Dimension is Compression

María López-Valdés*† Elvira Mayordomo*†

June 24, 2012

Abstract

Effective fractal dimension was defined by Lutz (2003) in order to quantitatively analyze the structure of complexity classes. Interesting connections of effective dimension with information theory were also found, in fact the cases of polynomial-space and constructive dimension can be precisely characterized in terms of Kolmogorov complexity, while analogous results for polynomial-time dimension haven't been found.

In this paper we remedy the situation by using the natural concept of reversible time-bounded compression for finite strings. We completely characterize polynomial-time dimension in terms of polynomial-time compressors.

1 Introduction

Effective fractal dimension was defined in [19] in order to quantitatively analyze the structure of complexity classes with the immediate motivation of overcoming the limitations of resource-bounded measure, a generalization of classical Lebesgue measure [18]. Important applications in Computational Complexity have been found including circuit-size complexity, polynomial-time degrees, the size of NP, zero-one laws, and oracle classes (see [21, 13, 9] for a summary of the main results).

In parallel, the connections of this effective dimension with algorithmic information started being patent, as it could be suspected from earlier results by Ryabko [25, 26], Staiger [27, 28], and Cai and Hartmanis [6]. The cases of constructive, recursive and polynomial-space dimension were characterized precisely as the best case asymptotic compression rate when using plain, recursive, and polynomial-space-bounded Kolmogorov complexity, respectively [23, 20, 11], and the low resource-bounds of finite-state and pushdown devices have been connected to the corresponding compression algorithms [8, 1]. See [24] for an overall survey.

*Dept. de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, SPAIN. {marlopez, elvira}@unizar.es

†This work was supported by the Spanish Ministry of Science and Innovation (Projects TIN2008-06582-C03-02, TIN2011-27479-C04-01)

But the case of polynomial-time bounds remained elusive [14]. This is not strange since computing even approximately the value of time-bounded Kolmogorov complexity seems to require an exponential search. The main difference with space-bounded Kolmogorov complexity is reversibility, in this later case the encoding phase can be performed within similar space-bounds.

In this paper we look at the usual notion of compression algorithm for finite strings. A polynomial-time compression scheme is just a pair of encoder and decoder algorithms, both working in polynomial-time. We consider encoders that do not completely start from scratch when working on an extension of a previous input. This last condition is formalized in Section 3 with a conditional-entropy like inequality.

We exactly characterize polynomial-time (or p) -dimension as the best case asymptotic (that is, i.o.) compression ratio attained by these polynomial-time compression schemes. Dually, strong polynomial-time-dimension [3] corresponds to the worst case asymptotic compression ratio. The proof of these results uses and interesting generalization of arithmetic coding (see [7] for an introduction to arithmetic coding and its history).

Several results on the polynomial-time dimension of complexity classes can be now interpreted as compressibility results. For example, the (characteristic sequences of) languages in a class of p-dimension 1 cannot be i.o. compressed by more that a sublinear amount. Here we obtain results on the compressibility of complete and autoreducible languages.

Buhrman and Longprè have given a characterization of p-measure in terms of compressibility in [5], but in that case the compressors are restricted to extenders and the encoder is required to give several alternatives, one of them being the correct output. In the light of our present results we can view effective dimension as an information content measure for infinite strings, whereas resource-bounded measure can only distinguish the extreme case of non measure 0 classes that are the most incompressible ones.

2 Preliminaries

The Cantor space $\mathbf{C}$ is the set of all infinite binary sequences. If $w \in \{0,1\}^*$ and $x \in \{0,1\}^* \cup \mathbf{C}$, $w \sqsubseteq x$ means that w is a prefix of x. For each $w \in \{0,1\}^*$ we define the w-cylinder, $C_w = \{x \in \mathbf{C} | w \sqsubseteq x\}$.

For $0 \leq i \leq j$, we write $x[i \dots j]$ for the string consisting of the i-th through the j-th bits of x. We use λ for the empty string. We write $s_0, s_1, s_2, \dots$ for the standard enumeration of $\{0,1\}^*$. We write $next(x)$ for the string following x in the standard enumeration, i.e., $next(s_n) = s_{n+1}$. $x < y$ denotes that string x precedes y in the same standard enumeration.

Let E= DTIME($2^{O(n)}$).

Definition. Let $s \in [0, \infty)$.

1. An *s-gale* is a function $d : \{0,1\}^* \to [0, \infty)$ satisfying

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

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

2. A *martingale* is a 1-gale, that is, a function $d : \{0,1\}^* \to [0,\infty)$ satisfying

$$d(w) = \frac{d(w0) + d(w1)}{2}$$

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

Definition. Let $s \in [0,\infty)$ and d be an s-gale.

1. We say that d *succeeds* on a sequence $S \in \mathbf{C}$ if

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

The success set of d is $S^\infty[d] = \{S \in \mathbf{C} \mid d \,\text{succeeds on}\, S\}$

2. We say that d *succeeds strongly* on a sequence $S \in \mathbf{C}$ if

$$\liminf_{n\to\infty} d(S[0\ldots n]) = \infty$$

The strong success set of d is $S^\infty_{\text{str}}[d] = \{S \in \mathbf{C} \mid d \,\text{succeeds strongly on}\, S\}$

Definition. We say that a function $d : \{0,1\}^* \to [0,\infty)$ is p-*computable* if there is a function $\hat{d} : \{0,1\}^* \times \mathbb{N} \to \mathbb{Q}$ such that $\hat{d}(w,r)$ is computable in time polynomial in $|w| + r$ and $|\hat{d}(w,r) - d(w)| \le 2^{-r}$ holds for all w and r.

We say that a function $d : \{0,1\}^* \to [0,\infty) \cap \mathbb{Q}$ is *exactly* p-*computable* if $d(w)$ is computable in time polynomial in $|w|$.

Definition. Let $X \subseteq \mathbf{C}$,

1. The p-*dimension* of X is

$$\dim_{\mathrm{p}}(X) = \inf\left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a p-computable } s\text{-gale } d \text{ s.t.} \\ X \subseteq S^\infty[d] \end{array} \right\}$$

2. The *strong* p-*dimension* of X is

$$\mathrm{Dim}_{\mathrm{p}}(X) = \inf\left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a p-computable } s\text{-gale } d \text{ s.t.} \\ X \subseteq S^\infty_{\text{str}}[d] \end{array} \right\}$$

By the exact computation lemma in [19] p-computable and exactly p-computable gales are interchangeable in the two definitions above.

