FINITE-STATE DIMENSION OF INDIVIDUAL SEQUENCES

by

Christopher M. Bourke

A THESIS

Presented to the Faculty of
The Graduate College at the University of Nebraska
In Partial Fulfillment of Requirements
For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Vinodchandran Variyam

Lincoln, Nebraska

May, 2004

FINITE-STATE DIMENSION OF INDIVIDUAL SEQUENCES

Christopher M. Bourke, M.S.

University of Nebraska, 2004

Advisor: Vinodchandran Variyam

Abstract

Classical Hausdorff dimension, popularly known as fractal dimension, has recently been effectivized by *gales*–functions that are essentially betting strategies that play against infinite binary sequences. Gales are a generalization of martingales and are sufficient for establishing fractal dimension on sets. When gales are restricted to functions computable in a certain complexity class, such a characterization endows sequences (or sets of sequences) with dimension within the complexity class. Countable sets and singletons, that would otherwise have Hausdorff dimension zero, may be given quantifiable positive dimension. In this thesis we restrict our examination to gale functions computable by *finite-state machines* and explore *individual* sequences within the topology of the Cantor space. We develop new concepts of periodicity, entropy and *betting trees* in terms of fixed "blocks" of a sequence. We use these to establish that the entropy rate with respect to blocks is an upper bound to the dimension of any sequence. We also extend these results to the usual Information Theoretic concept of entropy, yielding many implications for well studied *automatic sequences* such as characteristic sequences of regular languages.

ACKNOWLEDGEMENTS

First and foremost I thank Dr. Vinodchandran Variyam, my advisor for inspiring me to pursue studies in theoretical computer science.

I would also like to thank Dr. Jack Lutz for his talks and discussions at the Atlantic Theory Seminar and Dr. John Hitchcock for his talks and discussions during his time at UNL which made this thesis possible.

I thank Dr. Pascal Weil for his many helpful talks and directions.

I thank Dr. Jeffrey Shallit, Dr. Jean-Paul Allouche for their wonderful book and helpful correspondence and Dr. Jean Berstel for his correspondence.

I thank Dr. Jitender Deogun and Dr. Stephen Scott for the great input and serving on my committee.

Finally, I thank the love of my life, Lauren Gubbels for inspiration, strength and determination.

Contents

Chapter 1

Introduction

Classical Hausdorff dimension was originally developed by considering geometric coverings by spheres of diminishing radii on sets in topological spaces. In a sense, this concept can be viewed as an (unrestricted) function that allows one to compare relative sizes of sets.

Now consider functions computable within classes in the complexity hierarchy; polynomial time, exponential space, etc. and even classes in the arithmetic hierarchy. If we restrict our consideration to functions in a given complexity class, we endow sets dimension *within* the complexity class.

This model has the advantage of being able to quantify dimension of sets that would otherwise have dimension zero, an inherent limitation in classical Hausdorff dimension. Specifically, countable sets, finite sets and singleton sets all have Hausdorff dimension zero but may yield positive dimension with respect to certain complexity classes.

This effectivization of classical Hausdorff Dimension was developed by Lutz [10] in terms of *gales*. Gales are generalizations of martingales–betting strategies for a sequence of independent random coin tosses. Lutz also proved that this gale characterization and classical Hausdorff dimension are one in the same.

When the resources required to compute gale functions are bounded within complexity classes, *languages* within these classes are assigned Resource Bounded Dimension. Gales can also be effectivized relative to the arithmetic hierarchy endowing them with Constructive Dimension.

As we travel down the complexity hierarchy, we increasingly restrict the resources allowed to compute gales. At each step dimension can increase (or remain the same) relative to the complexity class. At the lowest level of the hierarchy are gales that can be computed by finite-state machines (automata). It is on this concept of *Finite-State Dimension* that we focus our attention.

We will work in the Cantor Space, a topology that includes all (right) infinite binary sequences. Our main model of computing gales will be *Finite-State Gamblers*,

essentially finite-state automatons that bet on successive bits in a sequence. In this thesis, our emphasis will be on individual sequences rather than sets of sequences. We may may consider sequences in and of themselves or as characteristic sequences of languages using the usual enumeration of finite strings.

Chapter 2 is devoted to presenting notation, definitions and results. In particular, we first establish notation and some tools that we will use throughout the rest of this thesis. For the most part, we have preferred to make use of notation common in the area of *stringology* in [1] and [9]. We then present Lutz's concepts and definitions effectivizing Hausdorff dimension. In the next subsection we define Finite-State Gamblers and cast gales in terms of Finite-State Dimension. We then present several motivating examples to clarify these concepts. The final section presents some interesting results established by the originators [5] of Finite-State Dimension.

In Chapter 3 we present our contributions and results. We start by developing general gamblers for individual sequences called *betting trees*. In the next section we offer a new concept of sequence periodicity in terms of fixed blocks of a sequence. In addition, we present our main result—entropy rates (with respect to blocks) provide an upper bound on a sequence's dimension. We then extend this result to the usual stringology notion of a sequence's entropy in terms of factor sets. The fourth section presents implications of our main result on the well studied class of sequences called *Automatic Sequences*. Of particular interest, we show that the finite-state dimension of the characteristic sequence of a regular language is zero. We conclude the chapter by analyzing the sequence hierarchy with respect to sequence complexity and dimension.

In the final chapter we pose several questions for further study. We ask if there are techniques to show non-trivial lower bounds on the dimension of individual sequences. We also ask if there are sequences of complexity that exist between our block characterization and the established notion of factor sets. We also discuss lower bounds on the number of states gamblers must have to succeed on various classes of sequences. Finally we describe possible extensions of our results to *scaled* finite-state dimension.

Chapter 2

Preliminaries

As is standard practice, $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote the sets of non-negative integers, integers, rational and real numbers respectively. We work over the signature (alphabet) $\Sigma = \{0, 1\}$ and $b \in \{0, 1\}$ will be referred to as a *bit* (letter) unless otherwise noted. Consequently all logarithms will be implicitly base-2 unless explicitly denoted. We denote the set of all finite *strings* Σ^*, with λ as the empty string. A *language* is a set $L \subseteq \Sigma^*$. For strings $w, u \in \{0, 1\}^*$, we denote the concatenation of u and w as uw. The string w is a *factor* of u if there exist $x, y \in \{0, 1\}^*$ such that $u = xwy$. The string w is a *prefix* of u and we write $w \sqsubseteq u$ if $x = \lambda$; w is a *proper prefix* of u and we write $w \sqsubset u$ if $y \neq \lambda$. The set of all factors of a string $w \in \{0, 1\}^*$ is denoted $F(w)$ while the set of all factors of length n is denoted $F_n(w)$.

In addition to finite strings, we define *infinite sequences* over Σ and denote the set of all such sequences as Σ^ω. Such a set constitutes a topology and is referred to as the Cantor Space. For simplicity we will denote it as $\mathbf{C}$. For sequences $S \in \mathbf{C}$ we denote a subsequence of S as $S[i \dots j]$ for the i-th through j-th bits with the convention that $S[0]$ is the first bit in the sequence (thus we only consider *right infinite* sequences). Factor sets for infinite sequences are defined as they are for finite strings, that is $F(S)$ and $F_n(S)$ are sets of finite strings that are factors (appear in) the sequence S of any length and length n respectively. A finite string $x \in \{0, 1\}^*$ is a prefix of a sequence if $x = S[0 \dots |x| - 1]$.

2.1 Definitions

Classical Hausdorff dimension provides a quantification of the complexity of topological subspaces by geometric coverings. Lutz [10] has recently effectivized classical Hausdorff dimension via the use of gale functions that define betting strategies on subsets of the Cantor Space.

In this section we first establish Lutz's characterization of classical Hausdorff dimension in terms of gale functions. We then concentrate on gale functions that are computable by finite-state machines.

2.1.1 Gales & Hausdorff Dimension

We begin by reviewing Lutz's gale characterization. All of the definitions in this subsection are from [10].

Definition 2.1. Let $s \in [0, \infty)$, an *s-gale* is a function $d : \{0,1\}^* \to [0, \infty)$ that satisfies the condition,

$$d(w) = 2^{-s}[d(w0) + d(w1)]$$

for all $w \in \{0,1\}^*$. A *martingale* is a 1-gale.

As in any game of chance, a good gambler is one that can win a lot (an infinite amount) of money. A gambler's ability to win or lose on certain sets or individual sequences can be quantified, in a sense, by putting it at a disadvantage and seeing if it can still win. That is exactly what an s-gale does. If $s = 1$ then a martingale ensures that the bets are fair. If $s > 1$ then the gale function guarantees an additional payoff over the capital gained by the bet itself. However, if $s < 1$ then the bets are not fair–some amount of capital is lost on each bet regardless of the wager's outcome. As $s \to 0$ our betting strategy will have to be increasingly more savvy (that is, be able to win *a lot* more money) in order to offset the penalty that we incur from s.

Definition 2.2. Let d be an s-gale with $s \in [0, \infty)$. We say that d *succeeds* on a sequence $S \in \mathbf{C}$ if

$$\limsup_{n \to \infty} d(S[0 \ldots n-1]) = \infty.$$

That is, as we progress in an infinite sequence our betting strategy can win an infinite amount of money infinitely often. We also define the *success set* of d as

$$S^\infty[d] = \Big\{ S \in \mathbf{C} \mid d \text{ succeeds on } S \Big\}.$$

For a *fixed* s-gale d, the success set includes all the sequences for which d can win an infinite amount of money on.

