TESIS DE LA UNIVERSIDAD DE ZARAGOZA

2011

32

María López Valdés

Aplicaciones de la dimensión efectiva a la complejidad computacional y a los algoritmos de comprensión de datos

Departamento

Informática e Ingeniería de Sistemas

Director/es

Mayordomo Cámara, Elvira

ISSN 2254-7606

Prensas de la Universidad Universidad Zaragoza

Universidad Zaragoza
1542

Tesis Doctoral

APLICACIONES DE LA DIMENSIÓN EFECTIVA A LA COMPLEJIDAD COMPUTACIONAL Y A LOS ALGORITMOS DE COMPRENSIÓN DE DATOS

Autor

María López Valdés

Director/es

Mayordomo Cámara, Elvira

UNIVERSIDAD DE ZARAGOZA

Informática e Ingeniería de Sistemas

2011

Tesis Doctoral

Aplicaciones de la Dimensión Efectiva a la Complejidad Computacional y a los Algoritmos de Compresión de Datos

María López Valdés

Directora: Elvira Mayordomo Cámara

Grupo de Ingeniería de Sistemas de Eventos Discretos (GISED).
Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior - Universidad de Zaragoza

Julio 2011

Índice de contenidos

Agradecimientos

Cuando pienso en esta tesis, no puedo dejar de pensar en lo mucho que ha cambiado mi vida y lo mucho que he cambiado yo desde que la empecé. Muchas personas me han ayudado en esta tesis, pero... seamos sinceros, no sólo me han ayudado a terminar una tesis, me han ayudado a convertirme en lo que soy ahora.

Intentando seguir un cierto orden cronólogico, me gustaría agradecer esta tesis:

- A mis padres y hermanas, por enseñarme a ser curiosa, inquieta y con ganas de aprender en todo momento.
- A mis compañeros del departamento de matemáticas, muy especialmente al área de análisis matemático, que me metieron el gusanillo de la investigación.
- A Elvira Mayordomo, mi directora de tesis, que me ha ofrecido la oportunidad de empezar y acabar esta tesis y que ha tenido la paciencia para tratar conmigo, ¡algo que no siempre es fácil!
- A mis compañeros del departamento de informática e ingeniería de sistemas, muy espacialmente a los "viejos" diisasters y al GISED, por acogerme y hacerme sentir muy orgullosa de ser parte de ellos.
- A todos los miembros de MOISES y SESAAME, proyectos en los que hemos participado junto con la Universidad de Málaga y la UPC, por esos workshops que me permitieron ampliar mi visión.
- A Jack Lutz y demás personas con las que tuve la oportunidad de trabajar en Iowa State University (USA). Con ellos aprendí lo amplias que pueden a llegar a ser las aplicaciones de dimensión.
- A Javier Mínguez, al que engatusé para que se casara conmigo en mitad de la tesis y que me ha apoyado y dado consejo de forma continúa desde entonces.

- A Jörg Flum y demás personas con las que tuve la oportunidad de trabajar en la Universidad de Freiburg (Alemania). Ellos me enseñaron mucha lógica.
- A Ricard Gavaldà por sus discusiones sobre complejidad y learning en la UPC.
- A Adrián Mínguez, que nació hace 16 meses y, junto con Javier, ha llenado del todo mi vida.
- A los chicos de BitBrain, a los que no siempre les he podido dedicar todo el tiempo que merecían.

¡Muchas gracias a todos!

Con esta tesis cierro un capítulo de mi vida, un capítulo que no siempre ha sido fácil pero un capítulo que sin duda volvería a escribir.

Abstract

During the last decades, the development and use of computer technology has experimented unbelievable progress. Within this pace of development, important computer applications become obsolete due to new resources or to the development of ingenious solutions. If in addition to this, we were able to generate results to understand the basis of this technology, the development could be much more structured and beneficial. This is the main interest of theoretical computer science: to build a formal layer to ground this work in order to facilitate the progress of knowledge, provide an overview of the problems, and an insight into future solutions that may be more appropriate.

In theoretical computer science there are many areas of study such as:

- The theory of computability: the study of whether a problem can be solved by an automatic procedure.

- The theory of computational complexity: the study of whether a problem can be solved efficiently.

- The theory of computational learning theory: the study of the difficulty of building a machine to be capable to learn to find solutions to a problem.