Theorem 2.1 *Let $X \subseteq \mathbf{C}$,*

$$\dim_{\mathrm{p}}(X) = \inf\left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \textit{there is an exactly } \text{p-}\textit{computable } s\textit{-gale } d \textit{ s.t.} \\ X \subseteq S^\infty[d] \end{array} \right\}$$

$$\mathrm{Dim}_{\mathrm{p}}(X) = \inf\left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \textit{there is an exactly } \text{p-}\textit{computable } s\textit{-gale } d \textit{ s.t.} \\ X \subseteq S^\infty_{\text{str}}[d] \end{array} \right\}$$

For a complete introduction and motivation of effective dimension and effective strong dimension see [21].

3 Compressors that do not start from scratch

In this section we develop the notion of compressors that "do not start from scratch" in the sense that when encoding successively longer extensions of an input, the outputs are restricted in the way we make precise below. The extreme case of this behavior is when the compressor is a mere extender, that is, $C(w)$ is always a prefix of $C(wu)$. We consider here a much weaker restriction than extension.

Definition. A pair of functions (C, D) (C the encoder, D the decoder) $C, D : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ is a *polynomial-time compressor* if:

(i) C and D can be computed in polynomial-time on their corresponding input length.

(ii) For all $w \in \{0,1\}^*$, $D(C(w), |w|) = w$.

In this paper, we can make all encoders prefix-free in each length, that is, $C(\{0,1\}^n)$ is a prefix-free set for each n. Notice that for the applications in this paper, that use asymptotic compression rates, this restriction is not significant since for each encoder there is a prefix free one with the same compression rate.

Notice that in the previous definition there is no restriction whatsoever on the behavior of C, the encoder, when working on two inputs that are one an extension of the other. For instance, we can have $|C(wu)| \ll |C(w)|$ and $C(wu)$ can have no common prefix with $C(w)$. In definition 3 we introduce a restriction on the compressor that has an effect on the variety of $C(wu)$ for different u, that will be controlled by $|C(w)|$.

Definition. A polynomial-time compressor (C, D) *does not start from scratch* if for all $\epsilon > 0$ there exist $c > 0$ such that for all but finitely many $w \in \{0,1\}^*$ there exists $k \leq c \log(|w|)$, $k > 0$, such that

$$\sum_{|u| \leq k} 2^{-|C(wu)|} \leq 2^{\epsilon k} 2^{-|C(w)|}. \tag{1}$$

We will consider only compressors that do not start from scratch.

Notice that when there is a constant k such that $\sum_{|u|\leq k} 2^{-|C(wu)|} \leq (k+1) \cdot 2^{-|C(w)|}$, condition (1) is trivial, while in general $\sum_{|u|\leq k} 2^{-|C(wu)|}$ can be as large as $k+1$, so condition (1) is a proper restriction on compressors.

We first remark that if $C(w)$ and $C(wu)$ have a long common prefix then C fulfills condition (1).

Lemma 3.1 *Polynomial-time compressors for which $C(w)$ and $C(wu)$ have a common prefix of length at least $|C(w)| - O(\log(|w|))$ (for all $w, u \in \{0,1\}^*$) don't start from scratch.*

Proof. Using that $(C(w), |w|)$ is injective,

$$\begin{aligned}\sum_{|u|\leq k} 2^{-|C(wu)|} &\leq \sum_{i=0}^{k} 2^{-|C(w)|+M\log(|w|)-i}\cdot 2^i \cdot (k+1) \\ &= (k+1)^2 2^{-|C(w)|+M\log(|w|)} \\ &\leq 2^{-|C(w)|+M\log(|w|)+2\log(k+1)}.\end{aligned}$$

□

We next present two easy examples of compressors not starting from scratch, including Lempel-Ziv algorithms.

Example 3.2 *For the following polynomial-time compressor lemma 3.1 holds*

1. *An extender, that is, for all* $w, w' \in \{0,1\}^*$

$$w \sqsubseteq w' \Rightarrow C(w) \sqsubseteq C(w').$$

In fact, if C *is an extender, then* C *satifies the common prefix condition in lemma 3.1 and therefore it is a compressor that does not start from scratch.*

2. *Lempel-Ziv data compression algorithm for its two most common variants [16, 17]. Notice that they are not extenders.*

   1. *LZ_{78} version. Let* $w \in \{0,1\}^*$ *and* $w = w_1 w_2 \ldots w_n v$ *where* $w_1, w_2, \ldots, w_n$ *are the phrases obtained by the Lempel-Ziv parsing. Then for* $u \in \{0,1\}^*$*, Lempel-Ziv parsing of* wu *will share the first* $n-1$ *phrases of* w *parsing and may differ in phrase* w_n*. This later phrase will be encoded with* $\lceil \log n + 1\rceil + 1$ *bits, therefore* $\mathrm{LZ}_{78}(w)$ *and* $\mathrm{LZ}_{78}(wu)$ *have a common prefix of length at least* $|\mathrm{LZ}_{78}(w)| - (\lceil \log n + 1\rceil + 1) \geq |\mathrm{LZ}_{78}(w)| - 3\log|w|$.
   2. *LZ_{77} version. Let* $w \in \{0,1\}^*$ *and* $w = w_1 w_2 \ldots w_n v$ *where* $w_1, w_2, \ldots, w_n$ *is the exhaustive history of* w*. Then for* $u \in \{0,1\}^*$*, the exhaustive history of* wu *will share the first* $n-1$ *phrases of* w *parsing and may differ in phrase* w_n*. This later phrase will be encoded with* $2\lceil \log |w|\rceil + 1$ *bits, therefore* $\mathrm{LZ}_{77}(w)$ *and* $\mathrm{LZ}_{77}(wu)$ *have a common prefix of length at least* $|\mathrm{LZ}_{77}(w)| - (2\lceil \log |w|\rceil + 1) \geq |\mathrm{LZ}_{77}(w)| - 3\log|w|$.