Definition 2.3. Let $X \subseteq \mathbf{C}$, the *gale set* is the set

$$\mathcal{G}(X) = \Big\{ s \in [0, \infty) \Big| \text{ there is an } s\text{-gale } d \text{ for which } X \subseteq S^\infty[d] \Big\}.$$

Finally, it has been shown that the classical Hausdorff dimension is equivalent to the greatest lower bound of the gale set.

Definition 2.4. The *Hausdorff Dimension* of a set $X \subseteq \mathbf{C}$ is

$$\dim_{\mathrm{H}}(X) = \inf \mathcal{G}(X).$$

2.1.2 Finite-State Gamblers & Dimension

The gale characterization of Hausdorff dimension allows us to consider any computable function. When we consider finite-state dimension, we restrict gale func-

tions to those computable by finite state devices such as finite-state automata. Thus, our primary model of quantifying sequence complexity will be *finite-state gamblers* (FSGs). A finite-state gambler is much like a finite-state automaton but rather than *accepting* or *rejecting* a finite string, it places bets on subsequent bits encountered in an infinite sequence and makes state transitions appropriately.

By convention, we restrict possible bets to rational numbers, $\mathbf{B} = \mathbb{Q} \cap [0, 1]$. All of the proceeding definitions are from [5].

Definition 2.5. A *finite-state gambler*[1] is a 5-tuple $G = (Q, \delta, \beta, q_0, c_0)$ where

- Q is a nonempty, finite set of *states*
- $\delta : Q \times \{0, 1\} \to Q$ is the *transition function*
- $\beta : Q \to \mathbf{B}$ is the *betting function*
- $q_0 \in Q$ is the *initial state*
- c_0 is the *initial capital*.

As with finite-state automata, the transition function can be extended to δ^*,

$$\delta^* : Q \times \{0, 1\}^* \to Q$$

by the recursive definition

$$\begin{array}{rcl} \delta^*(q, \lambda) & = & q, \\ \delta^*(q, wb) & = & \delta(\delta^*(q, w), b) \end{array}$$

for all $q \in Q$, $w \in \{0, 1\}^*$, and $b \in \{0, 1\}$. We write δ for δ^* and use the abbreviation $\delta(w) = \delta(q_0, w)$. We can now define martingale functions that are computed by finite-state gamblers.

Definition 2.6. Let G be a finite-state gambler. The *martingale* of G is the function

$$d_G : \{0, 1\}^* \to [0, \infty)$$

defined by the recursion

$$\begin{array}{rcl} d_G(\lambda) & = & c_0 \\ d_G(wb) & = & 2d_G(w)\big[(1 - b)(1 - \beta(\delta(w))) + b\beta(\delta(w))\big] \end{array}$$

for $w \in \{0, 1\}^*$ and $b \in \{0, 1\}$.

If G is in state $q \in Q$ it places a proportion of its current capital c, $\beta(q)c$ that the next bit will be $b = 1$ and $(1-\beta(q))c$ that the next bit in the sequence will be $b = 0$. Thus, a finite-state gambler is always forced to place all of its money at some risk. After such a bet, G will be in state $\delta(q, b)$ with expected capital

[1] Gamblers were originally defined to have k accounts to work with. However, for finite-state dimension, it is sufficient to consider single account FSGs [5].

$$2c\Big[(1-b)(1-\beta(q)) + b\beta(q)\Big] = \begin{cases} 2\beta(q)c & \text{if } b = 1 \\ 2(1-\beta(q))c & \text{if } b = 0. \end{cases}$$

To illustrate, we present the following example.

Example 2.7. Figure 2.1 represents a 4 state finite-state gambler $G = (Q, \delta, \beta, q_0, 1)$ where $Q = \{q_0, q_1, q_2, q_3\}$. Labels inside each state correspond to the betting function $\beta(q)$. We point out that $\delta(\lambda) = q_0$, $\delta(1) = q_3$, and $\delta(101) = q_1$. Observe that the martingale function is $d_G(\lambda) = 1$, $d_G(0) = \frac{1}{2}$, $d_G(00) = \frac{2}{3}$ and $d_G(001) = \frac{1}{3}$.

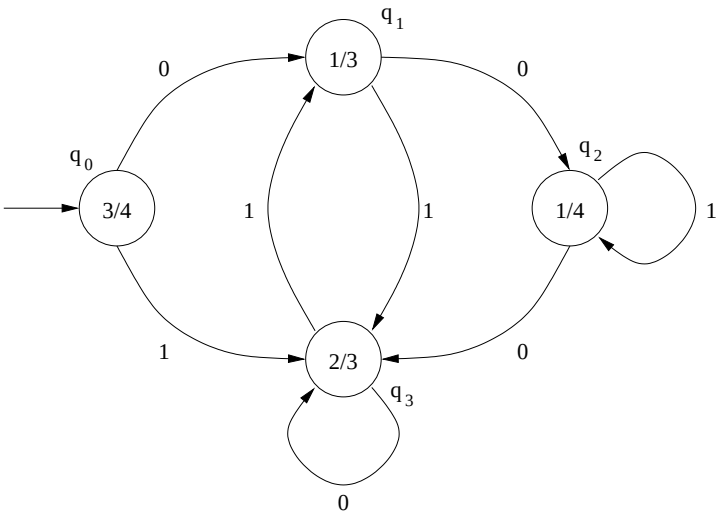

Figure 2.1: Example Finite-State Gambler G

A finite-state gambler also defines an s-gale function for any $s \in [0, \infty)$ in a very natural way.

Definition 2.8. Let $s \in [0, \infty)$. The *s-gale* of a finite-state gambler G is a function

$$d_G^{(s)} : \{0,1\}^* \to [0, \infty)$$

defined by

$$d_G^{(s)}(w) = 2^{(s-1)|w|} d_G(w)$$

for all $w \in \{0,1\}^*$.

As with general s-gales, a good gambler must overcome the handicap incurred from s and be able to win an infinite amount of money in order to *succeed*.

Definition 2.9. For $s \in [0, \infty)$, a *finite-state s-gale* is an s-gale d for which there exists a finite-state gambler G such that $d_G^{(s)} = d$. A *finite-state martingale* is a finite-state 1-gale.

Definition 2.10. Let $X \subseteq \mathbf{C}$, then the *finite-state gale set* is the set

$$\mathcal{G}_{\mathrm{FS}}(X) = \Big\{ s \in [0,\infty) \mid \text{ there is a finite-state } s\text{-gale } d \text{ for which } X \subseteq S^{\infty}[d] \Big\}.$$

Recall that the Hausdorff dimension of a set $X \subseteq \mathbf{C}$ is defined as $\dim_{\mathrm{H}}(X) = \inf \mathcal{G}(X)$, so it is natural to define finite-state dimension similarly.

Definition 2.11. Let $X \subseteq \mathbf{C}$ then the *finite-state dimension* of X is

$$\dim_{\mathrm{FS}}(X) = \inf \mathcal{G}_{\mathrm{FS}}(X).$$

That is, the finite-state dimension is the greatest lower bound of all $s \in [0,\infty)$ such that there exists a finite-state gambler that succeeds on all sequences in X for all $s' \geq s$. We additionally define the finite-state dimension of an individual sequence $S \in \mathbf{C}$ as the finite-state dimension of the singleton set,

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

Fact 2.12. [5] Let $X, Y \subseteq \mathbf{C}$.

1. $0 \leq \dim_{\mathrm{H}}(X) \leq \dim_{\mathrm{FS}}(X) \leq 1$.
2. If $X \subseteq Y$, then $\dim_{\mathrm{FS}}(X) \leq \dim_{\mathrm{FS}}(Y)$.

2.2 Examples

We now present several motivating examples that will clarify these concepts. One of the simplest ways of defining binary sequences is to consider a language's characteristic sequence.

Definition 2.13. [15] Let $\{0,1\}^* = \{s_0, s_1, s_2, \ldots\}$ such that each s_i is in lexicographical order. The *characteristic sequence* of a language $L \subseteq \{0,1\}^*$ is a sequence $\chi_L \in \mathbf{C}$ such that

$$\chi_L[i] = \begin{cases} 1 & \text{if } s_i \in L \\ 0 & \text{if } s_i \notin L. \end{cases}$$

2.2.1 All Strings

Example 2.14. As a primer, let us consider the regular language $\mathrm{ALL} = \Sigma^*$ the set of all (finite) strings. The characteristic sequence is simple enough,

$$\chi_{\mathrm{ALL}} = 1^{\infty}.$$

We will show that

$$\dim_{\mathrm{FS}}(\chi_{\mathrm{ALL}}) = 0.$$

Proof. An obvious finite-state gambler G would have a single state that always bets everything on 1 ($\beta = 1$) then loops back into itself. Obviously the only sequence G succeeds on is χ_{ALL} since, if a zero is ever encountered in the sequence, all money is lost and the game is over. Let $w \sqsubset \chi_{\text{ALL}}$, clearly then,

$$d_G(w) = 2^{|w|}.$$

It follows that the corresponding finite-state s-gale is

$$\begin{aligned} d_G^{(s)}(w) &= 2^{(s-1)|w|} d_G(w) \\ &= 2^{(s-1)|w|} 2^{|w|} \\ &= 2^{s|w|}. \end{aligned}$$

Since for any $s \in [0, \infty)$, and s' such that $s' > s \geq 0$,

$$\lim_{|w| \to \infty} 2^{s'|w|} = \infty,$$

$d_G^{(s')}$ succeeds on χ_{ALL}. Therefore $\chi_{\text{ALL}} \in S^{\infty}[d_G^{(s')}]$. It then follows that

$$\dim_{\text{FS}}(\chi_{\text{ALL}}) = \inf \mathcal{G}_{\text{FS}}(\chi_{\text{ALL}}) = 0.$$

□

2.2.2 Parity Language

Our next example is a bit more complicated and has interesting parallels with the Thue-Morse sequences that we will see later on.