- The theory of information: the study of the intrinsic complexity of the sequences (eg, solutions of a problem) as well as what can be compressed.

In the search for new knowledge, a development of many formal tools has occurred in each of these areas. For example, Lutz introduced in 2000 effective dimension that initially aimed to achieve results in the area of computational complexity. Now, between the different areas there are strong connections and therefore it seems logical that there are also connections between the different formal tools that are used. Finding these connections will allow to transfer results from an area to another and to gain a deeper understanding of the various problems that can be solved with a current computer or even with a computer that can be developed in the future.

This thesis has studied the relationship between effective dimension and Kolmogorov complexity, compression ratios or performance learning algorithms. These results have allowed results of computational learning complexity via information theory and vice versa. More specifically:

- Dimension and Kolmogorov complexity: Ryabko, Staiger, Cai and Hartmanis were the first researchers to study the relationship between classical Haussdorf dimension and Kolmogorov complexity. With the development of the constructive version of Hausdorff dimension, Mayordomo and Lutz showed that for all $X \subseteq \mathbf{C}$ there is a full characterization:

$$\mathrm{cdim}(X) = \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

In the cases of computable dimension and pspace dimension, Hitchcock proved that a complete characterization is also possible. In this case, it is necessary to use the version of Kolmogorov complexity with space-bounded resources:

$$\begin{aligned} \dim_{\mathrm{comp}}(X) &= \inf_{\mathrm{s} \in \mathrm{comp}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \\ \dim_{\mathrm{pspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pspace}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \end{aligned}$$

In this thesis, we generalize these results to scaled dimension (a refinement of effective dimension) and we obtain:

$$\begin{aligned} \mathrm{cdim}_g(X) &= \sup_{A \in X} \liminf_{n \to \infty} f_g^{n+1}(K(A[0..n-1])). \\ \dim^{(k)}_{\mathrm{comp}}(X) &= \inf_{\mathrm{s} \in \mathrm{comp}} \sup_{A \in X} \liminf_{n \to \infty} f_{(k)}^{n+1}(KS^{\mathrm{s}}(A[0..n-1])), \qquad \forall k \in \mathbb{Z}. \\ \dim^{(i)}_{\mathrm{pjspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pjspace}} \sup_{A \in X} \liminf_{n \to \infty} f_{(i)}^{n+1}(KS^{\mathrm{s}}(A[0..n-1])), \qquad \forall i, j \in \mathbb{N},\ i \leq j. \\ \dim^{(-i)}_{\mathrm{pjspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pjspace}} \sup_{A \in X} \liminf_{n \to \infty} f_{(i-)}^{n+1}(KS^{\mathrm{s}}(A[0..n-1])), \qquad \forall i, j \in \mathbb{N},\ i \leq j. \end{aligned}$$

where $f_g^{|w|+1}$ is the inverse function of g. To prove these results, scaled dimension for finite sequences, of independent interest, has been developed. Intuitively, these results show a close relationship between the size of a class of problems and the maximum compressibility achievable for each of these problems, even in the case of a measurement size (scaled dimension) that is very tight to the specific type of problems. Furthermore, these results are used to study the behavior of P/Poly-Turing reductions in ESPACE.

- Dimension and compression: The relationship between polynomial time dimension and Kolmogorov complexity does not seem to be clear, for that reason, this thesis uses the usual compression algorithm for finite sequences to characterize the dimension in polynomial time. With this new characterization, several known results for polynomial time dimension can be interpreted as compression results. For example, a class of languages with p-dimension 1 cannot be compressed (i.o.) in more than a sublinear quantity. In this way, results are obtained on the compressibility of complete and autoreductibles languages, two of the types of problems of most interest in computational complexity.

- Dimension and learning: In this thesis we study the relationships between dimensions and different learning models. In relation to on-line learning , we obtain upper bound of the polynomial-time dimension of class of concepts that can be learned by on-line algorithms in exponential time and with $\alpha 2^n$ errors. We also show that this upper bound is optimal. In relation to PAC learning, we show that the pspace-dimension of classes of concepts that are PAC-learnable is zero. In addition to this, we prove that it is also zero for classes of concepts that can be learned using a query based algortihm. These results have provided hypothesis implying non-learnability or complexity of the representations needed to learn a concept.