Polynomial-time compressors (C, D) that are length increasing in the encoder C and for which we can control, for all w and all $i \geq 0$, the number of strings u satisfying $|C(wu)| = |C(w)| + i$, don't start from scratch. More formally,

Lemma 3.3 *Polynomial-time compressors* (C, D) *that satisfy both of the following conditions don't start from scratch.*

i) For all $w, u \in \{0,1\}^*$, $|C(wu)| \geq |C(w)|$

ii) For all $\epsilon > 0$ *there exist* $c > 0$ *such that for all but finitely many* $w \in \{0,1\}^*$ *there exists* $k \leq c\log(|w|)$ *such that for all* $i \geq 0$

$$N_i = N_i(w,k) = \#\left\{u \in \{0,1\}^{\leq k} \;\middle|\; |C(wu)| = |C(w)| + i\right\} \leq 2^{i+\epsilon k - \log k}$$

Proof. To prove this we assume the worse possible case, that is, in the left hand sum of condition (1) $\sum_{|u|\leq k} 2^{-|C(wu)|}$, we have the maximum possible number of terms with $|C(wu)|$ the smallest possible. This means the we will add $2^{-|C(w)|}$ a total of $(2^{k\epsilon - \log k})$ times, $2^{-|C(w)|-1}$ a total of $(2^{1+k\epsilon-\log k})$ times, and similarly for each value of $|u|$, until we have finally added $2^{k+1}-1$ terms in the left hand sum. Notice that this is possible since

$$\sum_{i=0}^{k} 2^{i+k\epsilon-\log k} \geq 2^{k+1}.$$

Therefore

$$\begin{aligned}
\sum_{|u|\leq k} 2^{-|C(wu)|} &\leq \sum_{i=0}^{k} 2^{i+k\epsilon-\log k} 2^{-(|C(w)|+i)} \\
&= (k+1)2^{\epsilon k - \log k} 2^{-|C(w)|} \\
&\leq 2^{2\epsilon k} 2^{-|C(w)|}.
\end{aligned}$$

□

4 Main Theorem

In the main theorem we obtain an exact characterization of polynomial-time dimension in terms of polynomial-time compressors that don't start from scratch.

Our characterization holds both for the best and worst asymptotic compression ratio, corresponding to p-dimension and strong p-dimension.

We formalize the notion of a.e. (almost everywhere) and i.o. (infinitely often) compressibility for sets of infinite sequences as the asymptotic best (respectively worse) compression ratio.

Definition. For $\alpha \in [0,1]$ and $X \subseteq \mathbf{C}$,

1. X is *α-i.o. polynomial-time compressible* if there is a polynomial-time compressor (C,D) that does not start from scratch and such that for every $A \in X$

$$\liminf_{n} \frac{|C(A[0\ldots n-1])|}{n} \leq \alpha$$

2. X is *α-a.e. polynomial-time compressible* if there is a polynomial-time compressor (C, D) that does not start from scratch and such that for every $A \in X$

$$\limsup_n \frac{|C(A[0 \ldots n-1])|}{n} \leq \alpha$$

Definition. Let $X \subseteq \mathbf{C}$,

1. X is *i.o. polynomial-time incompressible* if, for all $\alpha < 1$, X is not α-i.o. polynomial-time compressible.

2. X is *a.e. polynomial-time incompressible* if, for all $\alpha < 1$, X is not α-a.e. polynomial-time compressible.

We next state our main theorem.

Theorem 4.1 *Let $X \subseteq \mathbf{C}$,*

$$\begin{aligned}
\dim_{\mathrm{p}}(X) &= \inf\{\, \alpha \,|\, X \textit{ is } \alpha\textit{-i.o. polynomial-time compressible}\} \\
\mathrm{Dim}_{\mathrm{p}}(X) &= \inf\{\, \alpha \,|\, X \textit{ is } \alpha\textit{-a.e. polynomial-time compressible}\}
\end{aligned}$$

The proof of theorem 4.1 will be split between sections 5 and 6. In section 5 we transform each gale into a compressor that requires only a time increase of a linear factor. In section 6 we show that compression is an upper bound on dimension.

Hitchcock showed in [12] that p-dimension can be characterized in terms of on-line prediction algorithms, using the well-studied log-loss prediction ratio. Our result can thus be interpreted as a joining bridge between the performance of polynomial-time prediction and compression algorithms, both in the best and the worse case.

5 Compression is at most dimension

We first prove that a dimension upper bound gives a compression upper bound.

Theorem 5.1 *Let $X \in \mathbf{C}$,*

$$\begin{aligned}
\dim_{\mathrm{p}}(X) &\geq \inf\{\, \alpha \,|\, X \textit{ is } \alpha\textit{-i.o. polynomial-time compressible}\}, \\
\mathrm{Dim}_{\mathrm{p}}(X) &\geq \inf\{\, \alpha \,|\, X \textit{ is } \alpha\textit{-a.e. polynomial-time compressible}\}.
\end{aligned}$$

To prove Theorem 5.1 we first transform each polynomial-time gale into a simple version that requires little accuracy. Then we apply a generalization of arithmetic coding [7] to this new gale.

We will use the following result.

Lemma 5.2 *[22] Let d_1 be a martingale. Let $c : \{0,1\}^* \to [0,+\infty)$ be an exactly p-computable function such that for each $w \in \{0,1\}^*$, $|c(w) - d_1(w)| \leq 2^{-|w|}$. Let d_2 be recursively defined as follows*

$$\begin{aligned} d_2(\lambda) &= c(\lambda) + 2 \\ d_2(wb) &= d_2(w) + \frac{c(wb) - c(w\bar{b})}{2} \end{aligned}$$

Then d_2 is an exactly p-computable martingale such that $|d_1(w) - d_2(w)| \leq 4$.

Using the previous lemma we show that very simple gales characterize p-dimension.

Lemma 5.3 *Let $X \subseteq \mathbf{C}$. If $\dim_{\mathrm{p}}(X) = \alpha$ then for all $\epsilon > 0$ there is an $s \in (\alpha, \alpha + \epsilon)$ and an exactly p-computable s-gale d with $X \subseteq S^\infty[d]$ such that for all $w \in \{0,1\}^*$, there exists $m_w, n_w \in \mathbb{N}$ with $n_w \leq |w| + 1$ and*

$$d(w)2^{-|w|s} = m_w 2^{-(n_w + |w|)}$$

Proof. If $\dim_{\mathrm{p}}(X) = \alpha$ then for all $\epsilon > 0$ there is an $s \in (\alpha, \alpha + \epsilon)$ and an exactly p-computable s-gale d', with $d'(\lambda) = 1$, that succeeds on X.
Let d_1 be the martingale $d_1(w) = 2^{(1-s)|w|}d'(w)$ and let $c : \{0,1\}^* \to [0,+\infty)$ be such that $c(w) = m'_w 2^{-n'_w}$ where

$$n'_w = \min\{n \in \mathbb{N} \mid \exists m \text{ s.t. } |m2^{-n} - d_1(w)| < 2^{-|w|}\}$$

$$m'_w = min\{m \in \mathbb{N} \mid |m2^{-n_w} - d_1(w)| < 2^{-|w|}\}$$

Notice that $n'_w \leq |w| + 1$ because within an interval of length $2^{-|w|}$ there exists at at least least one dyadic number $m2^{-n}$ with $n = |w| + 1$. Notice that for the same reason c is exactly p-computable.

