k/l.

2 Preliminaries

We write $\mathbb{Q}$ for the set of all rational numbers, $\mathbb{Z}$ for the set of all integers, $\mathbb{N}$ for the set of all natural numbers, $\mathbb{Z}^+$ for the set of all positive integers, and $\mathbb{R}^+$ for the set of all positive real numbers. For $r \in \mathbb{R}^+$, let $\log r = \log_2 r$. Given a finite set Ω, let $\Delta_{\mathbb{Q}}(\Omega)$ be the set of all rational probability measures over Ω.

Let Σ be a finite alphabet of characters. Σ^* is the set of all finite strings over from Σ. The length of a string $w \in \Sigma^*$ is denoted by $|w|$. λ denotes the empty string. For $l \in \mathbb{N}$, Σ^l denotes the set of all strings $w \in \Sigma^*$ such that $|w| = l$. $\overline{w}$ denotes the reverse of w. For $w, y \in \Sigma^*$, wy denotes the concatenation of w and y. For $i \geq 0$, w^i is recursively defined $w^0 = \lambda$ and $w^i = w^{i-1}w$ for $i \geq 1$. Σ^∞ is the set of all infinite sequences over Σ. For $S \in \Sigma^\infty$ or Σ^* and

$i, j \in \mathbb{N}$, we write $S[i]$ to denote the i^{th}h character of S, with $S[0]$ being the leftmost character, and we write $S[i\,..\,j]$ to denote the substring consisting of the i^{th} through j^{th} characters of S, with $S[i\,..\,j] = \lambda$ if $i > j$. We write $S \upharpoonright n$ to denote $S[0\,..\,n-1]$, the n^{th} prefix of S. If $n < 0, S \upharpoonright n = \lambda$. For $S \in \Sigma^\infty$, we write $S[n\,..]$ to denote S without its first n characters; i.e., $S[0\,..\,n-1]S[n\,..] = S$.

Let $w \in \Sigma^l$ and $S \in \Sigma^\infty$. Define $\#(w, S \upharpoonright n)$ to be the number of times w appears as a substring of $S \upharpoonright n$, i.e.,

$$\#(w, S \upharpoonright n) = |\{\ i \in \mathbb{N}\ \mid 0 \leq i \leq n - |w| \text{ and } w = S[i\,..\,i + |w| - 1]\ \}|.$$

Let the *frequency of w in $S \upharpoonright n$* be defined

$$\text{freq}(w, S \upharpoonright n) \triangleq \frac{\#(w, S \upharpoonright n)}{n - |w| + 1}.$$

Let the *frequency of w in S* be defined

$$\text{freq}(w, S) \triangleq \lim_{n\to\infty} \text{freq}(w, S \upharpoonright n) = \lim_{n\to\infty} \frac{\#(w, S \upharpoonright n)}{n}$$

when this limit exists. Note that it need not exist; consider, for instance, $S = 01^9 0^{90} 1^{900} 0^{9000} \ldots$, where $\text{freq}(0, S \upharpoonright n)$ oscillates forever between 0.1 and 0.9 as $n \to \infty$.

We state the following obvious lemma without proof, which states that adding a finite prefix to a sequence cannot alter the limiting frequency of any substring.

Lemma 2.1. *Let $S \in \Sigma^\infty$ and $w, u \in \Sigma^*$. Then, if* $\text{freq}(w, S)$ *is defined,*

$$\text{freq}(w, S) = \text{freq}(w, uS).$$

A sequence $S \in \Sigma^\infty$ is *(Borel) normal* if, for every $w \in \Sigma^*$,

$$\text{freq}(w, S) = |\Sigma|^{-|w|}.$$

In other words, S is normal if, for every string length l, all strings of length l occur with the same frequency.

Note that given $S \upharpoonright n$ and $l \leq n$, $\text{freq}(\cdot, S \upharpoonright n)$, when restricted to input strings of length l, defines a probability measure on the set Σ^l. Accordingly, we can speak of the entropy of this probability distribution. Let the l^{th} *normalized entropy of S* be denoted

$$H_l(S) \triangleq \frac{1}{l \log |\Sigma|} \liminf_{n\to\infty} \sum_{w \in \Sigma^l} \text{freq}(w, S \upharpoonright n) \log \frac{1}{\text{freq}(w, S \upharpoonright n)}.$$

Note that $H_l(S)$ exists even if $\text{freq}(w, S)$ does not, since the limit inferior is being used. $H_l(S)$ is the limiting entropy of the distribution of strings of length l in S, normalized by the term $\frac{1}{l \log |\Sigma|}$ to fall between 0 and 1. Thus, the more uniformly distributed are the strings of length l in S, the closer $H_l(S)$ is to 1. Let the *normalized entropy rate of S* be denoted

$$H(S) \triangleq \lim_{l\to\infty} H_l(S).$$

Ziv and Lempel [42] showed that the limit above exists. The closer $H(S)$ is to 1, the closer S is to normal, and $H(S) = 1$ if and only if S is normal (see [37] or [3]).

3 Dimension

3.1 Finite-State Dimension

See [28, 29] for a more comprehensive account of the theory of resource-bounded dimension. Finite-state dimension is defined as in [5]. In order to define finite-state dimension, we must first define martingales, s-gales, and finite-state gamblers.

Intuitively, a martingale is a strategy for betting in the following game. The gambler starts with some initial amount of money $d(\lambda)$, termed *capital*, and it reads an infinite sequence S of bits. The value $d(w)$ represents the capital the martingale has after reading the string w. At each step, the gambler bets some fraction of its capital on 0, and the remainder on 1. The capital that was bet on the bit that appears next is doubled, and the remaining capital is lost. Thus the martingale will make more money on a sequence if a larger fraction of capital is placed on the bits that actually occur in the sequence. All of the gambler's money must be bet, but it can "bet nothing" by betting half of its capital on each bit.

An s-gale is a martingale in which the amount of capital the gambler bet on the bit that occurred is multiplied by 2^s, as opposed to simply 2, after each bit. The lower the value of s, the faster money is taken away. Note that if a gambler's martingale is d, then, for all $s \in [0, \infty)$, its s-gale is given by $d^{(s)}(w) = 2^{(s-1)|w|}d(w)$.

Definition 3.1. (martingale and s-gale)

1. Given $s \in \mathbb{R}^+$, an *s-gale* is a function $d : \Sigma^* \to [0, \infty)$ that, for all $w \in \Sigma^*$, satisfies

$$d(w) = 2^{-s} \sum_{a \in \Sigma} d(wa).$$

2. A *martingale* is a 1-gale.

Definition 3.2. Let $P \subseteq \Sigma^*$. P is a *prefix set* if no string in P is a proper prefix of any other string in P.

Note that for any $l \in \mathbb{Z}^+$, Σ^l is a prefix set. The following generalization of the Kraft inequality was given in [29].

Lemma 3.3. *Let $s \in [0, \infty)$. If $d^{(s)}$ is an s-gale and $A \subseteq \{0,1\}^*$ is a prefix set, then for all $u \in \{0,1\}^*$,*

$$\sum_{w \in A} 2^{-s|w|} d^{(s)}(uw) \leq d^{(s)}(u).$$

Corollary 3.4. *Let $s \in [0, \infty)$. If $d^{(s)}$ is an s-gale and $A \subseteq \{0,1\}^*$ is a prefix set, then*

$$\sum_{w \in A} 2^{-s|w|} d^{(s)}(w) \leq 1.$$

A finite-state gambler, informally, is a gambler whose martingale can be computed by a finite-state machine.

Definition 3.5. (finite-state gambler)
A *finite-state gambler* is a 5-tuple $G = (Q, \Sigma, \delta, \beta, q_0)$ where

- Q is a finite set of *states*,
- Σ is the finite *input alphabet*,
- $\delta : Q \times \Sigma \to Q \cup \{\bot\}$ is the *transition function*,
- $\beta : Q \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*,
- $q_0 \in Q$ is the *start state*.

We write FSG to denote the set of all finite-state gamblers.

If $\delta(q, a) = \bot$, for some $q \in Q$ and $a \in \Sigma$, then that transition is undefined. We extend δ to take strings as input with the function $\delta^* : Q \times \Sigma^* \to Q$ defined by the recursion

$$\begin{aligned} \delta^*(q, \lambda) &= q, \\ \delta^*(q, wa) &= \delta(\delta^*(q, w), a). \end{aligned}$$

for all $q \in Q, w \in \Sigma^*$, and $a \in \Sigma$. The function δ^* is then abbreviated δ and $\delta(q_0, w)$ is abbreviated $\delta(w)$. Intuitively, this allows us to identify $\delta(w)$ as "the state G is in after reading string w."