Resumen

Durante las últimas décadas, la tecnología de las computadoras y el uso que de ellas se hace ha progresado de forma increíble. Dentro de este ritmo de desarrollo, importantes aplicaciones en informática quedan obsoletas gracias a los nuevos recursos o a la aparición de soluciones más ingeniosas. Ahora bien, si además de generar resultados fuéramos capaces de entender el fundamento de estos resultados, el desarrollo podría ser mucho más estructurado y beneficioso. Este es el principal interés de la informática teórica: el sentar las bases formales de la informática para facilitar así un avance de conocimiento, proporcionar una visión global de los problemas y una intuición sobre futuras soluciones que puedan ser más adecuadas.

Dentro de la informática teórica existen numerosas áreas de estudio como por ejemplo:

- La teoría de la computabilidad: estudia si un problema puede resolverse con un procedimiento automático.
- La teoría de la complejidad computacional: estudia si un problema se puede resolver de un modo eficiente.
- La teoría de aprendizaje computacional: estudia la dificultad de que una máquina sea capaz de aprender a encontrar las soluciones de un determinado problema.
- La teoría de la información: estudia la complejidad intrínseca de las secuencias (por ejemplo, de las soluciones de un determinado problema) así como lo que se pueden comprimir.

En la búsqueda de nuevos conocimientos en cada una de estas áreas se han desarrollado numerosas herramientas formales. Por ejemplo, Lutz introdujo en el año 2000 la dimensión efectiva orientada inicialmente a obtener resultados en el área de la complejidad computacional. Ahora bien, entre las diversas áreas existen fuertes conexiones y por lo tanto, parece lógico que también existan conexiones entre las distintas herramientas formales que se utilizan. Encontrar estas conexiones permite trasladar resultados de unas áreas a otras y obtener un conocimiento mucho más profundo de los diversos problemas que pueden ser resueltos con un computador actual o incluso con un computador que pueda ser desarrollado en un futuro.

En esta tesis se han estudiado las relaciones existentes entre la dimensión efectiva y la complejidad de Kolmogorov, los ratios de compresión o el rendimiento en algoritmos de aprendizaje. Esto ha permitido trasladar resultados de complejidad computacional a aprendizaje computacional o teoría de la información y viceversa, estableciendo nuevas cotas superiores e inferiores de complejidad para problemas y clases de problemas cuya dimensión se conoce, así como nuevos resultados sobre el tamaño de clases en función de su complejidad. Más concretamente:

- Dimensión y complejidad de Kolmogorov: Ryabko, Staiger, Cai y Hartmanis fueron los primeros en estudiar la relación entre dimensión clásica de Hausdorff y complejidad de Kolmogorov. Con el desarrollo de la versión constructiva de la dimensión de Hausdorff, Mayordomo y Lutz demostraron que, para todo $X \subseteq \mathbf{C}$, se conseguía una caracterización completa:

$$\mathrm{cdim}(X) = \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

  En cuanto a los casos de dimensión calculable y dimensión en espacio polinómico, Hitchcock demostró que también es posible una caracterización completa. En este caso es necesario utilizar la versión de complejidad de Kolmogorov con recursos de espacio acotados:

$$\begin{aligned} \dim_{\mathrm{comp}}(X) &= \inf_{\mathrm{s} \in \mathrm{comp}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \\ \dim_{\mathrm{pspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pspace}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \end{aligned}$$

  En esta tesis se generalizan todos estos resultados para la dimensión con escala (un refinamiento de la dimensión efectiva) obteniendo que:

$$\begin{aligned} \mathrm{cdim}_g(X) &= \sup_{A \in X} \liminf_{n \to \infty} f_g^{n+1}(K(A[0..n-1])). \\ \dim^{(k)}_{\mathrm{comp}}(X) &= \inf_{\mathrm{s} \in \mathrm{comp}} \sup_{A \in X} \liminf_{n \to \infty} f^{n+1}_{(k)}(KS^{\mathrm{s}}(A[0..n-1])), && \forall k \in \mathbb{Z}. \\ \dim^{(i)}_{\mathrm{pjspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pjspace}} \sup_{A \in X} \liminf_{n \to \infty} f^{n+1}_{(i)}(KS^{\mathrm{s}}(A[0..n-1])), && \forall i,j \in \mathbb{N},\ i \leq j. \\ \dim^{(-i)}_{\mathrm{pjspace}}(X) &= \inf_{\mathrm{s} \in \mathrm{pjspace}} \sup_{A \in X} \liminf_{n \to \infty} f^{n+1}_{(i-)}(KS^{\mathrm{s}}(A[0..n-1])), && \forall i,j \in \mathbb{N},\ i \leq j. \end{aligned}$$

  donde $f_g^{|w|+1}$ es la función inversa de g. Para demostrar estos resultados se ha desarrollado la dimensión con escala para secuencias finitas, de interés independiente.