Let d_2 obtained from d_1 as in lemma 5.2. There exists $m_w, n_w \in \mathbb{N}$ such that $d_2(w) = m_w 2^{-n_w}$ with $n_w \leq |w| + 1$. We prove this by induction, if $|w| = 0$ then $d_2(\lambda) = 3 = 3 \cdot 2^{-0}$.

$$\begin{aligned} d_2(wb) &= d_2(w) + \frac{c(wb) - c(w\bar{b})}{2} \\ &= m_w 2^{-n_w} + \frac{m'_{wb} 2^{-n'_{wb}} - m'_{w\bar{b}} 2^{-n'_{w\bar{b}}}}{2} \\ &= 2^{-n_w b} m_{wb} \end{aligned}$$

where $n_{wb} = \max\{n_w, n'_{w0} + 1, n'_{w1} + 1\} \leq |w| + 2$.

Using d_2 we define an s-gale d as follows

$$d(w) = 2^{(s-1)|w|} d_2(w).$$

It is then clear that d is exactly p-computable and that for each $w \in \{0,1\}^*$ we have that

(i) By definition, $d(w)2^{-|w|s} = 2^{-|w|}d_2(w) = m_w 2^{-(n_w+|w|)}$ is a dyadic number and $n_w \leq |w| + 1$.

(ii) $|d'(w) - d(w)| = 2^{(s-1)|w|}|d_1(w) - d_2(w)| \leq 2^{(s-1)|w|}4$ so $S^{\infty}[d'] = S^{\infty}[d]$.

□

That is, if $\dim_p(X) < s$, then there exists a p-computable s-gale d as in the previous lemma. We define a polynomial-time compressor that doesn't start from scratch using this s-gale. Roughly speaking, the idea for the encoder C is associate to each $w \in \{0,1\}^*$ an interval of size proportionally related to $d(w)$. By the properties of d, the extreme points of such interval are dyadic rational numbers. By using the following lemma, we encode each interval with a string z. We will take $C(w) = z$.

Lemma 5.4 *Let a, b be dyadic numbers. and let $I = [a, b)$ be an interval of length $r \in [0,1)$, then there exists a string z of length $-\lceil \log r \rceil + 1$ such that $a \leq 0.z < b$ and z can be computed in time polynomial in $|z|$.*

Proof. Let $n \in \mathbb{N}$ be such that $\lceil \log r \rceil = -n$. We divide the interval $[0,1)$ in intervals of length 2^{-n-1}. Notice that each of those intervals is exactly a cylinder C_w with $w \in \{0,1\}^{n+1}$ (see figure 1).

Since

$$\frac{r}{2} \leq \frac{1}{2^{n+1}} = \frac{2^{\lceil \log r \rceil}}{2} < r,$$

we have that at least one of the endpoints of these intervals in inside the interval (a, b). Furthermore, at most two of these interval endpoints can be within (a, b). The largest such endpoint is the $0.z$ we are searching.

We will represent z so that C_z is the interval with left endpoint $0.z$ (see Example 5.5). In order to do this, if $z = z_1 z_2 \ldots z_m$, we have that

$$0.z = z_1 2^{-1} + z_2 2^{-2} + \ldots z_m 2^{-m}.$$

Let us see that we can compute z in polynomial time. To do this, we compute $z = z_1 z_2 \ldots z_{n+1}$ one bit at a time in the following way,

$$z_1 = \begin{cases} 0 & \text{if } b \leq 1/2. \\ 1 & \text{if } b > 1/2. \end{cases}$$

Once we know bits $z_i \ldots z_i$ we define z_{i+1} such that

$$z_{i+1} = \begin{cases} 0 & \text{if } b \leq 0.z_1 \ldots z_i 1. \\ 1 & \text{if } b > 0.z_1 \ldots z_i 1. \end{cases}$$

We will obtain z with $a \leq 0.z < b$ and $|z| = n + 1 = -\lceil \log r \rceil + 1$. □

Example 5.5 *Consider the following situation, $a = 3 \cdot 2^{-2}$ and $b = 7 \cdot 2^{-3}$. In this case $r = b - a = 2^{-3}$ and therefore $\lceil \log r \rceil = -3$.*

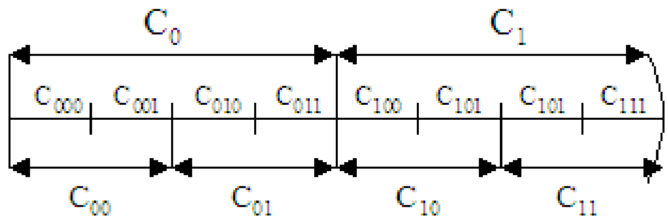

Figure 1: Example of the cylinders placement in $[0, 1)$.

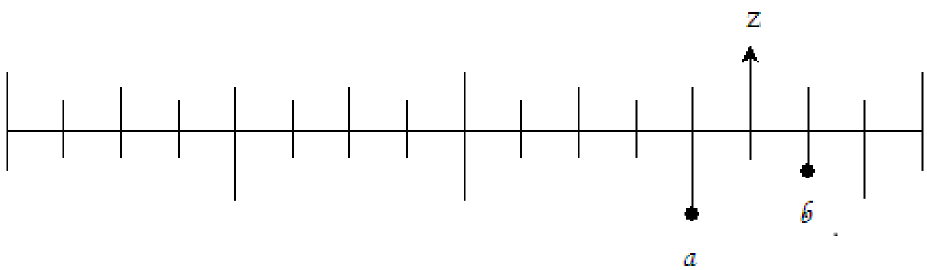

Figure 2: In this particular example, $z = 1101$.

We split interval $[0, 1)$ into length 2^{-4} intervals, as can be seen in figure 2. To compute z bitwise,

$$\begin{aligned}
z_1 &= 1, && \textit{since } b > 1/2. \\
z_2 &= 1, && \textit{since } b > 0.z_1 1 = 3/4. \\
z_3 &= 0, && \textit{since } b \leq 0.z_1 z_2 1 = 7/8. \\
z_4 &= 1, && \textit{since } b > 0.z_1 z_2 z_3 1 = 13/16.
\end{aligned}$$

And therefore $z = 1101$ and $|z| = -\lceil \log r \rceil + 1 = 4$.

Proof of Theorem 5.1. We prove the first inequality; the proof for strong dimension is analogous.