Intuitively, the martingale for a finite-state gambler G is determined as follows. A finite-state gambler $G = (Q, \Sigma, \delta, \beta, q_0)$ starts in state q_0 with initial capital 1. Assuming that after some time G has capital c and is in state q, the bet (the fraction of current capital) that G makes on each character $a \in \Sigma$ is given by $\beta(q)(a)$. Assuming the character b appears next in the sequence, G then transitions to state $\delta(q, b)$, and its capital becomes $c \cdot \beta(q)(b) \cdot |\Sigma|$. If we are considering instead the s-gale for G, its capital becomes $c \cdot \beta(q)(b) \cdot |\Sigma|^s$.

Definition 3.6. (finite-state martingale and s-gale)

Let G be a finite-state gambler.

1. The *martingale for G* is the function $d_G : \Sigma^* \to [0, \infty)$ defined by

$$\begin{aligned} d_G(\lambda) &= 1 \\ d_G(wa) &= d_G(w) \cdot \beta(\delta(w))(a) \cdot |\Sigma| \end{aligned}$$

   for all $w \in \Sigma^*$ and $a \in \Sigma$.

2. The *s-gale for G* is the function $d_G^{(s)} : \Sigma^* \to [0, \infty)$ defined by

$$\begin{aligned} d_G^{(s)}(\lambda) &= 1 \\ d_G^{(s)}(wa) &= d_G(w) \cdot \beta(\delta(w))(a) \cdot |\Sigma|^s \end{aligned}$$

   for all $s \in \mathbb{R}^+$, $w \in \Sigma^*$, and $a \in \Sigma$.

Let $G = (Q, \Sigma, \delta, \beta, q_0)$ be a finite-state gambler. For $q \in Q$, let $d_{G,q}$ be the martingale for G if G is started in state q instead of q_0, and let $d_{G,q}^{(s)}$ be the s-gale defined in the same way.

We now define finite-state dimension of a sequence to be the smallest s for which a finite-state gambler makes infinite money on the sequence, even with tax rate s.

Definition 3.7. (finite-state dimension)

Let $S \in \Sigma^\infty$. The *finite-state dimension of S* is

$$\dim_{\text{FS}}(S) = \inf \left\{ s \in [0,\infty) \;\middle|\; (\exists G \in \text{FSG}) \limsup_{n\to\infty} d_G^{(s)}(S \upharpoonright n) = \infty \right\}.$$

Thus, if $s > \dim_{\text{FS}}(S)$, then there is a finite-state gambler G that s-succeeds on S, meaning G can make unlimited money betting on S, even if its winnings are multiplied by $|\Sigma|^{s-1}$ after every character.

Let Σ, Σ' be finite alphabets with $\Sigma \subseteq \Sigma'$, and let $S \in \Sigma^\infty$. Let $\dim_{\text{FS}}^{(\Sigma')}(S)$ be the finite-state dimension of S when considered as a sequence over the alphabet Σ', even though S is actually composed only of characters from Σ. The next lemma shows that $\dim_{\text{FS}}^{(\Sigma')}(S)$ is completely determined by $\dim_{\text{FS}}^{(\Sigma)}(S)$.

Lemma 3.8. *Let $S \in \Sigma^\infty$, and let $\Sigma \subseteq \Sigma'$. Then*

$$\dim_{\text{FS}}^{(\Sigma')}(S) = \frac{\log|\Sigma|}{\log|\Sigma'|}\dim_{\text{FS}}^{(\Sigma)}(S).$$

Proof. We first show that $\dim_{\text{FS}}^{(\Sigma')}(S) \le \frac{\log|\Sigma|}{\log|\Sigma'|}\dim_{\text{FS}}^{(\Sigma)}(S)$. Let $s > \dim_{\text{FS}}^{(\Sigma)}(S)$. Then there exists a finite-state gambler $G = (Q, \Sigma, \delta, \beta, q_0)$ that s-succeeds on S. Construct the finite-state gambler $G' = (Q', \Sigma', \delta', \beta', q_0')$ as follows

- $Q' = Q$,
- $\delta'(q,a) = \begin{cases} \delta(q,a), & \text{if } a \in \Sigma \\ \perp, & \text{otherwise} \end{cases}$,
- $\beta'(q)(a) = \begin{cases} \beta(q)(a), & \text{if } a \in \Sigma \\ 0, & \text{otherwise} \end{cases}$,
- $q_0' = q_0$.

Since S contains no characters from $\Sigma' - \Sigma$, for all $n \in \mathbb{N}$,

$$d_{G'}(S \upharpoonright n) = \left(\frac{|\Sigma'|}{|\Sigma|}\right)^n d_G(S \upharpoonright n).$$

Let $t = s\frac{\log|\Sigma|}{\log|\Sigma'|}$. Then

$$\begin{aligned} d_{G'}^{(t)}(S \upharpoonright n) &\triangleq |\Sigma'|^{(t-1)n} d_{G'}(S \upharpoonright n) \\ &= |\Sigma'|^{(t-1)n} \left(\frac{|\Sigma'|}{|\Sigma|}\right)^n d_G(S \upharpoonright n) \\ &= \left(\frac{|\Sigma'|^t}{|\Sigma|}\right)^n d_G(S \upharpoonright n) \\ &= |\Sigma|^{(s-1)n} d_G(S \upharpoonright n) \qquad \text{substituting } t = s\frac{\log|\Sigma|}{\log|\Sigma'|} \\ &\triangleq d_G^{(s)}(S \upharpoonright n). \end{aligned}$$

Thus G' t-succeeds on S, since G s-succeeds on S. Since this holds for every $s > \dim_{\text{FS}}^{(\Sigma)}(S)$, $\dim_{\text{FS}}^{(\Sigma')}(S) \leq \frac{\log|\Sigma|}{\log|\Sigma'|}\dim_{\text{FS}}^{(\Sigma)}(S)$.

We next show that $\dim_{\text{FS}}^{(\Sigma')}(S) \geq \frac{\log|\Sigma|}{\log|\Sigma'|}\dim_{\text{FS}}^{(\Sigma)}(S)$. Let $t > \dim_{\text{FS}}^{(\Sigma')}(S)$. Then there exists a finite-state gambler $G = (Q, \Sigma', \delta, \beta, q_0)$ that t-succeeds on S. Since S contains no characters from $\Sigma' - \Sigma$, assume without loss of generality that $\beta(q, a) = 0$ for all $q \in Q$ and all $a \in \Sigma' - \Sigma$. This assumption can be made for the following reason. If a gambler does bet non-zero capital on $a \in \Sigma' - \Sigma$, we can always construct a gambler that takes the capital G bets on a and uniformly distributes it to the remaining characters in Σ. Since a does not appear in S, this new gambler will make strictly more money than the old one, and hence will s-succeed whenever the old gambler does.

Then a straightforward reversal of the previous direction of the proof suffices to show that there is a gambler $G' = (Q', \Sigma, \delta', \beta', q_0')$ that s-succeeds on S, where $s = t\frac{\log|\Sigma'|}{\log|\Sigma|}$. This establishes that $\dim_{\text{FS}}^{(\Sigma')}(S) \geq \frac{\log|\Sigma|}{\log|\Sigma'|}\dim_{\text{FS}}^{(\Sigma)}(S)$. □

3.2 Pushdown Dimension

Pushdown gamblers are nothing more than finite-state gamblers that make use of an unbounded stack memory, the top character of which can be used to inform the transition and betting functions. Additionally, a pushdown gambler is allowed to delay reading the next character of the input – it reads λ from the input – in order to alter the contents of the stack. During such a λ-transition, the gambler's capital remains unchanged.

Definition 3.9. (pushdown gambler)

A pushdown gambler is a 7-tuple $P = (Q, \Sigma, \Gamma, \delta, \beta, q_0, z)$, where

- Q is a finite set of *states*,
- Σ is the finite *input alphabet*,
- Γ is the finite *stack alphabet*,
- $\delta : Q \times \Gamma \times (\Sigma \cup \{\lambda\}) \to (Q \times \Gamma^*) \cup \{\perp\}$ is the *transition function*,
- $\beta : Q \times \Gamma \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*,
- $q_0 \in Q$ is the *start state*,
- $z \in \Gamma$ is the *stack start symbol*.

We write PDG to denote the set of all pushdown gamblers.

Note that the transition function δ outputs a next state and a string $w \in \Gamma^*$. The interpretation is that the top character on the stack is always popped and replaced with the string w. If $a \in \Gamma$ is the symbol currently on top of the stack, and P needs to add a character $b \in \Gamma$ to the top, it pushes the string ba. If P needs to leave the contents of the stack unchanged, it pushes the string a. If P needs to pop a character, it pushes the string λ. The strings are pushed in reverse order; the last character of the string is pushed first.

Note also that the transition function δ accepts λ as an input character in addition to elements of Σ. This is because P has the option not to read an input character and instead only to alter the stack. To enforce determinism, we require at least one of the following hold for all $q \in Q$ and all $a \in \Gamma$.

1. $\delta(q, a, \lambda) = \bot$, or
2. $\delta(q, a, b) = \bot$ for all $b \in \Sigma$.