Example 2.15. Let P be the regular language containing all strings with even parity,

$$P = \left\{ x \in \{0,1\}^* \mid h(x) = 2i, i \geq 0 \right\}.$$

Note that the empty string, λ, and strings with *no* 1s in it are in the language. Again, we will show that

$$\dim_{\text{FS}}(\chi_P) = 0.$$

Proof. We first observe that the characteristic sequence of P is

$$\chi_P = 1\,10\,1001\,10010110\,1001011001101001\,1001011001101001011010011001011 0\ldots$$

A pattern in this characteristic sequence is immediately apparent. Each "block" (i.e. each subsequent section of length 2^k, $k \geq 0$ representing strings of equal length) is a repetition of the previous block concatenated with the complement (each bit flipped) of the previous block. This means that after $\chi_P[0 \ldots 2]$ every series of four bits is either 1001 or 0110. This suggests the finite-state gambler as shown in Figure 2.2.

For any $w \in \{0,1\}^*$ such that $w \sqsubset \chi_P$, on any block of four bits we will double our money on three bets and remain at our current capital on the fourth one. Therefore,

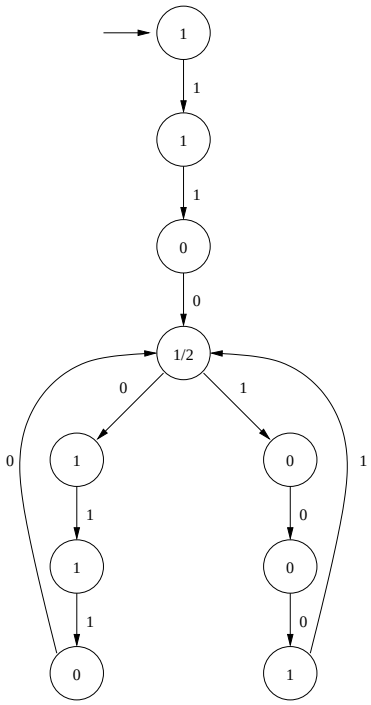

Figure 2.2: Finite-State Gambler for P

if $c_0 = 1$, our expected capital after $|w|$ bits is

$$2^{|w|-\lfloor\frac{|w|}{4}\rfloor}.$$

We observe that $\chi_P \in S^\infty[d_G]$. The corresponding finite-state s-gale is

$$\begin{aligned} d_G^{(s)}(w) &= 2^{(s-1)|w|} d_G(w) \\ &= 2^{(s-1)|w|} 2^{|w|-\lfloor\frac{|w|}{4}\rfloor} \\ &\geq 2^{|w|(s-\frac{1}{4})}. \end{aligned}$$

For any $s > \frac{1}{4}$,

$$\lim_{|w|\to\infty} 2^{|w|(s-\frac{1}{4})} = \infty.$$

But for any $0 \leq s \leq \frac{1}{4}$ we are not so lucky; $d_G^{(s)}(w)$ over time *loses*. The gambler in Figure 2.2 is, however, sufficient to prove that

$$\dim_{\mathrm{FS}}(\chi_P) = \inf \mathcal{G}_{\mathrm{FS}}(\chi_P) \leq \frac{1}{4}.$$

We can easily generalize the construction of G to accommodate any $s \in (0, \infty)$ as follows. We fix $\alpha \in \mathbb{Z}, \alpha \geq 1$ and proceed to construct a finite-state gambler G_α such that G_α makes definite bets on the first $2^\alpha - 1$ bits of χ_P directly using a path of $2^\alpha - 1$ states. The last state in the path leads into two separate loops each of

length 2^α. The first loop bets directly on the subsequence $\chi_P[2^\alpha - 1 \ldots 2^{\alpha+1} - 2]$ while the second bets directly on the compliment of this subsequence. Note that the initial construction in Figure 2.2 is G_2. The total number of states is thus $3(2^\alpha - 1)$. The corresponding martingale for this general construction is

$$d_{G_\alpha}(w) = 2^{|w| - \lfloor \frac{|w|}{2^\alpha} \rfloor},$$

which gives us a finite-state s-gale in terms of α as

$$\begin{aligned} d^{(s)}_{G_\alpha}(w) &= 2^{(s-1)|w|} d_G(w) \\ &= 2^{|w|(s - 2^{-\alpha})}. \end{aligned}$$

Therefore, for any $s > 0$, we can select a sufficiently large α and build a finite-state gambler that will succeed for all $s' > s$. Specifically we need only select $\alpha \in \mathbb{Z}^+$ such that $\alpha > -\log s$. To be complete, we note that $\chi_P \in S^\infty[d_{G_\alpha}]$ for any $\alpha \geq 1$ and $\chi_P \in S^\infty[d^{(s)}_{G_\alpha}]$ for any $\alpha > -\log s$ and $s' > s$. In addition, by the general construction for G_α, it is clear $\mathcal{G}_{\mathrm{FS}} = \{(0, \infty)\}$ so we conclude that

$$\dim_{\mathrm{FS}}(\chi_P) = \inf \mathcal{G}_{\mathrm{FS}}(\chi_P) = 0.$$

□

2.2.3 Even Length Strings

Our next example examines the regular language that contains all strings of even length (as well as λ).

Example 2.16. Let $E = (\Sigma^2)^*$, that is,

$$E = \big\{ x \in \{0,1\}^* \mid |x| = 2i, \forall i \geq 0 \big\},$$

then

$$\dim_{\mathrm{FS}}(\chi_E) = 0.$$

Proof. As with the previous examples, the characteristic sequence is highly structured.

$$\chi_E = 1^1 0^2 1^4 0^8 1^{16} 0^{32} 1^{64} 0^{128} 1^{256} \ldots$$

Another obvious pattern arises for this characteristic sequence; it is made up of alternating series of 1s and 0s such that each block grows in size by a factor of 2. Clearly we cannot build a finite-state gambler that adapts to the growing size of each sequence since Q, the set of states, is finite. We can, however, *fix* a block size that we want to bet on, say 2^α for some $\alpha \geq 1$. For a given α, we build a finite-state gambler, G_α as follows. Define a path of length $2^\alpha - 1$ that makes definite bets on the first $2^\alpha - 1$ bits of χ_E, thus for $w \sqsubset \chi_E$, $|w| = 2^\alpha - 1$, we have $d_{G_\alpha}(w) = 2^{|w|}$. Next we define two separate loops each of length 2^α that initially place an even bet on the first bit of each subsequence to determine if the current block is all ones or all zeros. It then proceeds to make $2^\alpha - 1$ definite bets on the same bit. Figure 2.3 demonstrates such a construction for $\alpha = 2$.

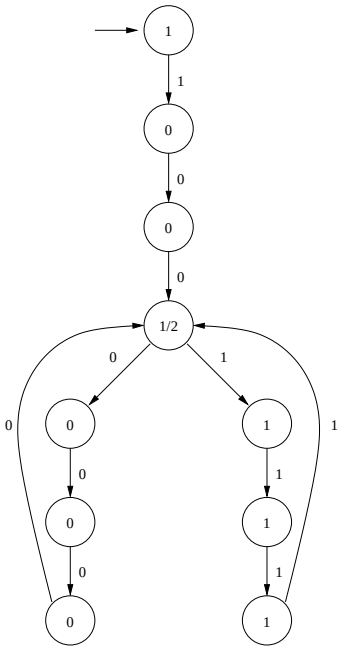

Figure 2.3: Finite-State Gambler G_2 for E

Following the exact same reasoning of the previous example, it is easy to see that for each construction of d_{G_α} we have a finite-state s-gale

$$d^{(s)}_{G_\alpha}(w) \geq 2^{|w|(s-2^{-\alpha})}.$$

For any $s > 0$ we choose $\alpha > \log s$ and build d_{G_α} which succeeds on all $s' > s$. Thus, $\mathcal{G}_{\mathrm{FS}} = \{(0, \infty)\}$ and we conclude that

$$\dim_{\mathrm{FS}}(\chi_E) = \inf \mathcal{G}(\chi_E) = 0.$$

□

2.2.4 Cantor Set

Though we are primarily interested in individual sequences, our final example considers *sets* of sequences and illustrates that, at least in some cases, finite-state gamblers are sufficient to show that the finite-state dimension can meet the equality in Fact 2.12. In particular, we consider the Cantor Set, a well-known fractal set, and show that it has finite-state dimension equal to its classical Hausdorff dimension. In terms of betting on successive "bits", we must consider a ternary signature, $\Sigma = \{0, 1, 2\}$.