Let X be such that $\dim_{\mathrm{p}}(X) < s$ with s rational, then by Lemma 5.3 there exists d' an exactly p-computable s-gale such that

i) $X \subseteq S^\infty[d']$.

ii) For each $w \in \{0, 1\}^*$, there exist $m_w, n_w \in \mathbb{N}^+$ with $n_w \leq |w| + 1$ such that
$$d'(w)2^{-|w|s} = m_w 2^{-(n_w + |w|)}. \tag{2}$$

Without loss of generality we can assume that $d'(\lambda) = 1$.

We define for each $w \in \{0,1\}^*$,

$$d(w) = 2^{(1-s)|w|} d'(w).$$

d is an exactly p-computable martingale such that,

i) For all $A \in X$, $\quad d(A[0 \ldots n-1]) > 2^{(1-s)n}$ i.o. n

ii) By equation (2), for each $w \in \{0,1\}^*$ there exists $m_w, n_w \in \mathbb{N}$ with $n_w \leq |w| + 1$

$$d(w) = m_w 2^{-n_w}.$$

Let $h : \{0,1\}^* \to \mathbb{R}$ be defined as follows.

$$h(w) := \sum_{|y|=|w|, y<w} d(y) 2^{-|w|}.$$

Notice that $h(w)$ is a dyadic number $m2^{-n}$ with $n \leq 2|w| + 1$, therefore by Lemma 5.4 there is a $z \in \{0,1\}^*$ such that $|z| \leq 2|w| + 2$ and

i) $h(w) \leq 0.z < h(next(w))$, $\quad$ if $w \neq 1^{|w|}$;

ii) $h(w) \leq 0.z < 1$, $\quad$ if $w = 1^{|w|}$.

In fact,

i) If $w \neq 11 \ldots 1$, then

$$l = h(next(w)) - h(w) = d(w) 2^{-|w|}.$$

ii) When $w = 11 \ldots 1$,

$$l = 1 - h(w) = 1 - \sum_{\substack{|y|=|w| \\ y<w}} d(y) 2^{-|w|} = d(w) 2^{-|w|}.$$

Where the las equality if obtained by applying the definition of martingale that establishes

$$\sum_{y \in \{0,1\}^{|w|}} d(y) = 2^{|w|} d(\lambda) = 2^{|w|}.$$

Then in any case

$$l = d(w) 2^{-|w|} = m_w 2^{-n_w - |w|}.$$

And therefore,

$$-\lceil \log l \rceil + 1 = -\lceil \log m_w \rceil + n_w + |w| + 1 \leq 2|w| + 2.$$

Let z_w be the first shortest string such that $h(w) \leq 0.z < h(next(w))$ (see figure 3). We define the encoder as $C(w) = z_w$. It is clear that C can be

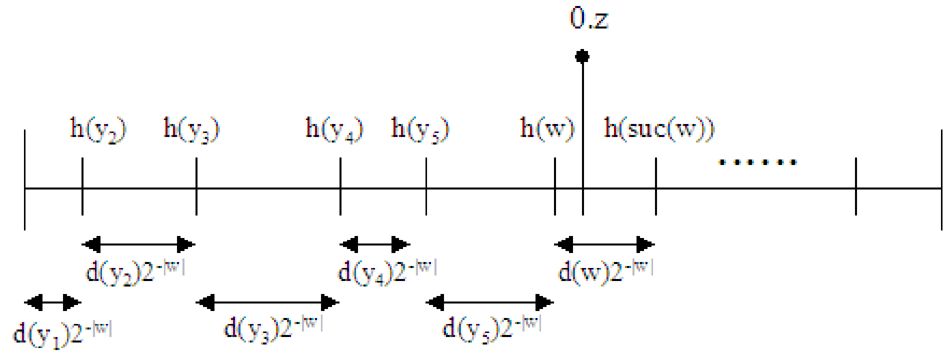

Figure 3: In this example, w is the sixth element of length $|w|$ in lexicographical order.

computed in polynomial time since z_w is computed in time polynomial in the length of z_w (Lemma 5.4) and $|z_w| \leq 2|w| + 2$.

To define our decoder D, let $z \in \{0,1\}^*$ and $n \in \mathbb{N}$, then to generate a string of length n from (z, n), simulate the martingale starting at λ on successively longer strings. Suppose we have generated the string w so far. If $h(w0) \leq 0.z < h(w1)$, then append 0 to w, if $h(w1) \leq 0.z$, then append 1 to w. Continue until $|w| = n$. At the end of this process, we have the string w of length n such that $h(w) \leq 0.z < h(next(w))$.

We next show that the polynomial-time compressor (C, D) does not start from scratch.

Notice that for each w the interval $[h(w), h(next(w)))$ has length exactly $d(w)2^{-|w|}$. Then by lemma 5.4, there is a string z of length $-\lceil \log(2^{-|w|}d(w)) \rceil + 1 \leq |w| - \lceil \log(d(w)) \rceil + 1$ such that $h(w) \leq 0.z < h(next(w))$. So,

$$|z_w| \leq |w| - \lceil \log(d(w)) \rceil + 1.$$

To see that C satisfies condition (1), we will prove that C satisfies the two conditions of remark 3.3.

i) It is clear that for all $w, u \in \{0,1\}^*$, $|C(wu)| \geq |C(w)|$ because the interval $[h(wu), h(next(wu)))$ is included in $[h(w), h(next(w)))$.

ii) Let $\epsilon > 0$, $w \in \{0,1\}^*$, $i \in \mathbb{N}$, $j \in \mathbb{N}$

$$N_i^j = \# \left\{ u \in \{0,1\}^* \;\middle|\; |u| = j \text{ and } |z_{wu}| = |z_w| + i \right\}$$

We have that

$$(N_i^j - 1)2^{-(|z_w|+i)} < d(w)2^{-|w|}$$
$$N_i^j < 1 + d(w)2^{-|w|+|z_w|+i}$$

but since $|z_w| \leq |w| - \lceil \log d(w) \rceil + 1$,

$$N_i^j < 1 + 2^{\log(d(w)) - \lceil \log d(w) \rceil} 2^{i+1} \leq 1 + 2^{i+2}.$$

Let $k \in \mathbb{N}$ and $N_i = \# \left\{ u \in \{0,1\}^{\leq k} \,\middle|\, |C(wu)| = |C(w)| + i \right\}$, then

$$N_i = \sum_{j=0}^{k} N_i^j \leq \sum_{j=0}^{k} 2^{i+3} \leq 2^{i+k\epsilon - \log k}$$

for all but finitely many k.

Finally, let us see that (C, D) compresses X. For all $A \in X$,

$$\begin{aligned} |C(A[0 \ldots n-1])| &= |z_{A[0\ldots n-1]}| \\ &\leq n - \lceil \log(d(A[0 \ldots n-1]) \rceil + 1 \\ &\leq n - \log(2^{(1-s)n}) + 1 \\ &= sn + 1 \end{aligned}$$

□

6 Dimension is at most compression

Next we prove that compressibility is an upper bound on dimension.