The determinism condition requires that the pushdown gambler cannot have the nondeterministic choice to read 0 or 1 characters; the number of characters read is entirely a function of the gambler's state and the character at the top of the stack.

We must also handle the special case that the stack start symbol gets popped. Since this represents the bottom of the stack, we restrict δ so that z cannot be removed from the bottom. We restrict δ so that, for every $q \in Q$ and $a \in \{\lambda\} \cup \Sigma$, either

$$\delta(q, z, a) = \bot$$

or

$$\delta(q, z, a) = (q', vz)$$

where $q' \in Q$ and $v \in \Gamma^*$.

As before, if $\delta(q, a, b) = \bot$ for some $q \in Q, a \in \Gamma$, and $b \in \{\lambda\} \cup \Sigma$, then that transition is undefined. We extend δ to the transition function

$$\delta^* : Q \times \Gamma^+ \times (\{\lambda\} \cup \Sigma) \to (Q \times \Gamma^*) \cup \{\bot\},$$

defined for all $q \in Q, a \in \Gamma, v \in \Gamma^*$, and $b \in \Sigma$ as follows.

$$\delta^*(q, av, b) = \begin{cases} (\delta_Q(q, a, b), \delta_\Gamma(q, a, b)v), & \text{if } \delta(q, a, b) \neq \bot; \\ \bot, & \text{otherwise.} \end{cases}$$

where $\delta(q, a, b) = (\delta_Q(q, a, b), \delta_\Gamma(q, a, b))$. δ^* is then abbreviated as δ. We then use the extended transition function

$$\delta^{**} : Q \times \Gamma^+ \times \Sigma^* \to (Q \times \Gamma^*) \cup \{\bot\},$$

in analogy to that used with finite-state gamblers, defined for all $q \in Q, a \in \Gamma, v \in \Gamma^*, w \in \Sigma^*$, and $b \in \Sigma$ by

$$\begin{aligned} \delta^{**}(q, av, \lambda) &= \begin{cases} \delta^{**}(\delta(q, av, \lambda), \lambda), & \text{if } \delta(q, av, \lambda) \neq \bot \\ (q, av), & \text{otherwise} \end{cases}, \\ \delta^{**}(q, av, wb) &= \begin{cases} \delta^{**}(\delta(\delta^{**}(q, av, w), \lambda), b), & \text{if } \delta^{**}(q, av, w) \neq \bot \text{ and } \delta(\delta^{**}(q, av, w), \lambda) \neq \bot \\ \delta(\delta^{**}(q, av, w), b), & \text{if } \delta^{**}(q, av, w) \neq \bot \text{ and } \delta(\delta^{**}(q, av, w), \lambda) = \bot \\ \bot, & \text{otherwise} \end{cases}. \end{aligned}$$

We then abbreviate δ^{**} to δ, and $\delta(q_0, z, w)$ to $\delta(w)$. Informally, this allows us to use $\delta(w)$ as shorthand for "the configuration (state and contents of the stack) of the gambler P after reading string w".

We also extend β for convenience to the function

$$\beta^* : Q \times \Gamma^+ \to \Delta_{\mathbb{Q}}(\Sigma),$$

defined for all $q \in Q$, $a \in \Gamma$, and $v \in \Gamma^*$ by

$$\beta^*(q, av) = \beta(q, a).$$

β^* is then abbreviated β. $\beta^*(q, av)(b)$ means, informally, "The amount bet on character b when in state q, when the string av is on the stack." Note that only the top character a of av can affect any single bet, but for the purpose of examining multiple steps of the gambler, it is necessary to keep track of the entire contents of the stack, since they may change from step to step.

Given a pushdown gambler P, define the martingale d_P and the s-gale d_P exactly as in the case of finite-state gambler. Pushdown dimension is then defined in exact analogy to finite-state dimension.

Definition 3.10. (pushdown dimension)
Let $S \in \Sigma^\infty$. The *pushdown dimension of* S is

$$\dim_{\mathrm{PD}}(S) = \inf \left\{ s \in [0,\infty) \;\middle|\; (\exists P \in \mathrm{PDG}) \limsup_{n\to\infty} d_P^{(s)}(S \upharpoonright n) = \infty \right\}.$$

4 Finite-State Dimension versus Pushdown Dimension

In this section we show that finite-state dimension may exceed pushdown dimension.

We use the technique mentioned in the introduction to construct a sequence over the "alphabet" $\Sigma \subsetneq \{0,1\}^l$ of arbitrary rational finite-state dimension. We then add "marker characters" – elements of $\{0,1\}^l$ that are not contained in Σ – to this sequence, without changing its finite-state dimension. These markers are intended to help a pushdown gambler delimit certain points in the sequence when it should stop pushing bits on its stack and begin popping the contents of its stack to bet better than a finite-state gambler could. The bits following the marker are simply the reverse of the bits before the marker, so the pushdown gambler knows exactly how to bet to double its money on every bit until the stack is empty, at which point it begins anew. Because the pushdown gambler acts like a finite-state gambler for half of the sequence, and it bets optimally on the other half of the sequence, the sequence has pushdown dimension no greater than half of its finite-state dimension.

4.1 Marker Characters and Finite-State Dimension

This section establishes that adding marker characters to a sequence, where the marker is not in the alphabet of the sequence, does not alter the finite-state dimension of the sequence, as long as the markers are spaced far enough apart. In other words, the addition of the markers cannot significantly hurt or help a finite-state gambler.

Recall that, for $S \in \Sigma^\infty$,

$$H(S) \triangleq \lim_{l\to\infty} \frac{1}{l \log|\Sigma|} \liminf_{n\to\infty} \sum_{w\in\Sigma^l} \mathrm{freq}(w, S \upharpoonright n) \log \frac{1}{\mathrm{freq}(w, S \upharpoonright n)}.$$

Let

$$\widehat{H}(S) \triangleq \lim_{l\to\infty} \frac{1}{l \log|\Sigma|} \limsup_{n\to\infty} \sum_{w\in\Sigma^l} \mathrm{freq}(w, S \upharpoonright n) \log \frac{1}{\mathrm{freq}(w, S \upharpoonright n)}.$$

Ziv and Lempel [42] showed that

$$\widehat{\rho}_{\mathrm{FS}}(S) = \widehat{H}(S),$$

where $\widehat{\rho}_{\mathrm{FS}}(S)$ is the optimal compression ratio achievable by any finite-state compressor (see [42] or [5] for a more complete description). Dai, Lathrop, Lutz, and Mayordomo [5] showed that $\dim_{\mathrm{FS}}(S)$ is identical to a slightly modified form of $\widehat{\rho}_{\mathrm{FS}}(S)$.

A straightforward modification of the proof of Lempel and Ziv, combined with the result of [5], yields the following lemma. (See also [3] for a self-contained proof.)

Lemma 4.1 (Ziv and Lempel [42])**.** *Let $S \in \Sigma^\infty$. Then*

$$\dim_{\mathrm{FS}}(S) = H(S).$$

Corollary 4.2. *Let $S \in \Sigma^\infty$. Then*

$$\dim_{\mathrm{FS}}(S) = 1 \iff S \text{ is normal.}$$

Let Σ be an alphabet. Let $\Sigma_m = \Sigma \cup \{m\}$, where $m \notin \Sigma$ is a *marker character.* Recall that $\dim_{\mathrm{FS}}^{(\Sigma_m)}(S)$ is the finite-state dimension of S when considered as a sequence over the alphabet Σ_m, even if it is actually composed only of characters from $\Sigma \subsetneq \Sigma_m$. The next lemma shows that the addition of marker characters to a sequence cannot alter its finite-state dimension, as long as the marker characters are placed increasingly far apart.

Lemma 4.3. *Let $S \in \Sigma^\infty$. Let $S' \in \Sigma_m^\infty$ be constructed from S by inserting the character m after positions $i_1 < i_2 < i_3 \ldots$ in S such that the function $f(j) = i_{j+1} - i_j$ is nondecreasing and unbounded. Then*

$$\dim_{\mathrm{FS}}^{(\Sigma_m)}(S') = \dim_{\mathrm{FS}}^{(\Sigma_m)}(S).$$

Proof. Let S and S' be as in the statement of the lemma. Let $l \in \mathbb{Z}^+$, and let $w \in \Sigma^l$.

Let there be k_n insertions of the marker character m in $S \upharpoonright n$ (i.e., the insertion indices satisfy $1 \leq i_1 < i_2 < \ldots < i_{k_n} \leq n < i_{k_n+1}$). Then $S' \upharpoonright (n+k_n)$ is the prefix of S' "corresponding" to $S \upharpoonright n$. Note that $\mathrm{freq}(m, S' \upharpoonright (n+k_n)) = \frac{k_n}{n+k_n}$.