Definition 2.17. The *Cantor Set*, denoted T_∞, is given by taking the interval $[0, 1]$ (set T_0), removing the middle third (T_1), removing the middle third of the two remaining pieces (T_2), and continuing this procedure ad infinitum. It is therefore

the set of points in the interval $[0,1]$ whose ternary expansions do not contain 1. That is,

$$T_\infty = \left\{ S \in \{0,1,2\}^\infty \middle| S[i] \neq 1 \forall i \geq 0 \right\}$$

Example 2.18. Let T_∞ be the Cantor Set, then

$$\dim_{\mathrm{FS}}(T_\infty) = \log_3 2.$$

Proof. Let $S \in T_\infty$. It is well established that $\dim_{\mathrm{H}}(T_\infty) = \log_3 2$, thus by Fact 2.12 it suffices to show that $\dim_{\mathrm{FS}}(T_\infty) \leq \log_3 2$, or simply that $S \in S^\infty[d_G^{(s)}]$ for any $s > s' > \log_3 2$. We construct an obvious, one-state finite-state gambler G that bets half of its capital that the next bit in the sequence is zero and the other half on 2. For any $w \in \{0,1,2\}^*$ such that $w \sqsubset S$ let $h(w,b)$ denote the number of occurrences of the bit b in w for $b \in \{0,1,2\}$. Then the finite-state martingale is

$$d_G(w) = 3^{|w|} \left(\frac{1}{2}\right)^{h(w,0)} \left(0\right)^{h(w,1)} \left(\frac{1}{2}\right)^{h(w,2)} = \left(\frac{1}{2}\right)^{|w|},$$

giving us a finite-state s-gale,

$$\begin{aligned} d_G^{(s)}(w) &= 3^{(s-1)|w|} d_G(w) \\ &= 3^{s|w|} \left(\tfrac{1}{2}\right)^{|w|} \\ &= \left(\tfrac{3^s}{2}\right)^{|w|}. \end{aligned}$$

It follows by our choice of s and s' that for any $s' > s$, as $|w| \to \infty$, $S \in S^\infty[d_G^{(s)}]$ and

$$\dim_{\mathrm{FS}}(T_\infty) = \inf \mathcal{G}_{\mathrm{FS}}(T_\infty) \leq \log_3 2$$

□

2.3 Established Results

In this section we survey some established results about sets of sequences and their finite-state dimension. We will then introduce some tools and concepts that were inspired by such results and which will be used throughout the remainder of the thesis.

2.3.1 Normal Sequences

Definition 2.19. A sequence $S \in \mathbf{C}$ is *normal*, and we write $S \in \mathrm{NORM}$, if for every $w \in \{0,1\}^*$,

$$\lim_{n\to\infty} \frac{1}{n} \left| \left\{ i < n \mid S[i \ldots i + |w| - 1] = w \right\} \right| = 2^{-|w|}.$$

This definition is a restriction of the usual concept of *normal numbers* with respect to their base 2 expansion. It should be noted that not all normal numbers are normal for every base $b \geq 2$, but those that are are referred to as *absolutely normal*. Very few sequences have been *proven* to be normal, but it has been conjectured that sequences corresponding to irrational numbers like π and $\sqrt{2}$ are absolutely normal. One sequence that has been shown to be absolutely normal [18] is the *Champernowne constant*, the infinite sequence produced from successively concatenating binary representations of positive integers;

$$\mathcal{C}_2 = 0.(1)(10)(11)(100)(101)(110)(111)\ldots$$

Expectedly, such sequences are the most difficult for a finite-state gambler to win on, thus every normal sequence has finite-state dimension 1.

Theorem 2.20. [14] Let $S \in \text{NORM}$. Then

$$\dim_{\text{FS}}(S) = 1.$$

2.3.2 AC_0 Sequences

The next result deals with the well studied complexity class AC_0 which includes languages that can be decided by a uniform family of circuits, polynomial in number, with unbounded fan-in AND, OR, and NOT gates and constant depth. Circuits in this class are able to do reasonably simply arithmetic operations including addition, subtraction and base-2 multiplications and divisions but not much else. Consequently, corresponding characteristic sequences of languages in AC_0 are relatively simple. Despite this seeming simplicity, we know the following result.

Theorem 2.21. [5] For every $r \in \mathbb{Q} \cap [0, 1]$ there exists a language $L \in \text{AC}_0$ such that $\dim_{\text{FS}}(\chi_L) = r$.

2.3.3 Rational Sequences

On the other side of the finite-state dimension spectrum we have *rational sequences*.

Definition 2.22. Let $n \in \mathbb{Z}^+$ and $S \in \mathbf{C}$.

1. S is *eventually periodic* with *period* n, and we write $S \in \mathbf{Q}_n$, if there exist $x \in \{0,1\}^*$ and $y \in \{0,1\}^n$ such that for all $k \in \mathbb{N}, xy^k \sqsubseteq S$. We also write $S = xy^\omega$.
2. S is *eventually periodic*, and we write $S \in \mathbf{Q}$ if there exists $n \in \mathbb{Z}^+$ such that $S \in \mathbf{Q}_n$.

Some literature refers to such sequences as *ultimately periodic* or *almost periodic*. We note, however, that $\mathbf{Q}$ is precisely the set of all binary expansions of elements of $\mathbb{Q} \cap [0, 1]$, so we will refer to sequences in $\mathbf{Q}$ as *rational sequences*. Rational sequences, being periodic, are very easy for a gambler to win on–a gambler simply

has to hold off on betting until the sequence becomes periodic. From then on a gambler can go into a loop, betting on each bit in y. Consequently, rational sequences have finite-state dimension zero.

Theorem 2.23. [5] For all $n \in \mathbb{Z}^+$,

$$\dim_{\mathrm{FS}}(\mathbf{Q}_n) = 0.$$

Since any rational sequence is periodic with period n for some n, it follows that any individual rational sequence also has finite-state dimension zero. On the other hand, the set of *all* such sequences, $\mathbf{Q}$ has dimension one.

Corollary 2.24. [5]

$$\dim_{\mathrm{FS}}(\mathbf{Q}) = 1.$$

2.3.4 Sequence Frequency

Another set of sequences that yield an interesting dimension result include sequences with similar frequencies. For any $w \in \{0,1\}^*$ the *height* of w, denoted $h(w)$ is the number of occurrences of 1 in w. Infinite sequences $S \in \mathbf{C}$ have limiting frequency (slope)

$$\mathrm{freq}_S(n) = \lim_{n\to\infty} \frac{h(S[0\ldots(n-1)])}{n}.$$

For each $\alpha \in [0,1]$, define

$$\mathrm{FREQ}(\alpha) = \big\{S \in \mathbf{C} \mid \mathrm{freq}_S(n) = \alpha\big\}.$$

Such sets have finite-state dimension corresponding to Shannon's binary entropy function,

$$\mathcal{H} : [0,1] \to [0,1]$$

$$\mathcal{H}(x) = x \log \frac{1}{x} + (1-x) \log \frac{1}{1-x}.$$

So that the function is continuous, its extreme points are defined as zero, $\mathcal{H}(0) = \mathcal{H}(1) = 0$.

Theorem 2.25. [5] For all $\alpha \in \mathbb{Q} \cap [0,1]$,

$$\dim_{\mathrm{FS}}(\mathrm{FREQ}(\alpha)) = \mathcal{H}(\alpha).$$

In the next chapter we will develop several tools, inspired by these results, that will aid us in our continued investigation.

Chapter 3

Finite State Dimension of Individual Sequences

Most of the results in Section 2.3 dealt with (infinite) sets of sequences in the Cantor space. In fact, classical Hausdorff Dimension can say nothing about individual sequences. Even a more refined concept, *constructive* dimension [11] gives dimension zero to any decidable sequence. By Fact 2.12 (2), we know that any individual sequence $S \in \text{FREQ}(\alpha)$ has *at most* $\mathcal{H}(\alpha)$ finite-state dimension, but such an upper bound is not very useful. Both Example 2.15 and 2.16 and are in $\text{FREQ}(\frac{1}{2})$ but have finite-state dimension 0. The Champernowne constant is in the set as well, but is a normal sequence with finite-state dimension 1.

From Theorem 2.21 we know that there are individual sequences with non-trivial dimension (that is, $0 < \dim_{\text{FS}}(S) < 1$). However, the proof of Theorem 2.21 (in [5]) uses a "dilution" technique–from a normal sequence in $S \in \text{AC}_0$ a dilution function pads S throughout with zero bits. This technique is shown to be able to transform the dimension to an arbitrary rational number in $(0, 1)$ while maintaining its membership in AC_0.

It would be nice to work the other way around. That is, *given* a sequence defined by a language, morphism[1], or some other standard method, it would be useful to have a technique to determine a non-trivial lower bound on said sequence. Of course it is relatively easy to give a non-trivial upper bound, one has only to built a finite-state gambler for the sequence. This is one of the major stepping stones to continuing the study of finite-state dimension and one that will most likely be pioneered in the coming years.

In this chapter we present our contributions and results towards this end. We begin in the first section by devloping a general type of finite-state gambler that we will use as our primary tool—betting trees. We then introduce a new conception of periodicity by considering fixed block sizes in a sequence. We use this concept to establish an upper bound for *any* sequence in terms of its block entropy rate. The

[1] See Section 3.4.

third section extends this to the usual notion of entropy. Next we use these results to evaluate some well studied classes of sequences. We conclude this chapter by giving an overview of the sequence hierarchy in terms of a sequence's complexity and dimension.

3.1 Betting Trees

Let $S \in \mathbf{C}$ be a sequence and fix $n \in \mathbb{Z}^+$. Now consider a subset $\Delta_n \subseteq \{0,1\}^n$. If Δ_n is missing some element(s) in $\{0,1\}^n$ then clearly we do not have to consider betting on such elements. This idea motivates what we will call *Betting Trees*.