  Intuitivamente, estos resultados presentan una estrecha relación entre el tamaño de una clase de problemas y la compresibilidad máxima alcanzable para cada uno de estos problemas, aún en el caso de una medida de tamaño muy ajustada al tipo concreto de problemas (dimensión con escala). Además, estos resultados se utilizan para estudiar el comportamiento de las reducciones P/poly-Turing en la clase ESPACE.

- Dimensión y compresión: La relación entre dimensión en tiempo polinómico y complejidad de Kolmogorov no parece ser clara, por ese motivo, en esta tesis se utiliza la noción usual de algoritmo de compresión para secuencias finitas para caracterizar la dimensión en tiempo polinómico. Gracias a esta nueva caracterización, varios resultados conocidos para dimensión en tiempo polinómico se pueden interpretar como resultados de compresión. Por ejemplo, los lenguajes de una clase de p-dimensión 1 no pueden comprimirse (i.o.) en más que una cantidad sublineal. Así se obtienen resultados sobre la compresibilidad de lenguajes completos y autoreducibles, dos de los tipos de problemas de más interés en complejidad computacional

- Dimensión y aprendizaje: En esta tesis se estudian la relación de la dimensión con distintos modelos de aprendizaje. En relación con el aprendizaje on-line se obtiene una cota superior de la dimensión en tiempo polinómico de clases de conceptos que pueden aprenderse con algoritmos on-line en tiempo exponencial y con $\alpha 2^n$ errores. Además se demuestra que esta cota superior es óptima. En relación con el aprendizaje PAC, se demuestra que la dimensión en espacio polinómico de clases de conceptos que son PAC-aprendibles es cero. Igualmente se demuestra que es cero para clases de conceptos que pueden aprenderse con un algoritmo basado en preguntas. Estos resultados han permitido obtener hipótesis que implican el no aprendizaje o la complejidad de las representaciones necesarias para aprender un concepto.

Capítulo 1

Introducción y preliminares

1.1. Introducción

Durante las últimas décadas, la tecnología de las computadoras y el uso que de ellas se hace ha progresado de forma increíble. Simplemente basta pensar en lo lentos y primitivos que pueden parecer a día de hoy los ordenadores que sólo algunos afortunados podían permitirse en los años 80 frente a los ordenadores personales actuales que aproximadamente el 70 % de los españoles poseemos en nuestros hogares.

Dentro de este ritmo de desarrollo, importantes aplicaciones en informática quedan obsoletas gracias a los nuevos recursos o a la aparición de soluciones más ingeniosas, que hace que cada día poseamos mejores aplicaciones. Ahora bien, si además de generar resultados fuéramos capaces de entender el fundamento de estos resultados, el desarrollo podría ser mucho más estructurado y beneficioso. Este es el principal interés de la informática teórica, el sentar las bases formales de la informática para facilitar así un avance de conocimiento, una visión global de los problemas y una intuición sobre futuras soluciones que puedan ser más adecuadas.

Quizá una pregunta especialmente importante para este propósito es conocer qué tareas se pueden realizar con un procedimiento automático. Aunque cualquier persona puede entender lo que esta pregunta significa, proporcionar un marco teórico adecuado para trabajar en la respuesta a esta pregunta no resulta algo trivial. Para empezar, no resulta evidente dar una definición formal de lo que significa procedimiento automático.