Since $f(j) = i_{j+1} - i_j$ is non-decreasing and unbounded, for all $p \in \mathbb{N}$, there exists $n_p \in \mathbb{N}$ such that all markers after position n_p are at least p characters apart. Hence $\mathrm{freq}(m, S'[n_p\,..]) \leq \frac{1}{p}$. By Lemma 2.1, $\mathrm{freq}(m, S') \leq \frac{1}{p}$. Since this holds for all $p \in \mathbb{N}$, $\mathrm{freq}(m, S') = 0$. Since $\mathrm{freq}(m, S' \upharpoonright (n+k_n)) = \frac{k_n}{n+k_n}$, then $k_n = o(n)$; k_n grows strictly slower than n.

Since there are k_n occurrences of m in $S' \upharpoonright (n+k_n)$, there are $k_n(l-1)$ substrings of length l in $S \upharpoonright n$ that could have been changed by having an m inserted into them. In the worst case, every one of these substrings was our chosen string w. Thus

$$\underbrace{\#(w, S' \upharpoonright (n+k_n))}_{\#\text{ of } w \text{ in } S' \upharpoonright (n+k_n)} \geq \underbrace{\#(w, S \upharpoonright n)}_{\#\text{ of } w \text{ in } S \upharpoonright n} - \underbrace{k_n(l-1)}_{\substack{\#\text{ of } w \text{ in } S \upharpoonright n \text{ that} \\ \text{could have changed}}} \tag{4.1}$$

Since $w \in \Sigma^l$, it does not contain an m. Adding m's to S cannot add more w's to S. Thus

$$\#(w, S' \upharpoonright (n+k_n)) \leq \#(w, S \upharpoonright n) \tag{4.2}$$

Recall that $k_n = o(n)$. Thus

$$\begin{aligned}
& \lim_{n\to\infty} \big(\mathrm{freq}(w, S' \upharpoonright (n+k_n)) - \mathrm{freq}(w, S \upharpoonright n)\big) \\
= & \lim_{n\to\infty} \left(\frac{\#(w, S' \upharpoonright (n+k_n))}{n+k_n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \\
\geq & \lim_{n\to\infty} \left(\frac{\#(w, S \upharpoonright n) - k_n(l-1)}{n+k_n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \qquad \text{by (4.1)} \\
= & \lim_{n\to\infty} \left(\frac{\#(w, S \upharpoonright n) - k_n(l-1)}{n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \qquad \text{since } k_n = o(n) \\
= & \lim_{n\to\infty} \left(\frac{-k_n(l-1)}{n-l+1} \right) \\
= & \ 0, \quad \text{since } k_n = o(n)
\end{aligned}$$

and

$$\begin{aligned}
& \lim_{n\to\infty} \big(\mathrm{freq}(w, S' \upharpoonright (n+k_n)) - \mathrm{freq}(w, S \upharpoonright n)\big) \\
= & \lim_{n\to\infty} \left(\frac{\#(w, S' \upharpoonright (n+k_n))}{n+k_n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \\
\leq & \lim_{n\to\infty} \left(\frac{\#(w, S \upharpoonright n)}{n+k_n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \qquad \text{by (4.2)} \\
= & \lim_{n\to\infty} \left(\frac{\#(w, S \upharpoonright n)}{n-l+1} - \frac{\#(w, S \upharpoonright n)}{n-l+1} \right) \qquad \text{since } k_n = o(n) \\
= & \ 0.
\end{aligned}$$

Thus

$$\lim_{n\to\infty} \big(\mathrm{freq}(w, S' \upharpoonright (n+k_n)) - \mathrm{freq}(w, S \upharpoonright n)\big) = 0.$$

This establishes that $\mathrm{freq}(w, S \upharpoonright n)$ and $\mathrm{freq}(w, S' \upharpoonright (n+k_n))$ approach each other as $n \to \infty$, for all $w \in \Sigma^l$. Let $w' \in \Sigma_m^l - \Sigma^l$. Then $\mathrm{freq}(w', S \upharpoonright n) = 0$ for all n, since no m's appear in S. Since $\mathrm{freq}(m, S') = 0$,

$$\begin{aligned}
\mathrm{freq}(w', S') & \triangleq \lim_{n\to\infty} \frac{\#(w', S' \upharpoonright n)}{n} \\
& \leq \lim_{n\to\infty} \frac{l\#(m, S' \upharpoonright n)}{n} \\
& = l \cdot \mathrm{freq}(m, S') \\
& = 0,
\end{aligned}$$

where the inequality follows from the fact that for each m that appears in $S' \upharpoonright n$, at most l substrings of length l in $S' \upharpoonright n$ could have that m in them, and hence belong to $\Sigma_m^l - \Sigma^l$. By the non-negativity of freq, $\mathrm{freq}(w', S') = 0 = \mathrm{freq}(w', S)$, implying

$$\lim_{n\to\infty} \big(\mathrm{freq}(w', S \upharpoonright n) - \mathrm{freq}(w', S' \upharpoonright (n+k_n))\big) = 0$$

for all $w' \in \Sigma_m^l - \Sigma^l$. Hence,

$$\left(\forall w \in \Sigma_m^l\right) \lim_{n\to\infty} \big(\mathrm{freq}(w, S' \upharpoonright (n+k_n)) - \mathrm{freq}(w, S \upharpoonright n)\big) = 0. \tag{4.3}$$

Thus

$$\begin{aligned}
H_l(S') &\triangleq \frac{1}{l\log|\Sigma_m|}\liminf_{n\to\infty}\sum_{w\in\Sigma_m^l}\mathrm{freq}(w,S'\restriction n)\log\frac{1}{\mathrm{freq}(w,S'\restriction n)}\\
&= \frac{1}{l\log|\Sigma_m|}\liminf_{n\to\infty}\sum_{w\in\Sigma_m^l}\mathrm{freq}(w,S'\restriction (n+k_n))\log\frac{1}{\mathrm{freq}(w,S'\restriction (n+k_n))}\\
&= \frac{1}{l\log|\Sigma_m|}\liminf_{n\to\infty}\sum_{w\in\Sigma_m^l}\mathrm{freq}(w,S\restriction n)\log\frac{1}{\mathrm{freq}(w,S\restriction n)} \qquad \text{by (4.3)}\\
&\triangleq H_l(S).
\end{aligned}$$

Since this holds for all l, $H(S) = H(S')$. By Lemma 4.1, $\dim_{\mathrm{FS}}^{(\Sigma_m)}(S) = \dim_{\mathrm{FS}}^{(\Sigma_m)}(S')$. □

4.2 Bitstring Characters and Finite-State Dimension

In this section, we will interpret bitstrings of length l to be characters, the alphabet of the sequence will be a subset of $\{0,1\}^l - \{1^l\}$, and the marker "character" will be 1^l.

An infinite binary sequence $S \in \{0,1\}^\infty$ will then be simultaneously interpreted as an infinite sequence $S \in A^\infty$, where $A \subsetneq \{0,1\}^l$. In other words, every l bits of S will constitute 1 character from A. We interpret $\dim_{\mathrm{FS}}^{(\{0,1\})}(S)$ to be the finite-state dimension of S when S is viewed as an infinite binary sequence, and we interpret $\dim_{\mathrm{FS}}^{(A)}(S)$ to be the finite-state dimension of S when viewed as an infinite sequence over A.

Note that this interpretation of $\dim_{\mathrm{FS}}^{(A)}(S)$ is different from the meaning of $\dim_{\mathrm{FS}}^{(A)}(S)$ when $\{0,1\} \subseteq A$ (i.e., in the sense of Lemma 3.8). In the current case, the boundaries between characters actually change when moving from alphabet $\{0,1\}$ to alphabet A, in that a string of l bits is required to constitute one character of A. In the former case, for $\Sigma \subseteq \Sigma'$ and $S \in \Sigma^\infty$, $\dim_{\mathrm{FS}}^{(\Sigma')}(S)$ treats each character $a \in \Sigma$ in S as a character from Σ'. We rely on context to distinguish these two scenarios.

The following theorem establishes the relationship between the finite-state dimension of a binary sequence and its finite-state dimension when viewed as a sequence over $A \subseteq \{0,1\}^l$.

Theorem 4.4. *Let $l \in \mathbb{Z}^+$ and $\varnothing \neq A \subseteq \{0,1\}^l$. Then, for all $S \in A^\infty$,*

$$\dim_{\mathrm{FS}}^{(\{0,1\})}(S) = \frac{\log|A|}{l}\dim_{\mathrm{FS}}^{(A)}(S).$$

Proof. We first show that $\dim_{\mathrm{FS}}^{(\{0,1\})}(S) \geq \frac{\log|A|}{l}\dim_{\mathrm{FS}}^{(A)}(S)$. This holds trivially if $|A| = 1$, so assume $|A| \geq 2$. Let $s \in [0,\infty)\cap\mathbb{Q}$ such that $s > \dim_{\mathrm{FS}}^{(\{0,1\})}(S)$.

By our choice of s, there exists a finite-state gambler $G = (Q, \{0,1\}, \delta, \beta, q_0)$ such that G s-succeeds on S. Construct a finite-state gambler $G' = (Q', \Sigma', \delta', \beta', q_0')$ as follows.