Theorem 6.1 *Let $X \in \mathbf{C}$,*

$$\begin{aligned} \dim_{\mathrm{p}}(X) &\leq \inf\{\, \alpha \mid X \textit{ is } \alpha\textit{-i.o. polynomial-time compressible}\} \\ \mathrm{Dim}_{\mathrm{p}}(X) &\leq \inf\{\, \alpha \mid X \textit{ is } \alpha\textit{-a.e. polynomial-time compressible}\} \end{aligned}$$

Proof. We will prove the first inequality since the second one is analogous.

Let α be such that X is α-i.o. polynomial-time compressible and let (C, D) be the (non starting from scratch) polynomial-time compressor witnessing this fact. Let $s > \alpha$ be rational and $\epsilon > 0$ such that $s - \alpha > 2\epsilon$. Let N be such that condition (1) is true for each $w \in \{0,1\}^{\geq N}$. For each of these w, let $k = k(w, \epsilon) = O(\log(|w|))$ be the smallest one such that

$$\sum_{|u| \leq k} 2^{-|C(wu)|} \leq 2^{\epsilon k} 2^{-|C(w)|}$$

Let $w = w_1 \ldots w_n$ with $|w_1| = N$ and $|w_i| = k(w_1 \ldots w_i - 1, \epsilon)$ for $i > 0$.

We define function d as follows

$$\begin{aligned} d(wu) &:= d(w) \tfrac{2^{-|C(wu)|}}{\sum_{|v| \leq k} 2^{-|C(wv)|}} 2^{s|u|} && \text{if } |u| = k(w, \epsilon), \\ d(w\tilde{u}) &:= \textstyle\sum_{\tilde{u} \sqsubseteq u, |u| = k} d(wu) 2^{s(|\tilde{u}| - |u|)} && \text{if } |\tilde{u}| < k(w, \epsilon). \end{aligned}$$

Let us see that d is an s-gale. For that we will distinguish several cases

i) Let $w = w_1 \dots w_n \tilde{u}$, where w_i is as before. We denote $\tilde{w} = w_1 \dots w_n$. Let us consider in this case that $0 < |\tilde{u}| < k(\tilde{w}, \epsilon)$ and $|\tilde{u}| + 1 < k(\tilde{w}, \epsilon)$. Then,

$$\begin{aligned}
[d(w0) + d(w1)]2^{-s} &= [d(\tilde{w}\tilde{u}0) + d(\tilde{w}\tilde{u}1)]2^{-s} \\
&= [\sum_{\substack{\tilde{u}0 \sqsubseteq u \\ |u|=k}} d(\tilde{w}u)2^{s(|\tilde{u}|+1-|u|)} \\
&+ \sum_{\substack{\tilde{u}1 \sqsubseteq u \\ |u|=k}} d(\tilde{w}u)2^{s(|\tilde{u}|+1-|u|)}] \cdot 2^{-s} \\
&= \sum_{\substack{\tilde{u} \sqsubseteq u \\ |u|=k}} d(\tilde{w}u)2^{s(|\tilde{u}|-|u|)} = d(w),
\end{aligned}$$

ii) Let us suppose now that w is exactly of the form $w_1 \dots w_n$. In this case, we have that

$$\begin{aligned}
[d(w0) + d(w1)]2^{-s} &= [\sum_{\substack{0 \sqsubseteq u \\ |u|=k}} d(wu)2^{(1-|u|)s} \\
&+ \sum_{\substack{1 \sqsubseteq u \\ |u|=k}} d(wu)2^{(1-|u|)s}] \cdot 2^{-s} \\
&= 2^{-ks} \sum_{|u|=k} d(wu).
\end{aligned}$$

Since $|u| = k$ we have that

$$d(wu) = d(w)\frac{2^{-|C(wu)|}}{\sum_{|v|\leq k} 2^{-|C(wv)|}}2^{s|u|}$$

and therefore

$$\begin{aligned}
[d(w0) + d(w1)]2^{-s} &= 2^{-ks} \sum_{|u|=k} d(w)\frac{2^{-|C(wu)|}}{\sum_{|v|\leq k} 2^{-|C(wv)|}}2^{s|u|} \\
&= d(w)\frac{\sum_{|u|=k} 2^{-|C(wu)|}}{\sum_{|v|\leq k} 2^{-|C(wv)|}} \\
&\leq d(w).
\end{aligned}$$

iii) For the last case, let us suppose that $w = w_1 \dots w_n \tilde{u}$, where w_i is as before. We denote $\tilde{w} = w_1 \dots w_n$. Let us consider in this case that $0 < |\tilde{u}| < k(\tilde{w}, \epsilon)$ and $|\tilde{u}| + 1 = k(\tilde{w}, \epsilon)$. Then

$$\begin{aligned}
d(w) = d(\tilde{w}\tilde{u}) &= \sum_{\substack{\tilde{u} \sqsubseteq u \\ |u|=k}} d(\tilde{w}u)2^{s(|\tilde{u}|-|u|)} \\
&= [d(w0) + d(w1)]2^{-s}.
\end{aligned}$$

Therefore d is an s-gale. In addition d is polynomial-time computable. In fact the number of additive terms that appear in the definition of d is at most $2^{k(w,\epsilon)+1}$ and since $k(w,\epsilon) = O(\log(|w|))$ we have that the number of terms is polynomial in the input length. Each term in the definition of d is polynomial-time computable.

On the other hand if we expand d definition we have that if $w = w_1 w_2 \dots w_n$ with $|w_1| = N$ and $|w_i| = k(w_1 \dots w_{i-1}, \epsilon)$, then,

$$d(w) = d(w_1) 2^{s(|w|-N)} \prod_{h=1}^{n-1} \frac{2^{-|C(w_1 \dots w_{h+1})|}}{\sum_{|v| \le k(w_1 \dots w_h, \epsilon)} 2^{-|C(w_1 \dots w_h v)|}}.$$

By condition (1),

$$\begin{aligned} d(w) &\ge d(w_1) 2^{(\epsilon - s)N} 2^{|C(w_1)|} 2^{(s-\epsilon)|w|} 2^{-|C(w)|} \\ &\ge a 2^{(s-\epsilon)|w|} 2^{-|C(w)|} \end{aligned}$$

where a is the minimum of

$$d(w_1) 2^{|C(w_1)|} 2^{(\epsilon - s)N}$$

for $w_1 \in \{0,1\}^N$.