Betting Trees are essentially directed binary trees (with orientation "downward" from the root to leaves) of depth n such that the edge from a vertex to a left sub-child corresponds to 0, 1 for the right sub-child. For each element $x \in \Delta_n$ there is a corresponding path from the root to the leaf (with directed edge back to the root). If $\Delta_n = \{0,1\}^n$, the betting tree will be a full binary tree. We define the betting function for each state in a manner that favors the left or right-sub tree proportionally to how many leaves are contained in them. This ensures that the expected capital after each block of size n is evenly distributed at the end of the betting tree. For sets Δ_n that are not full, the expected capital is greater than what we started out with. We now make a formal definition.

Definition 3.1. Let $S \in \mathbf{C}$ be a sequence and fix $n \in \mathbb{Z}^+$ for a set $\Delta_n \subseteq \{0,1\}^n$. Let $G_n = (Q, \delta, \beta, q_\lambda, c)$ be a finite-state gambler defined as follows.

- $Q = \{\ q_u \mid u \sqsubset x \text{ for } x \in \Delta_n\}$
- If $q_u, q_{ub} \in Q$ define

$$\delta(q_u, b) = \begin{cases} q_{ub} & \text{if } |u| < n-1 \\ q_\lambda & \text{if } |u| = n-1 \end{cases}$$

The betting function $\beta(q)$ for each state $q \in Q$ will be defined as a ratio from the number of possible leaves in the right-sub tree of q (corresponding to 1 as the next bit) to the total number of possible leaves reachable from q. To this end for all $u \in \{0,1\}^*$ such that q_u is a state, define

$$Y_u = \{\ x \in \Delta_n \mid u \sqsubset x\}.$$

- For all $q_u \in Q$ define

$$\beta(q_u) = \frac{|Y_{u1}|}{|Y_u|}$$

Such a gambler will be referred to as a *betting tree of order n*.

For clarity, we present an example.

Example 3.2. Fix $n = 4$ and let S_ϕ be the Fibonacci sequence defined by the morphism

$$\phi: \begin{array}{ccc} 0 & \to & 01 \\ 1 & \to & 0 \end{array}$$

$$S_\phi = 0100101001001010010100100101001001010010100101001001010\ldots$$

Define Δ_4 to be the factor set $F_4(S_\phi) = \{0010, 0100, 0101, 1001, 1010\}$. The finite-state betting tree for Δ_4 is shown in Figure 3.1. Note that state labels have been omitted. Rather, the values in each state correspond to $\beta(q)$.

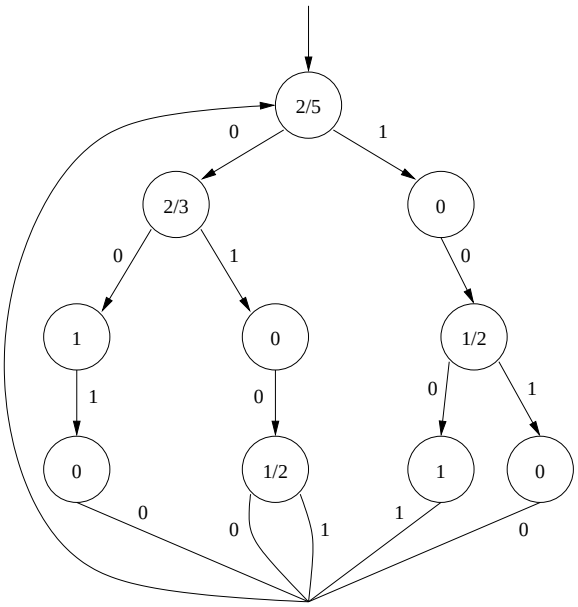

Figure 3.1: Betting Tree G_4

To conclude this section, we note that it is usual in the literature to allow the concatenation of a finite prefix to any right infinite sequence and only consider the infinite suffix. We can do the same with respect to finite-state dimension.

Proposition 3.3. Let $S, S' \in \mathbf{C}$ be sequences such that $S = xS'$ for $x \in \{0,1\}^*$, then

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

Proof. Let $S, S' \in \mathbf{C}$ be sequences such that $S = xS'$ for some $x \in \{0,1\}^*$. We will prove each inequality both ways.

Let $\dim_{\mathrm{FS}}(S) = r$. This means that there is an FSG $G = (Q, \delta, \beta, q_0, c_0)$ such that $S \in S^\infty[d_G^{(s)}]$ for any $s > r$. We transform G into a gambler G' for S' by redefining the initial state; $G' = (Q, \delta, \beta, \delta(q_0, x), c_0)$. Clearly $S' \in S^\infty[d_{G'}^{(s)}]$ for all $s > r$, therefore

$$\dim_{\mathrm{FS}}(S') \leq r = \dim_{\mathrm{FS}}(S).$$

Now say that $\dim_{\mathrm{FS}}(S') = r'$. This means that there is an FSG $G' = (Q', \delta', \beta', q'_0, c'_0)$ such that $S' \in S^{\infty}[d^s_{G'}]$ for any $s > r$. We transform G' into a gambler $G = (Q, \delta, \beta, q, c_0)$ for S as follows. Let $Q = Q' \cup \{q_{-|x|}, q_{-|x|+1}, \dots q_{-1}\}$ with transitions $\delta(q_i, b) = q_{i+1}$ for $-|x| \leq i \leq -2$ and $\delta(q_{-1}, b) = q'_0$ for any $b \in \{0, 1\}$. Though it makes little difference, we define $\beta(q_i) = \frac{1}{2}$ for $-|x| \leq i \leq -1$. All other transitions and betting functions remain the same as in G'. Finally, we set the initial state to $q = q_{-|x|}$. Since we place no bets on the prefix x, we observe that for any $w \in \{0, 1\}^*$ such that $|w| > |x|$,

$$d_G(w) = d_{G'}(w[|x| \dots |w| - 1]).$$

Since $w[|x| \dots |w| - 1] \sqsubset S'$ we conclude that $S \in S^{\infty}[d^{(s)}_G]$ for all $s > r$, therefore

$$\dim_{\mathrm{FS}}(S) \leq r' = \dim_{\mathrm{FS}}(S').$$

□

3.2 Block Periodicity

The factor set $F_n(S)$ of a sequence $S \in \mathbf{C}$ is defined such that a factor $x \in F_n(S)$ is allowed to appear starting at $S[i]$ for any index $i \geq 0$. In this section we offer a new notion of factor sets–block factor sets, wherein we fix a block size on a sequence and consider only strings appearing within these blocks. In effect, we exclude any strings that are strictly overlapping factors with respect to the block size.

Definition 3.4. Let $S \in \mathbf{C}$. For all $n \geq 0$ the *block factor set* of S is

$$B_n(S) = \Big\{ x \in \{0, 1\}^n \Big| \; S[in \dots (i+1)n - 1] = x \text{ for some } i \geq 0 \Big\}.$$

By Proposition 3.3, we can relax this definition and only consider block factor sets *after* a certain finite prefix.

We note that for normal sequences, $S \in \mathrm{NORM}$, $B_n(S) = F_n(S) = \{0, 1\}^n$ for all $n \geq 1$. For rational sequences, $S \in \mathbf{Q}$, after a certain finite prefix, $|B_n(S)| = 1$ where n is its period. For ever increasing block sizes, we can define the growth of the cardinality of the set $B_n(S)$.

Definition 3.5. Let $S \in \mathbf{C}$. The *block complexity function* of S is a function $\mathrm{bcf}_S : \mathbb{N} \to \mathbb{N}$ defined by

$$\mathrm{bcf}_S(n) = |B_n(S)|.$$

Note that when we say "complexity" we are referring to the growth rate of the function, *not* how the sequence may relate to computational complexity or the complexity hierarchy. We can now consider periodicity in terms of *bounding* the growth of the block factor set for ever increasing block sizes.

Definition 3.6. Let $S \in \mathbf{C}$. We say that S is *f-block periodic* if there exist infinitely many n such that

$$\mathrm{bcf}_S(n) \leq f(n).$$

By convention we will only consider the minimum such function $f(n)$. We will also write f in terms of its asymptotic classification. For instance, it is clear that *every* sequence is $\mathcal{O}(2^n)$-block periodic. Rational sequences are $\mathcal{O}(1)$-block periodic, and normal sequences are $\Theta(2^n)$-block periodic. Analogous to this new concept of periodicity, we define the entropy of individual sequences in terms of block factor sets.

Definition 3.7. The *block entropy* of an individual sequence $S \in \mathbf{C}$ is defined as

$$\mathtt{bh}(S) = \liminf_{n\to\infty} \frac{\log \mathrm{bcf}_S(n)}{n}.$$

We are now ready to present our main result. The block-entropy rate for any individual sequence provides an upper bound to the finite-state dimension of the sequence.

Theorem 3.8. Let $S \in \mathbf{C}$ be a sequence, then

$$\dim_{\mathrm{FS}}(S) \leq \mathtt{bh}(S).$$

The proof of Theorem 3.8 will use a betting tree as a gambler. First, however, we require the following lemma that establishes a general martingale function for any betting tree.

Lemma 3.9. Let $S \in \mathbf{C}$ be a sequence and fix $n \in \mathbb{Z}^+$. Let G be the betting tree for $\Delta_n = B_n(S)$. Then for all $w \in \{0,1\}^{nj}$, $j \geq 0$ such that $w \sqsubset S$, the martingale function for G is

$$d_G(w) = \left(\frac{2^n}{\mathrm{bcf}_S(n)}\right)^{\frac{|w|}{n}}.$$

Proof. Note that without loss of generality we've assumed that $|w| = nj$. Though we may lose money at intermediate steps within the betting tree, we will eventually make it up at the end of each cycle through the tree, thus this restriction poses no problems to a general martingale function for any $w \in \{0,1\}^*$, such that $w \sqsubset S$.