- $Q' = Q$.
- $\Sigma' = A$.
- for all $q \in Q'$ and $w \in A$,

$$\delta'(q,w) = \delta(q,w).$$

- for all $q \in Q'$ and $w \in A$,

$$\beta'(q)(w) = \begin{cases} \frac{\widetilde{B}(q)(w)}{\widetilde{B}(q)(A)}, & \text{if } \widetilde{B}(q)(A) > 0 \\ 0, & \text{if } \widetilde{B}(q)(A) = 0 \end{cases},$$

  where

$$\widetilde{B}(q)(w) = \prod_{i=1}^{l} \beta(\delta(q, w_{i-1}))(w[i])$$

  and

$$\widetilde{B}(q)(A) = \sum_{w \in A} \widetilde{B}(q)(w).$$

- $q'_0 = q_0$.

Note that for all $q \in Q'$, $d_{G',q}$ is a martingale, and that $A \subseteq \{0,1\}^l$ is a prefix set. Let $q \in Q'$. Then

$$\begin{aligned} \widetilde{B}(q)(A) &\triangleq \sum_{w \in A} \widetilde{B}(q)(w) \\ &= \sum_{w \in A} \prod_{i=1}^{l} \beta(\delta(q, w_{i-1}))(w[i]) \\ &= \sum_{w \in A} d_{G',q}(w) \\ &\leq 1 \quad \text{by Corollary 3.4.} \end{aligned}$$

So

$$\widetilde{B}(q)(A) \leq 1 \tag{4.4}$$

for all $q \in Q'$.

Let $w \in A$ and let $q \in Q'$. Then

$$\begin{aligned} d_{G',q}(w) &= \frac{\widetilde{B}(q)(w)}{\widetilde{B}(q)(A)} \\ &\geq \widetilde{B}(q)(w) \qquad \text{by (4.4)} \\ &= \prod_{i=1}^{l} \beta(\delta(q, w_{i-1}))(w[i]) \\ &= d_{G,q}(w). \end{aligned}$$

So by induction, for all $z \in A^*$,

$$d_{G'}(z) \geq d_G(z).$$

Let $z \in A^*$ and $w \in A$, and let $q = \delta(z)$. Then

$$\begin{aligned} d_G(zw) &= 2^l \widetilde{B}(q)(w) d_G(z) \\ \Longrightarrow d_{G'}(z) &\geq \frac{1}{2^l \widetilde{B}(q)(w)} d_G(zw), \end{aligned}$$

so

$$\begin{aligned} d_{G'}(zw) &= |A|\frac{\widetilde{B}(q)(w)}{\widetilde{B}(q)(A)} d_{G'}(z) \\ &\geq \frac{|A|}{2^l \widetilde{B}(q)(A)} d_G(zw) \\ &\geq \frac{|A|}{2^l} d_G(zw) \qquad \text{by (4.4).} \end{aligned}$$

Then by induction, for all $n \in \mathbb{N}$ and $z \in \{0,1\}^{nl}$,

$$d_{G'}(z) \geq \left(\frac{|A|}{2^l}\right)^{\frac{|z|}{l}} d_G(z). \tag{4.5}$$

Let $t = \frac{sl}{\log|A|}$. Then

$$\begin{aligned} d_{G'}^{(t)}(z) &\triangleq |A|^{(t-1)\frac{|z|}{l}} d_{G'}(z) \\ &\geq |A|^{(t-1)\frac{|z|}{l}} \left(\frac{|A|}{2^l}\right)^{\frac{|z|}{l}} d_G(z) \quad \text{by (4.5)} \\ &\geq 2^{(s-1)|z|} d_G(z) \quad \text{since } |A| \geq 2 \\ &\triangleq d_G^{(s)}(z). \end{aligned}$$

Thus G' t-succeeds whenever G s-succeeds. This establishes that

$$\dim_{\mathrm{FS}}^{(\{0,1\})}(S) \geq \frac{s}{t} \dim_{\mathrm{FS}}^{(A)}(S) = \frac{\log|A|}{l} \dim_{\mathrm{FS}}^{(A)}(S).$$

We next show that $\dim_{\mathrm{FS}}^{(\{0,1\})}(S) \leq \frac{\log|A|}{l} \dim_{\mathrm{FS}}^{(A)}(S)$. Let $s \in \mathbb{Q}^+$ such that $s > \dim_{\mathrm{FS}}^{(A)}(S)$, and let $t = \frac{s\log|A|}{l}$. Then it suffices to show that $\dim_{\mathrm{FS}}^{(\{0,1\})}(S) \leq t$. By our choice of s, there exists a finite-state gambler $G = (Q, A, \delta, \beta, q_0)$ such that G s-succeeds on S.

Let $\mathrm{ppref}(A)$ be the set of all proper prefixes of the strings in A. Construct the finite-state gambler $G' = (Q', \Sigma', \delta', \beta', q_0')$ as follows.

- $Q' = Q \times \mathrm{ppref}(A)$.
- $\Sigma' = \{0,1\}$.
- for all $q \in Q'$, $w \in \mathrm{ppref}(A)$, and $b \in \{0,1\}$,

$$\delta'((q,w),b) = \begin{cases} (q, wb), & \text{if } wb \in \mathrm{ppref}(A) \\ (\delta(q,wb), \lambda), & \text{if } wb \in A \\ \perp, & \text{otherwise} \end{cases}.$$

- for all $q \in Q'$, $w \in \mathrm{ppref}(A)$, and $b \in \{0,1\}$,

$$\beta'(q,w)(b) = \begin{cases} \frac{\widetilde{B}(q,wb)}{\widetilde{B}(q,w)}, & \text{if } \widetilde{B}(q,w) > 0 \\ 0, & \text{if } \widetilde{B}(q,w) = 0 \end{cases},$$

where

$$\widetilde{B}(q,w) = \sum_{u \in A(w)} \beta(q)(wu)$$

and

$$A(w) = \{u \in \{0,1\}^* \mid wu \in A\}.$$

- $q'_0 = (q_0, \lambda)$.

In the non-degenerate case (where $\widetilde{B}(q,w) > 0$)

$$\beta'(q,w)(0) + \beta'(q,w)(1) = \frac{\widetilde{B}(q,w0) + \widetilde{B}(q,w1)}{\widetilde{B}(q,w)}.$$

For all $w \in \mathrm{ppref}(A)$, $A(w)$ is the disjoint union of $A(w0)$ and $A(w1)$. So $\widetilde{B}(q,w0)+\widetilde{B}(q,w1) = \widetilde{B}(q,w)$. Therefore $\beta'(q,w)(0) + \beta'(q,w)(1) = 1$.

Note that for all $q \in Q$, $\widetilde{B}(q,w) \leq 1$ for all $w \in \mathrm{ppref}(A) \cup A$. This follows from the fact that $w = \lambda$ maximizes $\widetilde{B}(q,w)$. $\widetilde{B}(q,\lambda) = \sum_{w \in A} \beta(q)(w) = 1$, by the constraint that $\beta(q)$ is a probability measure over A.

Intuitively, G''s martingale bets l times every l bits, in such a way that the l bets made will simulate the bet made once every l bits by G.

Let $z \in A^*$, $w \in A$, and $q = \delta(z)$. Then

$$\begin{aligned}
d_{G'}(zw) &= 2^l d_{G'}(z) \prod_{i=1}^{l} \beta'(q, w_{i-1})(w[i]) \\
&= 2^l d_{G'}(z) \prod_{i=1}^{l} \frac{\widetilde{B}(q,w_i)}{\widetilde{B}(q,w_{i-1})} \\
&= 2^l d_{G'}(z) \frac{\widetilde{B}(q,w)}{\widetilde{B}(q,\lambda)} \\
&\geq 2^l d_{G'}(z) \widetilde{B}(q,w) \\
&= 2^l d_{G'}(z) \beta(q)(w),
\end{aligned}$$

and

$$d_G(zw) = |A| \beta(q)(w) d_G(z).$$

So by induction

$$d_{G'}(z) \geq \prod_{i=1}^{\frac{|z|}{l}} 2^l \beta(\delta(z \upharpoonright il))(w),$$

and

$$d_G(z) = \prod_{i=1}^{\frac{|z|}{l}} |A| \beta(\delta(z \upharpoonright il))(w).$$

So

$$d_{G'}(z) \geq \left(\frac{2^l}{|A|}\right)^{\frac{|z|}{l}} d_G(z).$$

Thus

$$\begin{aligned}
d_{G'}^{(t)}(z) &\triangleq 2^{(t-1)|z|} d_{G'}(z) \\
&\geq 2^{(t-1)|z|} \left(\frac{2^l}{|A|}\right)^{\frac{|z|}{l}} d_G(z) \\
&= |A|^{(s-1)\frac{|z|}{l}} d_G(z) \quad \text{substituting } t = \frac{s \log |A|}{l} \\
&\triangleq d_G^{(s)}(z).
\end{aligned}$$

Therefore G' t-succeeds when G s-succeeds. This establishes that

$$\dim_{\mathrm{FS}}^{(\{0,1\})}(S) \leq \frac{t}{s} \dim_{\mathrm{FS}}^{(A)}(S) = \frac{\log |A|}{l} \dim_{\mathrm{FS}}^{(A)}(S).$$

□

4.3 Variations on the Champernowne Sequence

This section presents two variations on the Champernowne sequence [4] and shows them to be normal.