Ya en los años 30, antes de que existieran los ordenadores tal como los conocemos hoy en día, la comunidad matemática empezó a preocuparse por dar una definición de procedimiento automático que fuera consistente con la idea intuitiva que todos tenemos. De este modo, diversos investigadores propusieron definiciones de lo que significaba que una tarea pudiera resolverse con un procedimiento automático. Por ejemplo, Turing mediante su famosa máquina de Turing [91, 92] (fig. 1.1) o Church mediante el Lambda-Cálculo [18, 19]. Este fue el nacimiento de la Teoría de la Computabilidad,

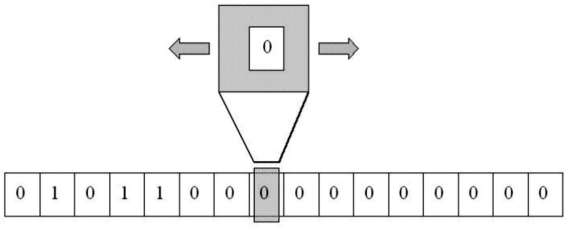

Figura 1.1: Representación de una Maquina de Turing con una cinta y un conjunto de estados. La máquina accede a la información de la cinta a través de una cabeza lectora/escritora. La cabeza lee la celda en la que se encuentra y, dependiendo del símbolo leído y del estado, escribe un nuevo símbolo, se mueve a la izquierda o derecha y cambia de estado.

a finales de los años 30 y, con el tiempo, se ha demostrado la importancia de este campo. A día de hoy, aún con ordenadores cada vez más potentes, la noción de computabilidad sigue asociada a las máquinas de Turing gracias a la Tesis de Church: "Todo algoritmo o procedimiento efectivo es Turing-calculable". Esta tesis nos dice que, en particular, todo lo que se puede calcular con un ordenador hoy en día se puede calcular con una máquina de Turing.

Así pues, dentro de la informática teórica, la Teoría de la Computabilidad nos fija los límites de lo que puede o no ser calculado mediante un ordenador (o mediante cualquier otro procedimiento automático), con la ventaja de que, tratándose de un marco puramente formal, no es necesario utilizar ningún aparato físico real para encontrar dichos límites.

Una vez conocidos los límites de lo que se puede y no calcular, la siguiente pregunta importante es la de conocer la eficiencia de este cálculo: no es lo mismo que se necesiten años y miles de ordenadores trabajando en paralelo para calcular algo a que se necesiten unos simples segundos en un ordenador convencional. Así pues, ¿qué tareas pueden realizarse con un procedimiento automático de un modo efectivo?

En este sentido, la Teoría de la Complejidad Computacional proporciona el marco teórico adecuado para tratar esta pregunta. Más concretamente, la Teoría de la Complejidad Computacional se centra en el estudio de la complejidad intrínseca de las tareas calculables. Por ejemplo, utilizando nociones y herramientas propias de Complejidad Computacional es posible demostrar que un conjunto de tareas pueden resolverse en tiempo polinómico, sin necesidad de encontrar el algoritmo concreto que las resuelve.

En muchos casos, la investigación en Complejidad Computacional tiende a fijar los recursos de cálculo y estudiar las tareas que pueden resolverse dentro de esos límites (por ejemplo, la clase P no es mas que el conjunto de tareas que pueden resolverse en tiempo polinómico). De este modo, surgen de manera natural numerosos conceptos, como por ejemplo:

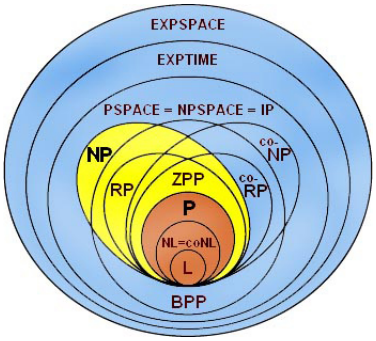

Figura 1.2: Esta figura muestran las relaciones conocidas (hasta el momento) entre diversas clases de Complejidad, como por ejemplo P$\subseteq$ NP.

i) *Clases de complejidad*: conjuntos de tareas calculables agrupadas según los recursos de cálculo necesarios para resolverlas. Estos recursos incluyen cotas en tiempo o memoria, y utilizar máquinas deterministas, no deterministas, probabilistas, circuitos booleanos ....

ii) *Reducciones y problemas completos*: conceptos introducidos como herramientas para comparar la complejidad de problemas específicos. Intuitivamente, que un problema se reduzca a otro, significa que es más fácil de resolver, siendo los problemas completos aquellos que son más difíciles.