By Definition 3.4 it suffices to show that for any $x \in B_n(S)$,

$$d_G(x) = \left(\frac{2^n}{\mathrm{bcf}_S(n)}\right).$$

We will proceed by induction on the length of the prefixes of x. Let $u \in \{0,1\}^*$ and define

$$Y_u(S) = \{\ x \in B_n(S) \mid u \sqsubseteq x\}.$$

We will show that the martingale of any string $u \sqsubseteq x \in B_n(S)$ is

$$d_G(u) = 2^{|u|}\frac{|Y_u(S)|}{\mathrm{bcf}_S(n)}. \tag{3.1}$$

Equation 3.1 is obvious for $|u| = 0$, the root of the betting tree corresponds to the initial capital $c_0 = 1$. Note also that $|Y_\lambda(S)| = \mathrm{bcf}_S(n)$. We now assume 3.1 holds

for $|u| = i$,

$$d_G(u) = 2^i \frac{|Y_u(S)|}{\mathrm{bcf}_S(n)}.$$

It is clear that if $|Y_u(S)| = 0$ then $x \notin B_n(S)$ and so the capital is zero. For $b \in \{0,1\}$, the expected payoff after ub is

$$\begin{aligned} d_G(ub) &= 2d_G(u)\left[(1-b)(1-\beta(q_u)) + b\beta(q_u)\right] \\ &= 2^{i+1}\frac{|Y_u(S)|}{\mathrm{bcf}_S(n)}\left[(1-b)(1-\frac{|Y_{u1}|}{|Y_u|}) + b\frac{|Y_{u1}|}{|Y_u|}\right]. \end{aligned}$$

It is clear that $Y_u(S) = Y_u$ for every proper prefix $u \sqsubset x$, so we have

$$d_G(ub) = 2^{i+1}\frac{|Y_{ub}(S)|}{\mathrm{bcf}_S(n)}.$$

At the end of our tree, $u = x$ so $Y_x(S) = \{x\}$ and we have

$$d_G(x) = 2^n \frac{|Y_x(S)|}{\mathrm{bcf}_S(n)} = \frac{2^n}{\mathrm{bcf}_S(n)}.$$

□

Proof. of Theorem 3.8. Lemma 3.9 gives us a general martingale function for any betting tree G of order n for any sequence $S \in \mathbf{C}$ with respect to the block factor set $B_n(S)$. For any $w \in \{0,1\}^{ni}, i \geq 0$ such that $w \sqsubset S$ The corresponding finite-state s-gale is

$$\begin{aligned} d_G^{(s)}(w) &= 2^{(s-1)|w|} d_G(w) \\ &= 2^{(s-1)|w|}\left(\tfrac{2^n}{\mathrm{bcf}_S(n)}\right)^{\frac{|w|}{n}} \\ &= 2^{(s-1)|w|}2^{|w|}(\mathrm{bcf}_S(n)^{-\frac{|w|}{n}}) \\ &= \left(2^s(\mathrm{bcf}_S(n)^{-\frac{1}{n}})\right)^{|w|} \\ &= \left(2^{s-\frac{\log \mathrm{bcf}_S(n)}{n}}\right)^{|w|}. \end{aligned}$$

For any $s > \mathtt{bh}(S)$, we will show that $\dim_{\mathrm{FS}}(S) \leq s$. For any s' such that $s > s' > \mathtt{bh}(S) = \lim_{n\to\infty} \frac{\log \mathrm{bcf}_S(n)}{n}$ we choose n large enough such that $s' > \frac{\log \mathrm{bcf}_S(n)}{n}$ and construct the s-gale $d_G^{(s)}$ defined by the betting tree of order n. Clearly, $S \in S^\infty[d_G^{(s)}]$ and we conclude that

$$\dim_{\mathrm{FS}}(S) = \inf \mathcal{G}_{\mathrm{FS}} \leq \mathtt{bh}(S).$$

□

Corollary 3.10. Let S be an f-block periodic sequence such that $f = 2^{o(n)}$ then

$$\dim_{\mathrm{FS}}(S) = 0.$$

Proof. Let S be $2^{o(n)}$-block periodic. By Definition 3.6 it follows that

$$\mathtt{bh}(S) = \liminf_{n\to\infty} \frac{\log 2^{o(n)}}{n} = 0.$$

It follows from Theorem 3.8 that

$$\dim_{\mathrm{FS}}(S) \leq \mathtt{bh}(S) = 0.$$

□

3.3 Entropy

Each concept that applied to fixed blocks of a sequence in the previous section has corresponding concepts in stringology with respect to factor sets $F_n(S)$.

Definition 3.11. [9] For a sequence $S \in \mathbf{C}$, the *complexity function* is a function that counts, for each integer $n \geq 0$, the number $p_S(n)$ of factors of length n appearing in S, thus

$$p_S(n) = |F_n(S)|.$$

Obviously, for any $S \in \mathbf{C}$, $p_S(0) = 1$ since $F_0(S) = \{\lambda\}$ and $p_S(1) = 2$ since $F_1(S) = \{0, 1\}$ (for any sequence that is not unary of course).

It is clear from the definitions that for all n, the complexity function is *at least* as big the block-complexity function,

$$\mathrm{bcf}_S(n) \leq p_S(n).$$

Definition 3.12. [1] The *entropy* of an individual sequence $S \in \mathbf{C}$ is defined as

$$\mathtt{h}(S) = \lim_{n\to\infty} \frac{\log p_S(n)}{n}.$$

It is clear here as well that the entropy rate is always at least the block entropy rate,

$$\mathtt{bh}(S) \leq \mathtt{h}(S).$$

Corollary 3.13. For any $S \in \mathbf{C}$,

$$\dim_{\mathrm{FS}}(S) \leq \mathtt{h}(S).$$

Proof. Since for any $S \in \mathbf{C}$, $\mathtt{bh}(S) \leq \mathtt{h}(S)$ the corollary follows from Theorem 3.8. The proof of Theorem 3.8 can be modified for $\Delta_n = F_n(S)$ providing a proof independent of our block characterization. □

Similarly, factor sets bounded by $2^{o(n)}$ have entropy zero, and thus finite-state dimension zero.

Corollary 3.14. Let $S \in \mathbf{C}$ be a sequence such that $p_S(n) = 2^{o(n)}$, then

$$\dim_{\mathrm{FS}}(S) = 0.$$

Proof. Let $S \in \mathbf{C}$ have a complexity function that is bounded by $2^{o(n)}$, then the entropy rate is

$$\mathtt{h}(S) = \lim_{n\to\infty} \frac{\log 2^{o(n)}}{n} = 0.$$

So by Corollary 3.13 we conclude that $\dim_{\mathrm{FS}}(S) = 0$. □

Note the subtle difference here. If the usual complexity function is bounded by $2^{o(n)}$ then so is the block complexity function. Similar for the entropy rates. However, if $p_S(n) = 2^{cn}$ for $c \in (0, 1]$, it *may* be the case that $\mathrm{bcf}_S(n)$ is still bounded by $2^{o(n)}$ in which case, though the usual entropy rate fails to give a tight upper bound on the finite-state dimension, the block entropy rate still does.

This poses a question for further study–do such sequences even exist? If so then the inequality in 3.13 is strict and our block characterization gives a new, very useful way of conceptualizing sequences. We continue discussion of such questions in the final chapter.

3.4 Automatic Sequences

We now turn our attention to some very well studied sequences called *Automatic Sequences*. Consider a deterministic finite automaton (DFA) D that on input w is able to output 1 if it accepts w and 0 if it rejects w. Such an automaton defines a *finite-state function* $f_D : \Sigma^* \to \{0, 1\}$. Now consider feeding the DFA with the canonical binary representations of non-negative integers, $0, 1, 2, \ldots$, then the finite-state function *generates* a sequence, $(a_n)_{n\geq 0}$ where $a_n = 1$ if D accepts and 0 if D rejects. This type of sequence is known as an *automatic sequence*[2].

Theorem 3.15. [1] A sequence $S \in \mathbf{C}$ is automatic if and only if there exists a deterministic finite automaton such that $S = (a_n)_{n\geq 0}$.

Expectedly, with the weak power of finite-state machines, automatic sequences are not too complex.

Theorem 3.16. [4] Let $S \in \mathbf{C}$ be an automatic sequence over $\Sigma = \{0, 1\}$, then S has a linear complexity function.

$$p_S(n) \in \mathcal{O}(n).$$

[2]Usually k-automatic is the term used, which refers to a general signature where $k = |\Sigma|$. Since we restrict ourselves to $\Sigma = \{0, 1\}$, we simply refer to them as automatic sequences when we really mean 2-automatic sequences.

Of course not all sequences are automatic. Take for instance the sequence representing the prime numbers,

n	0	1	2	3	4	5	6	7	8	...
PRIMES_n	0	0	1	1	0	1	0	1	0	...

It has been shown that PRIMES_n is not automatic. Interestingly, however, its entropy is zero [1], $\mathtt{h}(\text{PRIMES}_n) = 0$ and thus has finite-state dimension zero as well. Because of their close relation to finite-state machines, it is not surprising that automatic sequences have finite-state dimension zero.