First we need the following lemma, which establishes that splicing two normal sequences together results in a normal sequence, as long as the splicing takes increasingly longer substrings from each sequence.

Lemma 4.5. *Let $S, T \in \Sigma^\infty$ be normal over the alphabet Σ. Let*

$$Z = S[0..i_1]T[0..i_1]S[i_1+1..i_2]T[i_1+1..i_2]S[i_2+1..i_3]T[i_2+1..i_3]\ldots$$

such that the function $f(j) = i_{j+1} - i_j$ is nondecreasing and unbounded. Then Z is normal over the alphabet Σ.

Proof. Let $n = i_j$, for some $j \in \mathbb{Z}^+$. Let $k_n = j$. Intuitively, k_n is the number of splices each taken from $S \upharpoonright n$ and $T \upharpoonright n$ to form $Z \upharpoonright 2n$. Since $i_{j+1} - i_j$ is nondecreasing and unbounded, $\lim_{n\to\infty} \frac{k_n}{n} = 0$.

Let $l \in \mathbb{Z}^+$, and let $w \in \Sigma^l$. Then $\mathrm{freq}(w, S) = \mathrm{freq}(w, T) = |\Sigma|^{-l}$. Because there are only k_n places in $S \upharpoonright n$ at which it was "broken" to be spliced into $T \upharpoonright n$, at most $k_n(l-1)$ instances of w in $S \upharpoonright n$ could have been disrupted by the splicing and hence not appear in $Z \upharpoonright 2n$. The same argument applies to instances of w in $T \upharpoonright_n$. Thus

$$\#(w, Z \upharpoonright 2n) \geq \#(w, S \upharpoonright n) + \#(w, T \upharpoonright n) - 2k_n(l-1)$$

Therefore

$$\begin{aligned}
\mathrm{freq}(w, Z) &\triangleq \lim_{n\to\infty} \frac{\#(w, Z \upharpoonright n)}{n} = \lim_{n\to\infty} \frac{\#(w, Z \upharpoonright 2n)}{2n} \\
&\geq \lim_{n\to\infty} \frac{\#(w, S \upharpoonright n) + \#(w, T \upharpoonright n) - 2k_n(l-1)}{2n} \\
&= \frac{1}{2} \lim_{n\to\infty} \frac{\#(w, S \upharpoonright n)}{n} + \frac{1}{2} \lim_{n\to\infty} \frac{\#(w, T \upharpoonright n)}{n} - (l-1) \lim_{n\to\infty} \frac{k_n}{n} \\
&= \frac{1}{2} \lim_{n\to\infty} \frac{\#(w, S \upharpoonright n)}{n} + \frac{1}{2} \lim_{n\to\infty} \frac{\#(w, T \upharpoonright n)}{n} \triangleq \frac{1}{2}\mathrm{freq}(w, S) + \frac{1}{2}\mathrm{freq}(w, T) \\
&= |\Sigma|^{-l}.
\end{aligned}$$

This holds for all $w \in \Sigma^l$. $\sum_{w\in\Sigma^l} \text{freq}(w, Z) = 1$, so, for all $w \in \Sigma^l$, $\text{freq}(w, Z) = |\Sigma|^{-l}$. Since this holds for all $l \in \mathbb{Z}^+$, Z is normal. □

We now construct a sequence with pushdown dimension at most half its finite-state dimension. Let $d \in (0,1) \cap \mathbb{Q}$, with $d = n/l$ for integers n and l. d will be the finite-state dimension of the sequence. Let $A \subseteq \{0,1\}^l - \{1^l\}$ such that $\log|A| = n$ and $|A| \geq 2$.

Let $\alpha_i \in A^*$ be the string consisting of all strings of length i over the alphabet A, concatenated in lexicographical ordering. Let $c = 1^l$, and let $A_c = A \cup \{c\}$. Define the sequences

$$\begin{aligned} S &= \alpha_1\overline{\alpha_1}\alpha_2\alpha_2\overline{\alpha_2\alpha_2}\alpha_3\alpha_3\alpha_3\overline{\alpha_3\alpha_3\alpha_3}\ldots \\ S' &= \alpha_1 c\overline{\alpha_1}\alpha_2\alpha_2 c\overline{\alpha_2\alpha_2}\alpha_3\alpha_3\alpha_3 c\overline{\alpha_3\alpha_3\alpha_3}\ldots \end{aligned}$$

Note that $|\alpha_i| = i|A|^i l \Rightarrow |\alpha_i^i| = i^2|A|^i l$. Champernowne [4] (see also [31]) showed that the sequence $Z = \alpha_1\alpha_2\alpha_2\alpha_3\alpha_3\alpha_3\ldots$ is normal over the alphabet A. The same technique easily gives the following.

Lemma 4.6 (Champernowne [4])**.** *Let*

$$T = \overline{\alpha_1}\ \overline{\alpha_2^2}\ \overline{\alpha_3^3}\ldots.$$

Then T is normal over alphabet Σ.

We combine these results to obtain that S is normal.

Lemma 4.7. *S is normal over the alphabet Σ.*

Proof. This follows immediately from Lemmas 4.6 and 4.5 and the normality of Z. □

Note, however, that S and S' are not normal over the alphabet $\{0,1\}$, because no more than $2l-2$ 1's appear consecutively in either sequence. They both have finite-state dimension equal to d, as established next.

Lemma 4.8. $\dim_{\text{FS}}^{(\{0,1\})}(S') = d$.

Proof. Recall that since S is normal, $\dim_{\text{FS}}^{(A)}(S) = 1$.

$$\begin{aligned}
\dim_{\text{FS}}^{(\{0,1\})}(S') &= \frac{\log(|A\cup\{c\}|)}{l}\dim_{\text{FS}}^{(A\cup\{c\})}(S') && \text{Theorem 4.4} \\
&= \frac{\log(|A\cup\{c\}|)}{l}\dim_{\text{FS}}^{(A\cup\{c\})}(S) && \text{Lemma 4.3} \\
&= \frac{\log(|A\cup\{c\}|)}{l}\frac{\log|A|}{\log(|A\cup\{c\}|)}\dim_{\text{FS}}^{(A)}(S) && \text{Lemma 3.8} \\
&= \frac{\log|A|}{l} && \text{Lemma 4.7 and Corollary 4.2} \\
&= d.
\end{aligned}$$

□

4.4 Pushdown Gambling on a Marked Sequence

We now show that the sequence S' presented in section 4.3 has pushdown dimension bounded above by half of its finite state dimension.

Lemma 4.9. $\dim_{\mathrm{PD}}^{(\{0,1\})}(S') \leq \frac{1}{2}d$.

Proof. Let $s > s' > d$. It suffices to show that $\dim_{\mathrm{PD}}^{(\{0,1\})}(S') \leq \frac{1}{2}s$.

We construct a pushdown gambler P that does the following. It reads the sequence $S' = \alpha_1 c \overline{\alpha_1} \alpha_2^2 c \overline{\alpha_2^2} \ldots$ in two alternating modes. The first mode involves reading the substring $\alpha_i^i c$, and the second mode involves reading the substring $\overline{\alpha_i^i}$. In the first mode, P bets optimally for any finite-state gambler, while the bits it reads are pushed onto the stack. Once c has been read, P pops c from the stack, and then uses the string it pushed, which is α_i^i, to bet optimally on the string that follows, which is $\overline{\alpha_i^i}$. It pops bits until the stack is empty, at which point α_{i+1}^{i+1} follows, and the gambler begins again.

As P is pushing bits onto its stack, it bets an equal amount ($d_P(a) = 2^l \frac{1-\epsilon}{|A|}$) on all bitstrings $a \in A$. It bets a small amount ($d_P(c) = 2^l \epsilon$) on the bitstring $c = 1^l$, and this bet can be made vanishingly smaller by shrinking ϵ, although some positive bet must be made so P's capital does not become 0 when it encounters c. The requirement that $\epsilon < 1 - 2^{l(s'-s)}$ ensures that P $\left(\frac{1}{2}s\right)$-succeeds on S', which is shown formally below. P bets nothing on any bitstring $a \notin A_c$.

Thus, P's bets in agreement with the optimal finite-state gambler for $\alpha_1 c \alpha_2 c \ldots$ when reading that subsequence of S', and it doubles its money on every bit when reading the subsequence $\overline{\alpha_1} \overline{\alpha_2^2} \ldots \overline{\alpha_i^i} \ldots$.