Let us see that d succeeds on X. For that let $A \in X$. Then by hypothesis

$$\liminf_n \frac{|C(A[0 \dots n-1])|}{n} \le \alpha$$

thus there is a sequence of natural numbers $(b_n)_{n \in \mathbb{N}}$ that satisfies

$$\lim_n \frac{|C(A[0 \dots b_n - 1])|}{b_n} \le \alpha.$$

I.e., there are infinitely many n's for which

$$|C(A[0 \dots b_n - 1])| \le b_n(\alpha + \epsilon). \tag{3}$$

Let $(a_n)_{n \in \mathbb{N}}$ be defined as

$$\begin{aligned} a_1 &= k_0 = N, \\ a_{i+1} &= a_i + k_i \qquad \text{for } i > 1, \end{aligned}$$

where $k_i = k(A[0 \dots a_i - 1], \epsilon)$, that is, $k_i = O(\log a_i)$.

Then
$$d(A[0 \dots a_i - 1]) \ge a 2^{(s-\epsilon)a_i} 2^{-|C(A[0 \dots a_i - 1])|}.$$

For each n, let $m \in \mathbb{N}$ be such that $a_m < b_n \le a_{m+1}$ (see figure 4).

Since $b_n - a_m \le k_m$ we have that by condition (1)

$$2^{-|C(A[0 \dots b_n - 1])|} \le 2^{k_m \epsilon} 2^{-|C(A[0 \dots a_m - 1])|},$$

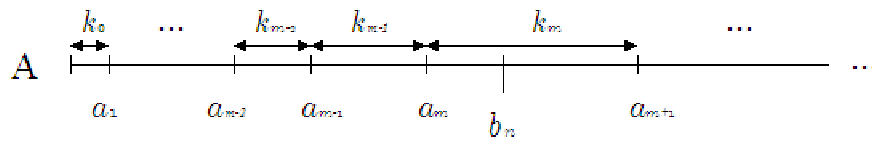

Figure 4: Position of a_i's in relation with b_n and k_i within sequence A.

and therefore

$$|C(A[0\dots a_m - 1])| \leq |C(A[0\dots b_n - 1])| + k_m\epsilon.$$

Then for all but finitely many n,

$$\begin{aligned}
d(A[0\dots a_m - 1]) &\geq a2^{(s-\epsilon)a_m}2^{-|C(A[0\dots a_m-1])|} \\
&\geq a2^{(s-\epsilon)a_m}2^{-(|C(A[0\dots b_n-1])|-k_m\epsilon} \\
&\geq a2^{(s-\epsilon)a_m}2^{-b_n\alpha-b_n\epsilon-k_m\epsilon} \\
&= a2^{(s-\alpha-2\epsilon)a_m+(\alpha+\epsilon)(a_m-b_n)-k_m\epsilon} \\
&\geq a2^{(s-\alpha-2\epsilon)a_m-k_m(2\epsilon+\alpha)}
\end{aligned}$$

And d succeeds on X since i) $k_m = O(\log a_m)$ and ii) $s - \alpha > 2\epsilon$. We have thus proven that $\dim_{\mathrm{p}}(X) \leq s$. Since we took and arbitrary $\epsilon > 0$ we can take $s > \alpha$ arbitrarily and conclude the result.

□

Notice that in the last proof we didn't need the decoder to be polynomial-time computable. This gives us the following stronger characterization and corollary.

Definition. *A polynomial-time encoder C that does not start from scratch* is a polynomial-time computable and injective function $C : \{0,1\}^* \to \{0,1\}^*$ that satisfies condition (1).

Definition. For $\alpha \in [0,1]$ and $X \subseteq \mathbf{C}$,

1. X is *α-i.o. one-way polynomial-time compressible* if there is a polynomial-time encoder C that does not start from scratch and such that for every $A \in X$
$$\liminf_n \frac{|C(A[0\dots n-1])|}{n} \leq \alpha$$

2. X is *α-a.e. one-way polynomial-time compressible* if there is a polynomial-time encoder C that does not start from scratch and such that for every $A \in X$
$$\limsup_n \frac{|C(A[0\dots n-1])|}{n} \leq \alpha$$

Theorem 6.2 *Let $X \subseteq \mathbf{C}$,*

$$\begin{aligned}\dim_{\mathrm{p}}(X) &= \inf\{\,\alpha \mid X \textit{ is } \alpha\textit{-i.o. one-way polynomial-time compressible}\} \\ \mathrm{Dim}_{\mathrm{p}}(X) &= \inf\{\,\alpha \mid X \textit{ is } \alpha\textit{-a.e. one-way polynomial-time compressible}\}\end{aligned}$$

Corollary 6.3 *Let C be a polynomial-time encoder that does not start from scratch. Then there exist (C', D') a polynomial-time compressor that does not start from scratch and such that for every $A \in \mathbf{C}$*

$$\liminf_n \frac{C'(A[0\ldots n-1])}{n} \le \liminf_n \frac{C(A[0\ldots n-1])}{n}$$

$$\limsup_n \frac{C'(A[0\ldots n-1])}{n} \le \limsup_n \frac{C(A[0\ldots n-1])}{n}$$

Invertibility phenomena similar to this last corollary have been investigated for other families of compressors, for instance in [15] Huffman works on inversion of finite-state compressors.

7 Applications of the Main Result

In this section we obtain interesting consequences of our characterization for the polynomial-time compressibility of complete and autoreducible sets from previously known p-dimension results.

Notice that in this section we identify each language A with its characteristic sequence χ_A, therefore compressibility of a class always means compressibility of the corresponding characteristic sequences. Recall that E= DTIME($2^{O(n)}$).

We start by showing that no polynomial-time compressor works on all many-one complete sets.

Theorem 7.1 *The class of polynomial-time many-one complete sets for E is i.o. polynomial-time incompressible.*

Proof. Ambos-Spies et al. prove in [2] that the class has p-dimension 1.

□

Next we consider $\deg^{\mathrm{p}}_{\mathrm{m}}(A)$, the class of sets that are equivalent to A by $\le^{\mathrm{P}}_{\mathrm{m}}$-reductions. The compression ratio of $\deg^{\mathrm{p}}_{\mathrm{m}}(A)$ and $\deg^{\mathrm{p}}_{\mathrm{m}}(B)$, for $A \le^{\mathrm{P}}_{\mathrm{m}} B$, is related by the following theorem.