Corollary 3.17. Let $S \in \mathbf{C}$ be an automatic sequence, then

$$\dim_{\text{FS}}(S) = 0.$$

Proof. Let $S \in \mathbf{C}$ be an automatic sequence. By Theorem 3.16, for automatic sequences, $p_S(n) \in \mathcal{O}(n)$ thus

$$\mathtt{h}(S) = \lim_{n\to\infty} \frac{\log p_S(n)}{n} = 0.$$

By Corollary 3.13 it follows that $\dim_{\text{FS}}(S) = 0$. □

Regular languages are ones that can be decided by DFAs, thus there is a very natural correspondence between regular languages and automatic sequences. They are, in fact, one in the same.

Theorem 3.18. [13] Let $L \subseteq \Sigma^*$ be a language. L is regular if and only if χ_L is automatic.

We have greatly simplified the wording of Theorem 3.18 in order to avoid introducing too many more definitions. Rigo [13] showed that one can generalize the finite-state function to feed a DFA with *every* string $x \in \Sigma^*$ in lexicographic order to generate a language's characteristic sequence. He showed that this model of a finite-state function is equivalent to the more restrictive model because one can use a finite-state transducer that, given n in canonical form, computes[3] the n-th lexicographically ordered string in Σ^*. The usual DFA is then run on this string to define f_D.

Corollary 3.19. Let $L \subseteq \Sigma^*$ be a regular language with characteristic sequence χ_L, then

$$\dim_{\text{FS}}(\chi_L) = 0.$$

Proof. Let $L \subseteq \Sigma^*$ be a regular language. Then there exists a finite-state automaton M that decides L. We modify M to output 0 in every reject state and 1 in every accept state. The resulting machine defines a finite-state function and generates the characteristic sequence χ_L. It follows then that χ_L is an automatic sequence so by Corollary 3.17,

$$\dim_{\text{FS}}(\chi_L) = 0.$$

[3] You simply take n in binary, add one and drop the leading 1 bit

□

It is interesting to note that our entire investigation started out by considering Corollary 3.19. Several different approaches were taken until the more general result in Theorem 3.8 was proven, at which point all other results followed. Though a single characteristic sequence of a regular language may have finite-state dimension zero, the set of all such sequences has dimension one.

Theorem 3.20. [2] If the characteristic sequence χ_L of a language L is a rational sequence (eventually periodic) then L is regular a regular language.

Obviously the converse of Theorem 3.20 is not true, Examples 2.15 and 2.16 testify to this. However, this does yield an immediate observation.

Observation 3.21. Define the set REG $\subseteq \mathbf{C}$ to be the set of all characteristic strings of regular languages.

$$\text{REG} = \big\{\chi_L \in \mathbf{C} \;\big|\; L \text{ is a regular language}\big\},$$

then

$$\dim_{\text{FS}}(\text{REG}) = 1.$$

Proof. By [5], we know that $\dim_{\text{FS}}(\mathbf{Q}) = 1$ so by Fact 2.12 (2) it suffices to show that $\mathbf{Q} \subseteq$ REG. Let $S \in \mathbf{Q}$, clearly S must be a rational sequence. By Theorem 3.20, S corresponds to a regular language thus $S \in$ REG. □

Automatic sequences are closely related to morphic sequences. A function $\varphi : \Sigma^* \to \Sigma^*$ is called a *morphism* if $\varphi(xy) = \varphi(x)\varphi(y)$ for all $x, y \in \Sigma^*$. The iterative application of a morphism φ is defined as $\varphi^0(b) = b$ and $\varphi^i(b) = \varphi(\varphi^{i-1}(b))$ for $b \in \{0, 1\}$. A morphism is *expanding* if $|\varphi(b)| \geq 2$ for all $b \in \{0, 1\}$. We call a morphism k-uniform if $|\varphi(b)| = k$ for all $b \in \{0, 1\}$. A 1-uniform morphism is called a *coding*. Morphisms can be very naturally applied to sequences $S \in \mathbf{C}$,

$$\varphi(S) = \varphi(S[0])\varphi(S[1])\varphi(S[2])\ldots$$

If $\varphi(S) = S$ then φ is called a *fixed point morphism*.

The continued application of an expanding morphism may define a sequence $S \in \mathbf{C}$. If for some $b \in \{0, 1\}$ and $x \in \Sigma^+$, $\varphi(b) = bx$ then we say that φ is *prolongable* on b. The sequence defined by such a morphism *converges* to

$$S = \varphi^\omega(b) = bx\varphi(x)\varphi^2(x)\varphi^3(x)\ldots$$

which is also a fixed point of φ. That is, $\varphi(\varphi^\omega(b)) = \varphi^\omega(b)$. Such a sequence is called a *pure morphic sequence*. If there is a coding $\tau : \Sigma \to \Sigma$ such that $S = \tau(\varphi^\omega(b))$ then it is simply a *morphic sequence*.

Theorem 3.22. [6] The complexity of a sequence $S \in \mathbf{C}$ that is a fixed point of any morphism (not necessarily of constant length) satisfies

$$p_S(n) \in \mathcal{O}(n^2)$$

Theorem 3.22 tells us that *any* morphic sequence will have a factor set bounded by some quadratic function. It follows then that any morphic sequence has finite state dimension zero.

Corollary 3.23. Let $S \in \mathbf{C}$ be a morphic sequence, then

$$\dim_{\mathrm{FS}}(S) = 0.$$

Proof. Clear. □

This covers sequences that can be produced or recognized by relatively simple morphisms and finite-state machines. However, "most" sequences in the Cantor Space do not have such low complexity.

Theorem 3.24. [1] Almost all sequences $S \in \mathbf{C}$ are *complete sequences*. That is, $p_S(n) = 2^n$ for all $n \geq 0$.

3.5 Hierarchy Results

The complexity function for factor sets partitions all sequences into two categories. If ever $p_S(n) = p_S(n+1)$ then S is a rational sequence, otherwise $p_S(n)$ is always monotonically increasing.

Theorem 3.25. [9] Let $S \in \mathbf{C}$. Then the following hold.

1. If $S \in \mathbf{Q}_n$ then $p_n(S)$ is strictly increasing until $p_n(S) = n$, then it is constant thereafter.
2. If S is aperiodic, then $p_n(S)$ is strictly increasing, $p_n(S) \geq n+1$ for all $n \geq 1$.

Sequences that meet the inequality in condition 2 above are called *Sturmian sequences*. Sturmian sequences have been well studied and many equivalent descriptions have been developed leading to far reaching applications in linear filters, network routing and computer graphics. Sturmian sequences constitute a "gap" in the sequence hierarchy with respect to the complexity function.

Observation 3.26. There exists no sequence $S \in \mathbf{C}$ such that $p_n(S) \in o(n)$.

Sturmian sequences are the lowest order of complexity for aperiodic sequences. It is unclear, however, if sequences exist that have a block complexity function $B_n(S) \in o(n)$ or not. There are, however, sequences that have monotonically increasing complexity functions but that are *constantly-block periodic* ($\Theta(1)$-block periodic).

Example 3.27. The *Thue-Morse* sequence is an automatic sequence $S_\mu = \mu^\omega(0)$ defined by the morphism

$$\mu : \begin{array}{ccc} 0 & \to & 01 \\ 1 & \to & 10 \end{array}$$

$$S_\mu = 0110100110010110100101100110100 1\ldots$$

For any $n = 2^k$, $k \geq 0$, $B_n(S) = 2$, but since S_μ is automatic it has a linear complexity function. Experimentally it seems that $p_{S_\mu}(n) \approx 3n$.

Furthermore, Examples 2.15 and 2.16 are constantly-block periodic with $c = 2$. Rather simple sequences can be built out of regular languages for any $c \in \mathbb{Z}^+$. However, it is not likely that *any* Sturmian sequence is also constantly-block periodic.

Conjecture 3.28. Let $S \in \mathbf{C}$ be a Sturmian Sequence. For every $n \in \mathbb{Z}^+$, $x \in F_n(S)$, x eventually appears (infinitely often and within bounded occurrences of each other) starting at every index r mod n.

Though no proof could be formulated, this conjecture is believed to hold [17]. By definition, a Sturmian sequence is $(n+1)$-block periodic. However, if Conjecture 3.28 holds, then *no* Sturmian Sequence is constantly-block periodic. This constitutes a similar "gap" in the sequence hierarchy with respect to the block complexity function.

As a conclusion to this chapter, we offer the following view of the *hierarchy* of sequences with respect to the complexity function $p_S(n)$.