Formally, the pushdown gambler $P = (Q', \Sigma', \Gamma', \delta', \beta', q_0', z)$ is defined as follows on input $S \in \Sigma^\infty$.

```
P(S)
i ← 1                                  ▷ current bit of S
while TRUE                             ▷ each iteration k reads α_k^k c ‾α_k^k‾
      do repeat                        ▷ push bits until marker found
                  w ← λ
                  for j ← 1 to l       ▷ set w to next block of length l
                      do bet according to β(w) on S[i]
                         w ← wS[i]
                         push S[i] onto stack
                         i ← i + 1
            until w = 1^l
         pop l bits from stack
         while stack is not empty
               do bet all capital on bit on top of stack
                  read S[i]
                  i ← i + 1
                  pop 1 bit from stack
```

where

$$\begin{aligned}
\beta(w)(b) &= \begin{cases} \frac{\widetilde{B}(wb)}{\widetilde{B}(w)}, & \text{if } \widetilde{B}(w) > 0; \\ 0, & \text{otherwise.} \end{cases} \\
\widetilde{B}(w) &= \sum_{u \in A_c(w)} B(wu) \\
A_c(w) &= \{u \in \{0,1\}^* \mid wu \in A_c\} \\
B(a) &= \begin{cases} \frac{1-\epsilon}{|A|}, & \text{if } a \in A; \\ \epsilon, & \text{if } a = c; \end{cases} \\
0 < \epsilon &< 1 - 2^{l(s'-s)}.
\end{aligned}$$

Note that, for all $a \in A_c$,

$$\begin{aligned}
d_P(a) &= 2^l \prod_{i=1}^{l} \beta(a_{i-1})(a[i]) = 2^l \prod_{i=1}^{l} \left(\frac{\widetilde{B}(a_i)}{\widetilde{B}(a_{i-1})} \right) = 2^l \frac{\widetilde{B}(a)}{\widetilde{B}(\lambda)} d_P(a) \\
&= 2^l \frac{\sum\limits_{u \in A_c(a)} B(au)}{\sum\limits_{u \in A_c(\lambda)} B(\lambda u)} = 2^l \frac{B(a)}{\sum\limits_{u \in A_c} B(u)} = 2^l \frac{B(a)}{\epsilon + \sum\limits_{u \in A} \frac{1-\epsilon}{|A|}} = 2^l B(a).
\end{aligned}$$

Thus, for all $a \in A$,

$$d_P(a) = 2^l \frac{1-\epsilon}{|A|},$$

and, for $c = 1^l$

$$d_P(c) = 2^l \epsilon.$$

Recall that $d_P(c) = 2^l \epsilon$, and that P makes the same capital ($d_P(a) = 2^l \frac{1-\epsilon}{|A|}$) on each "character" $a \in A$. Since $|A|^k$ is the total number of strings, and k is the number of characters per string,

$$\begin{aligned}
d_P(\alpha_k^k) &= (d_P(\alpha_k))^k \\
&= \left(d_P(a)^{|A|^k} \cdot k \right)^k \\
&= 2^{k^2|A|^k l} \left(\frac{1-\epsilon}{|A|} \right)^{k^2|A|^k},
\end{aligned}$$

and

$$\begin{aligned}
d_P(\overline{\alpha_k^k}) &= 2^{|\overline{\alpha_i^i}|} \\
&= 2^{k^2|A|^k l}.
\end{aligned}$$

Thus,

$$\begin{aligned}
d_P(\alpha_k^k c \overline{\alpha_k^k}) &= \left(2^{k^2|A|^k l} \left(\frac{1-\epsilon}{|A|} \right)^{k^2|A|^k} \right) \left(2^l \epsilon \right) \left(2^{k^2|A|^k} \right) \\
&= \epsilon 2^{2k^2|A|^k l + l} \left(\frac{1-\epsilon}{|A|} \right)^{k^2|A|^k}.
\end{aligned}$$

Let $t = \frac{1}{2}s$. Then

$$\begin{aligned}
d_G^{(t)}(\alpha_k^k c \overline{\alpha_k^k}) &= 2^{(t-1)|\alpha_k^k c \overline{\alpha_k^k}|} \epsilon 2^{2k^2|A|^k l + l} \left(\frac{1-\epsilon}{|A|}\right)^{k^2|A|^k} \\
&= 2^{tl} \epsilon 2^{t2k^2|A|^k l} \left(\frac{1-\epsilon}{|A|}\right)^{k^2|A|^k} \\
&= 2^{tl} \epsilon 2^{t2k^2|A|^k l} \left(\frac{(1-\epsilon)^{\frac{1}{tl}}}{|A|^{\frac{1}{tl}}}\right)^{tlk^2|A|^k} \\
&= 2^{tl} \epsilon \left(2^2 \frac{(1-\epsilon)^{\frac{1}{tl}}}{(2^{dl})^{\frac{1}{tl}}}\right)^{tlk^2|A|^k} \\
&> 2^{tl} \epsilon \left(2^2 \frac{(1-\epsilon)^{\frac{1}{tl}}}{(2^{s'l})^{\frac{1}{tl}}}\right)^{tlk^2|A|^k} \\
&= 2^{tl} \epsilon \left(4^{1-\frac{s'}{s}} (1-\epsilon)^{\frac{2}{sl}}\right)^{tlk^2|A|^k} .
\end{aligned}$$

Recall that $\epsilon < 1 - 2^{l(s'-s)}$. Then $4^{1-\frac{s'}{s}}(1-\epsilon)^{\frac{2}{sl}} > 1$. Thus $d_G^{(t)}(\alpha_k^k c \overline{\alpha_k^k})$ grows without bound as $k \to \infty$, whence P t-succeeds on S'. Therefore

$$\dim_{\mathrm{PD}}^{(\{0,1\})}(S') \leq \frac{1}{2}s \implies \dim_{\mathrm{PD}}^{(\{0,1\})}(S') \leq \frac{1}{2}d.$$

□

Recall that $\dim_{\mathrm{FS}}^{(\{0,1\})}(S') = d$, where d was chosen to be an arbitrary element of $(0,1) \cap \mathbb{Q}$. The main theorem of the paper follows and establishes that the pushdown dimension of the sequence S' constructed in this way is bounded above by half of its finite-state dimension.

Theorem 4.10. *For every rational* $0 < d < 1$*, there exists a sequence* S' *with finite-state dimension* d *such that* $\dim_{\mathrm{PD}}(S') \leq \frac{1}{2}\dim_{\mathrm{FS}}(S')$.

Proof. This follows immediately from Lemmas 4.8 and 4.9. □

5 Conclusion

We have shown that there exist sequences with pushdown dimension strictly less than their finite-state dimension. This was done by the addition of special marker strings that are placed increasingly far apart in the sequence. Because these marker strings do not occur in other parts of the sequence, the sequence is not normal, and this prevents our proof from showing that any normal sequence has pushdown dimension less than 1. The marker strings are needed for our proof, but it is not known whether they are essential to bound the pushdown dimension. It is possible that the original sequence, without the markers, has the same pushdown dimension.

It is implicit in the paper of Merkle and Reimann [31], and made explicit in the Master's thesis of Nichols [36], that there is a normal sequence S such that a pushdown gambler can succeed on S, whereas the normality of S establishes that no finite-state gambler can succeed on S. However, the pushdown gambler fails to show that $\dim_{\mathrm{PD}}(S) < 1$, since the gambler makes money so slowly that it fails on S if any money is taken away at each step (i.e., if the "tax rate" s is less than 1).

Question 5.1. *Is there a normal sequence S such that* $\dim_{\mathrm{PD}}(S) < 1$*?*

We have shown that there exist sequences S such that $\dim_{\mathrm{PD}}(S) \leq \frac{1}{2}\dim_{\mathrm{FS}}(S)$. The factor $\frac{1}{2}$ seems artificial, and in our proof, it is an artifact of the particular pushdown gambler we designed. However, the factor $\frac{1}{2}$ may be fundamental to bounding the difference between finite-state gamblers and pushdown gamblers. A pushdown gambler must essentially "act like a finite-state gambler" when pushing characters onto its stack; its only advantage over finite-state gamblers comes from the ability to pop characters off the stack to remember information from long ago. Since the gambler cannot pop more characters than it pushes, it may be that a pushdown gambler can only gain a solid advantage over a finite-state gambler on half of the characters, which may explain why the separation achieved was only $\frac{1}{2}$. It is an open question whether this could be strengthened to show a larger separation between pushdown and finite-state dimension.

Question 5.2. *Is there a sequence S such that* $\dim_{\mathrm{PD}}(S) < \frac{1}{2}\dim_{\mathrm{FS}}(S)$*?*

Clearly, any pushdown gambler can be simulated by a Turing machine in linear time, whence $\dim_{\mathrm{p}}(S) \leq \dim_{\mathrm{PD}}(S)$ for all $S \in \Sigma^{\infty}$, where $\dim_{\mathrm{p}}(S)$ is the *polynomial-time dimension* of S, defined in [28].