Theorem 7.2 *Let A, B be sets in E such that $A \le^{\mathrm{P}}_{\mathrm{m}} B$, then*

1. *The i.o. p-compression ratio of* $\deg^{\mathrm{p}}_{\mathrm{m}}(A)$ *is at most the i.o. p-compression ratio of* $\deg^{\mathrm{p}}_{\mathrm{m}}(B)$.
2. *The a.e. p-compression ratio of* $\deg^{\mathrm{p}}_{\mathrm{m}}(A)$ *is at most the a.e. p-compression ratio of* $\deg^{\mathrm{p}}_{\mathrm{m}}(B)$.

Proof. Ambos-Spies et al. prove 1. in [2] for p-dimension. Athreya et al. prove in [3] the strong dimension result for 2. □

We next consider the property of autoreducibility. A set A is autoreducible if A can be decided by using A as an oracle but without asking query x on input x. We obtain incompressibility results both in the case of polynomial-time many-one autoreducibility and for the *complement* of i.o. p-Turing autoreducible sets. Therefore for each polynomial-time bound there are i.o. incompressible sets that are $\leq_{\mathrm{m}}^{\mathrm{P}}$-autoreducible and other that are not even i.o. $\leq_{\mathrm{T}}^{\mathrm{P}}$-autoreducible.

Theorem 7.3 *The class of polynomial-time many-one autoreducible sets are i.o. polynomial-time incompressible.*

Proof. Ambos-Spies et al. prove in [2] that the class has p-dimension 1. □

Theorem 7.4 *The class of sets that are NOT i.o. polynomial-time Turing autoreducible are i.o. polynomial-time incompressible.*

Proof. Beigel et al. prove in [4] that the class has p-dimension 1. □

We next show that there exist polynomial-time many-one degrees with every possible value for both a.e. and i.o. compressibility.

Theorem 7.5 *Let x, y be computable reals such that $0 \leq x \leq y \leq 1$. Then there is a set A in E such that the i.o. p-compression ratio of $\deg_{\mathrm{m}}^{\mathrm{p}}(A)$ is x and the a.e. p-compression ratio of $\deg_{\mathrm{m}}^{\mathrm{p}}(A)$ is y.*

Proof. Athreya et al. prove in [3] the result for p-dimension and strong p-dimension. □

This last theorem includes the extreme case for which the i.o. compression ratio is 0 whereas the a.e. ratio is 1.

Finally, the hypothesis "NP has positive p-dimension" can be interpreted in terms of incompressibility. This hypothesis has interesting consequences on the approximation algorithms for MAX3SAT.

Theorem 7.6 *If for some $\alpha > 0$ NP is not α-i.o-compressible in polynomial-time then any approximation algorithm $\mathcal{A}$ for MAX3SAT must satisfy at least one of the following*

1. *For some $\delta > 0$, $\mathcal{A}$ uses time at least 2^{n^δ}*
2. *For all $\epsilon > 0$, $\mathcal{A}$ has performance ratio less than $7/8 + \epsilon$ (that is, $\mathcal{A}(x) < (7/8 + \epsilon) \cdot MAX3SAT(x)$) on an exponentially dense set of satisfiable instances.*

Proof. Hitchcock proves in [10] that the consequence follows from NP having positive p-dimension. □

References

[1] P. Albert, E. Mayordomo, and P. Moser. Bounded pushdown dimension vs lempel ziv information density. Technical Report TR07-051, ECCC: Electronic Coloquium on Computational Complexity, 2007.

[2] K. Ambos-Spies, W. Merkle, J. Reimann, and F. Stephan. Hausdorff dimension in exponential time. In *Proceedings of the 16th IEEE Conference on Computational Complexity*, pages 210–217, 2001.

[3] K. B. Athreya, J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. Effective strong dimension in algorithmic information and computational complexity. *SIAM Journal on Computing*, 37:671–705, 2007.

[4] R. Beigel, L. Fortnow, and F. Stephan. Infinitely-often autoreducible sets. In *Proceedings of the 14th Annual International Symposium on Algorithms and Computation*, volume 2906 of *Lecture Notes in Computer Science*, pages 98–107. Springer-Verlag, 2003.

[5] H. Buhrman and L. Longpré. Compressibility and resource bounded measure. *SIAM Journal on Computing*, 31(3):876–886, 2002.

[6] J. Cai and J. Hartmanis. On Hausdorff and topological dimensions of the Kolmogorov complexity of the real line. *Journal of Computer and Systems Sciences*, 49:605–619, 1994.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, N.Y., 1991.

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

[9] R. Downey and D. Hirschfeldt. *Algorithmic randomness and complexity.* Springer-Verlag, 2010.

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

[11] J. M. Hitchcock. *Effective Fractal Dimension: Foundations and Applications.* PhD thesis, Iowa State University, 2003.

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

[13] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. The fractal geometry of complexity classes. *SIGACT News Complexity Theory Column*, 36:24–38, 2005.

[14] J. M. Hitchcock and N. V. Vinodchandran. Dimension, entropy rates, and compression. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, pages 174–183, 2004.

[15] D. A. Huffman. Canonical forms for information-lossless finite-state logical machines. *IRE Trans. Circuit Theory CT-6 (Special Supplement)*, pages 41–59, 1959. Also available in E.F. Moore (ed.), Sequential Machine: Selected Papers, Addison-Wesley, 1964, pages 866-871.

[16] A. Lempel and J. Ziv. A universal algortihm for sequential data compression. *IEEE Transaction on Information Theory*, 23:337–343, 1977.

[17] A. Lempel and J. Ziv. Compression of individual sequences via variable rate coding. *IEEE Transaction on Information Theory*, 24:530–536, 1978.

[18] J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–254. Springer-Verlag, 1997.

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

[20] J. H. Lutz. The dimensions of individual strings and sequences. *Information and Computation*, 187:49–79, 2003.

[21] J. H. Lutz. Effective fractal dimensions. *Mathematical Logic Quarterly*, 51:62–72, 2005.

[22] E. Mayordomo. *Contributions to the study of resource-bounded measure.* PhD thesis, Universitat Politècnica de Catalunya, 1994.

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

[24] E. Mayordomo. Effective fractal dimension in algorithmic information theory. In *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 259–285. Springer-Verlag, 2008.

[25] B. Ya. Ryabko. Coding of combinatorial sources and hausdorff dimension. *Soviets Mathematics Doklady*, 30:219–222, 1984.

[26] B. Ya. Ryabko. Noiseless coding of combinatorial sources. *Problems of Information Transmission*, 22:170–179, 1986.

[27] L. Staiger. Kolmogorov complexity and Hausdorff dimension. *Information and Computation*, 103:159–94, 1993.

[28] L. Staiger. A tight upper bound on Kolmogorov complexity and uniformly optimal prediction. *Theory of Computing Systems*, 31:215–29, 1998.