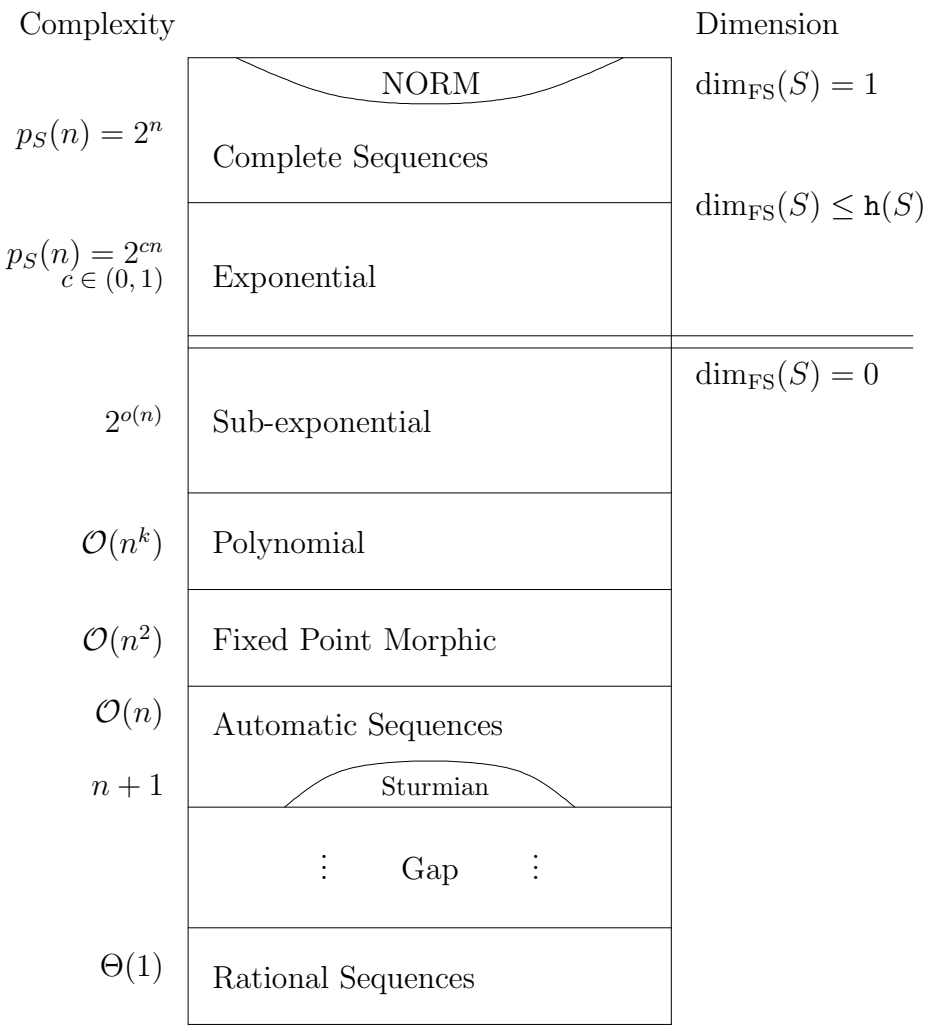

Figure 3.2: Sequence Hierarchy w.r.t. $p_S(n)$ Complexity Function

Chapter 4

Further Study & Conclusions

4.1 Further Study

As a prelude to our concluding remarks, we pose several open questions and considerations that we believe are worthy of continued investigation.

4.1.1 Finite-State Lower Bounds

Theorem 3.8 and Corollary 3.13 both provide a nice upper bound in terms of block and factor set entropy rates respectively. However, these bounds are not at all tight. Sequences can be constructed that have entropy 1 but finite-state dimension 0. Showing strict equality in finite-state dimension is relatively simple when dealing with *sets* of infinite sequences. In that case, one can use a generalized Kraft inequality to argue that a sequence exists in the set in question such that no gambler can win on it. Such techniques were used in [5] and [10] to obtain results like Theorem 2.25. Unfortunately, when dealing with *singleton* sets (individual sequences), such an argument doesn't apply. Additional techniques will have to be developed to show non-trivial lower bounds on individual sequences.

4.1.2 Block Characterization

Recall that the Thue-Morse sequence represents a sequence that is linear in terms of its complexity function, $p_{S_\mu}(n) \approx 3n$, but which is constantly-block periodic, $\mathrm{bcf}_{S_\mu}(n) = 2$ for $n = 2^k, k \geq 0$. However, the entropy rates for both characterizations are zero. This then begs the question, are there sequences that have positive entropy rates but are constantly-block periodic? More generally, we ask the following.

Open Question 4.1. Do there exist sequences $S \in \mathbf{C}$ such that $p_S(n) = 2^{cn}$ for $c \in (0, 1]$ such that $\mathtt{bh}(S) < \mathtt{h}(S)$?

If such sequences were to exist then the usual notion of entropy would not provide a strict upper bound on finite-state dimension for most sequences. This seems rather unlikely given our views in the previous subsection.

4.1.3 State Sizes

Recall that in Examples 2.15 and 2.16 we gave generalized constructions that required an exponential number of states with respect to α. However, we also had an exponential *decrease* in the upper bound on the dimension for increasing α.

Now consider our betting trees. Each tree G_n has a depth of n and a width corresponding to the cardinality of the set Δ_n. A very general bound on the number of states in any betting tree is thus $\mathcal{O}(n|\Delta_n|)$.

Recall that Sturmian sequences are ones with a minimal complexity function, $p_S(n) = n + 1$ for all n. It follows then that for any Sturmian sequence the betting tree has $\Theta(n^2)$ states. For each increasing value of n we get a linear blow up in the number of states but also receive a linear decrease in the s parameter of the s-gale defined by G_n.

More generally, say a sequence's block factor set (or factor set) is bounded by $2^{f(n)}$. If $f(n) \in \mathcal{O}(\log n)$ then the number of states in G_n is polynomial. If $f(n) \in \omega(\log n)$ then we necessarily have a sub-exponential number of states. Finally, if $f(n) \in \mathcal{O}(n)$ then we have a strictly exponential number of states.

We naturally ask then, for a given sequence S, is there a gambler for S with a constantly bound number of states? Linearly bound? Polynomially bound? More generally,

Open Question 4.2. What is the lower bound on the number of states required for a gambler to succeed on a given sequence S?

For Sturmian sequences it may be possible to design a linearly sized gambler. An earlier approach used Rauzy graphs [9] (factor graphs) to show Sturmian sequences have finite-state dimension zero. This idea was abandoned after the more general result in Theorem 3.8 subsumed it. It may be worthwhile to reconsider. Experimentally it seems that though the gamblers are linear in n, one must go exponentially far in n to get a polynomial decrease in the dimensional bound.

4.1.4 Scaled Dimension

Recall that the s parameter in s-gales can be interpreted as a house advantage against a gambler. For each bet, a gambler loses money if $s < 1$. As $s \to 0$ it becomes more and more difficult for a gambler to win. In fact, there is a very natural hierarchy of *scaled dimension*. For each integer i the i-th order dimension, $\dim^{(i)}$, can be defined on a set of sequences. By rescaling dimension we gain a finer characterization of a set's dimension. Intuitively, scaled dimension requires a gambler to be able to win by increasing orders of magnitude. The house advantage

that s represented in s-gales is compounded with each higher order dimension. Such ideas are present in the classical Hausdorff setting and have been extended to Lutz's effectivization model [7]. It is natural to ask then, to what degree can our results be extended to scaled dimension? It would be much stronger, for example, to say that every automatic sequence has finite-state dimension zero in higher order scaled dimensions. We anticipate this line of investigation to be full of potential and we are currently considering it.

4.2 Conclusion

Lutz's gale characterization of classical Hausdorff dimension is very intuitive and extensible to many areas of mathematics and computer science. The results thus far have been encouraging and we are proud to make our contributions to the area. Our block characterization and corresponding upper bound that the block entropy rate represents may prove to be quite useful in continued research. We have also established the finite-state dimension of a large number of well studied sequences. This area is still young, yet already many interesting results have been established. No doubt much more will be discovered in the coming years.

Bibliography

[1] Jean-Paul Allouche and Jeffrey Shallit. *Automatic Sequences*. Cambridge University Press, first edition, 2003.

[2] Klaus Ambos-Spies and Edgar Busse. Automatic forcing and genericity: On the diagonalization strength of finite automata. *Discrete Mathematics and Theoretical Computer Science*, pages 97–108, 2003.

[3] Jean Berstel. Recent results in sturmian words. *Developments In Language Theory II*, pages 13–24, 1995.

[4] Alan Cobham. Uniform tag sequences. *Mathematical Systems Theory*, 6:164–192, 1972.

[5] Jack J. Dai, James I. Lathrop, Jack H. Lutz, and Elvira Mayordomo. Finite-state dimension. *Theoretical Computer Science*, 310:1–33, 2004.

[6] A. Ehrenfeucht, K.P. Lee, and G. Rozenberg. Subword complexities of various classes of deterministic developmental languages without interaction. *Theoretical Computer Science*, 1:59–75, 1975.

[7] John M. Hitchcock, Jack H. Lutz, and Elvira Mayordomo. Scaled dimension and nonuniform complexity. *30th International Colloquium on Automata, Languages, and Programming*, pages 278–290, 2003.

[8] M. Lothaire. *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company, 1983.

[9] M. Lothaire. *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2002.

[10] Jack H. Lutz. Dimension in complexity classes. *SIAM Journal on Computing*, 32:1236–1259, 2003.

[11] Jack H. Lutz. The dimension of individual strings and sequences. *Information and Computation*, 187:49–79, 2003.

[12] Elvira Mayordomo. Effective hausdorff dimension. *Foundations of Formal Sciences II, Applications of Mathematical Logic in Philosophy and Linguistics*, pages 1–16, 2000.

[13] Michel Rigo. Generalization of automatic sequences for numeration systems on a regular language. *Theoretical Computer Science*, 244:271–281, 2000.

[14] C. P. Schnorr and H. Stimm. Endliche automaten und zufallsfolgen. *Acta Informatica*, 1:345–359, 1972.

[15] Michael Sipser. *Introduction to the Theory of Computation*. PWS Publishing Company, first edition, 1997.

[16] M. J. Strauss. Normal numbers and sources for bpp. *Theoretical Computer Science*, 178:155–169, 1997.

[17] Pascal Weil and Jean Berstel. Personal communication, 2004.

[18] Eric Weinstein. Champernowne constant. *MathWorld–A Wolfram Web Resource.* `http://mathworld.wolfram.com/ChampernowneConstant.html`, 2002.