The well-known LZ compression algorithm [42] translates easily into a martingale [10]. The LZ martingale doubles its money once for each bit compressed by the LZ compression algorithm. Hence LZ-dimension is easily defined in an analogous manner to finite-state and pushdown dimension. Like pushdown martingales, the LZ martingale is strictly more powerful than finite-state martingales [26], but is also computable in linear time. The relationship between pushdown dimension and LZ-dimension is open.

Finite-state dimension has many equivalent characterizations in terms of gamblers [5], compressors [5], decompressors [8, 38], entropy rates [3, 42], and log-loss predictors [18]. The results of [18] are easily modified to show such a characterization holds for log-loss pushdown predictors; it is open whether other such characterizations hold for pushdown dimension.

Finally, we note that our results are easily extended to $\mathrm{Dim}_{\mathrm{FS}}(S)$, finite-state *strong* dimension [2] of a sequence S. The finite-state strong dimension of a sequence is defined by replacing the limit superior in the definition of finite-state dimension with a limit inferior. It was shown in [2] that this definition exactly characterizes packing dimension [40, 41] when the gambler is not computationally bounded. Defining pushdown strong dimension $\mathrm{Dim}_{\mathrm{PD}}$ similarly, the techniques of the present paper show that, for every rational $0 < d < 1$, there is a sequence S such that $\mathrm{Dim}_{\mathrm{FS}}(S) = d$ and $\mathrm{Dim}_{\mathrm{PD}}(S) \leq \frac{1}{2}\mathrm{Dim}_{\mathrm{FS}}(S)$.

6 Acknowledgements

The authors thank Jim Lathrop, Jack Lutz, and Giora Slutzki for their help and for useful discussions.

References

[1] K. Ambos-Spies, W. Merkle, J. Reimann, and F. Stephan *Hausdorff dimension in exponential time.* Proceedings of the 16th IEEE Conference on Computational Complexity, CCC'2001 (Chicago, Illinois, June 18-21, 2001), pp. 210–217.

[2] K. B. Athreya, J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. *Effective strong dimension, algorithmic information, and computational complexity.* SIAM Journal on Computing, to appear. Preliminary version appeared in *V. Diekert and M. Habib (eds.), Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science*, Springer Lecture Notes in Computer Science, Montpellier, France, March 25-27, 2004, pp. 632–643.

[3] C. Bourke, J. M. Hitchcock, and N. V. Vinodchandran. *Entropy rates and finite-state dimension.* Theoretical Computer Science, 349(3):392-406, 2005.

[4] D. G. Champernowne, *The Construction of Decimals Normal in the Scale of Ten.* J. London Math. Soc., 2(8):254–260, 1933.

[5] J. J. Dai, J. I. Lathrop, J. H. Lutz, and E. Mayordomo. *Finite-state dimension.* Theoretical Computer Science, 310(1-3):1-33, 2004.

[6] D. Doty. *Every sequence is decompressible from a random one.* In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker (eds.): Logical Approaches to Computational Barriers, Proceedings of the Second Conference on Computability in Europe (CiE 2006), Swansea, UK, July 2006.

[7] D. Doty. *Dimension extractors and optimal decompression.* Theory of Computing Systems, to appear. Special issue of invited papers from Computability in Europe 2006.

[8] D. Doty and P. Moser. Finite-state dimension and lossy decompressors. Technical Report cs.CC/0609096, Computing Research Repository, 2006.

[9] K. Falconer. *The Geometry of Fractal Sets.* Cambridge University Press, 1985.

[10] M. Feder. *Gambling using a finite-state machine*, IEEE Transcations on Information Theory, 37:1459-1461, 1991.

[11] L. Fortnow and J. H. Lutz. *Prediction and dimension.* Journal of Computer and System Sciences, 70(4):570-589, 2005.

[12] L. Fortnow, J. M. Hitchcock, A. Pavan, N. V. Vinodchandran, and F. Wang. *Extracting Kolmogorov complexity with applications to dimension zero-one laws.* Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (Venice, Italy, July 10-14, 2006), Lecture Notes in Computer Science 4051, pages 335-345. Springer-Verlag, 2006.

[13] C. Glaßer, M. Ogihara, A. Pavan, A. Selman, and L. Zhang. *Autoreducibility, mitoticity, and immunity.* In Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science, pages 387-398. Springer-Verlag, 2005.

[14] X. Gu. *A note on dimensions of polynomial size circuits.* Theoretical Computer Science, 359(1-3):176-187, 2006.

[15] X. Gu and J. H. Lutz. *Dimension characterizations of complexity classes.* Computational Complexity, to appear.

[16] F. Hausdorff. *Dimension und äußeres Maß.* Math. Ann. 79:157-179, 1919.

[17] J. M. Hitchcock. Effective Fractal Dimension: Foundations and Applications. Ph.D. Dissertation, Iowa State University, 2003.

[18] J. M. Hitchcock. *Fractal dimension and logarithmic loss unpredictability.* Theoretical Computer Science, 304(1–3):431–441, 2003.

[19] J. M. Hitchcock. *MAX3SAT is exponentially hard to approximate if* NP *has positive dimension.* Theoretical Computer Science, 289(1):861-869, 2002.

[20] J. M. Hitchcock. *Online learning and resource-bounded dimension: Winnow yields new lower bounds for hard sets.* SIAM Journal on Computing, 36(6):1696-1708, 2007.

[21] J. M. Hitchcock and A. Pavan. *Hardness hypotheses, derandomization, and circuit complexity.* In Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science, pages 336-347. Springer-Verlag, 2004.

[22] J. M. Hitchcock, A. Pavan, and N. V. Vinodchandran. *Partial bi-immunity, scaled dimension, and NP-completeness.* Theory of Computing Systems, to appear.

[23] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. *Scaled dimension and nonuniform complexity.* Journal of Computer and System Sciences, 69(2):97-122, 2004.

[24] J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, New York, N.Y., 1979.

[25] W. Kuich, A. Salomaa, Semirings, Automata, Languages, EATCS Monographs On Theoretical Computer Science; Vol. 5, 1985.

[26] J. I. Lathrop and M. Strauss. *A universal upper bound on the performance of the Lempel-Ziv algorithm on maliciously-constructed data*, Proceedings of the Compression and Complexity of Sequences, IEEE, Salerno, Italy, June 11-13, 1997, pp. 123-135.

[27] M. López-Valdés and E. Mayordomo. *Dimension is compression.* Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS'05). Springer-Verlag Lecture Notes in Computer Science 3618, 676-685 (2005).

[28] J. H. Lutz. *Dimension in complexity classes.* SIAM Journal on Computing, 32(5):1236-1259, 2003.

[29] J. H. Lutz. *The dimensions of individual strings and sequences.* Information and Computation, 187(1):49-79, 2003.

[30] E. Mayordomo. *A Kolmogorov complexity characterization of constructive Hausdorff dimension.* Information Processing Letters, 84(1):1–3, 2002.

[31] W. Merkle and J. Reimann. *On selection functions that do not preserve normality.* Theory of Computing Systems, 39(5):685-697, 2006.

[32] P. Moser. BPP *has effective dimension at most 1/2 unless* BPP = EXP. Technical Report TR03-029, Electronic Colloquium on Computational Complexity, 2003.

[33] P. Moser. *Derandomization and Quantitative Complexity.* PhD thesis, Université de Genève, 2004.

[34] P. Moser. *Generic density and small span theorem.* In Proceedings of the 15th International Symposium on Fundamentals of Computation Theory, pages 92-102. Springer-Verlag, 2005.

[35] P. Moser. *Martingale families and dimension in* P. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker (eds.): Logical Approaches to Computational Barriers, Proceedings of the Second Conference on Computability in Europe (CiE 2006), Swansea, UK, July 2006.

[36] J. Nichols, *Pushdown gamblers and pushdown dimension*, Master's thesis, Iowa State University, 2004.

[37] C. Schnorr, H. Stimm. *Endliche Automaten und Zufallsfolgen*, Acta Informatica 1:345–359, 1972.

[38] D. Sheinwald, A. Lempel, and J. Ziv. On encoding and decoding with two-way head machines. *Information and Computation*, 116(1):128–133, 1995.

[39] L. Staiger, *Weighted finite automata and metrics in Cantor space*, Journal of Automata, Languages and Combinatorics, 8(2):353-360, 2003.

[40] D. Sullivan. *Entropy, Hausdorff measures old and new, and limit sets of geometrically finite Kleinian groups.* Acta Mathematica, 153:259–277, 1984.

[41] C. Tricot. *Two definitions of fractional dimension.* Mathematical Proceedings of the Cambridge Philosophical Society, 91:57–74, 1982.

[42] J. Ziv and A. Lempel. *Compression of individual sequences via variable-rate coding.* IEEE Transactions on Information Theory, 24(5):530–536, 1978.