Como objetivos fundamentales de la Complejidad Computacional está el encontrar conexiones entre las clases de complejidad (fig. 1.2) y otros conceptos, lo cual produce un avance de conocimiento muy importante en los fundamentos teóricos de la informática.

Una herramienta que ha sido especialmente relevante en este sentido ha sido la medida de recursos acotados [62], que no es más que una adaptación a la medida de Lebesgue utilizada en matemática clásica. Básicamente, las medidas de recursos acotados proporcionan una manera eficiente de medir con respecto a los recursos de cálculo que se estén acotando.

Como ya hemos dicho, un problema importante en Complejidad Computacional es saber cómo se relacionan las clases de complejidad. Un ejemplo sería el saber si dos clases son iguales o bien si una clase está incluida en otra. Únicamente la respuesta positiva a estas preguntas proporciona información realmente valiosa. Por el contrario, una respuesta negativa (saber que una clase no es igual a otra) lo único que asegura es la existencia de un problema que no está en las dos clases, pero eso no significa que las clases sean muy distintas, que una clase tenga muchos más elementos que otra, o que la mayoría de los problemas no pertenezcan a ambas clases. Gracias a las medidas de recursos acotados se puede obtener este tipo de información cuantitativa.

Las medidas de recursos acotados fueron introducidas por Lutz en [62] y su motivación fue precisamente el obtener resultados que proporcionara más información que los de existencia. Es decir, en

vez de conseguir resultados del tipo "existe un problema que está en Y y no está en X" , las medidas de recursos acotados sirven para obtener resultados del tipo "la mayoría de los problemas en Y no están en X", expresado formalmente como "X tiene medida 0 en Y". Además, con el trabajo de numerosos investigadores, las medidas de recursos acotados también han resultado ser una herramienta útil a la hora de obtener resultados relacionados por ejemplo con bi-immunidad, *complexity cores*, estructura de las clases E y EXP bajo diversas reducciones, complejidad no uniforme, complejidad de Kolmogorov, *natural proofs* y generadores pseudoaleatorios, la densidad de problemas *hard*, etc... (para un resumen de resultados ver [65, 3, 14]).

La medida de recursos acotados es pues una herramienta de gran utilidad en Complejidad Computacional pero también tiene sus limitaciones. En realidad, esto resulta lógico, dado que la medida de Lebesgue clásica de la que proviene también las tiene (por ejemplo, la existencia de conjuntos no medibles).

La forma en que la matemática clásica ha superado las limitaciones de la medida de Lebesgue ha sido mediante el concepto de dimensión de Hausdorff [28]. Esta herramienta permite trabajar con todo tipo de conjuntos (incluidos los no medibles) y distinguir entre conjuntos de medida cero. Del mismo modo, ante las limitaciones que la medida de recursos pueda tener, se introdujo en [67] la dimensión de recursos acotados.

1.2. Principales contribuciones

El principal objetivo de esta tesis es profundizar en las relaciones existentes entre la dimensión de recursos acotados y con escala y diversas medidas de complejidad como la complejidad de Kolmogorov, los ratios de compresión o el rendimiento de los algoritmos de aprendizaje. Estas relaciones constituirán un puente entre los resultados de complejidad y los de dimensión, permitiendo establecer nuevas cotas superiores e inferiores de complejidad para problemas y clases de problemas cuya dimensión se conoce, así como nuevos resultados sobre el tamaño de clases en función de su complejidad.

A continuación se describen con detalle las principales contribuciones de esta tesis. Los conceptos básicos de complejidad que se mencionan pueden encontrarse en 1.3.

1.2.1. Relaciones entre dimensión y complejidad de Kolmogorov

Uno de los campos de interés de la dimensión es el estudiar cómo se relaciona con otros conceptos bien conocidos de la Teoría de la Información. En particular, es natural intentar establecer una relación con la complejidad de Kolmogorov, puesto que ambas son medidas de cantidad de información. Por ejemplo, Ryabko [80, 81], Staiger [87, 88], y Cai y Hartmanis [16] fueron los primeros en estudiar la relación entre dimensión clásica de Hausdorff y complejidad de Kolmogorov. Por un

lado, Ryabko [81] proporcionó una cota superior para la dimensión de Hausdorff en términos de la complejidad de Kolmogorov. Más concretamente, demostró que, para todo conjunto de secuencias infinitas $X \subseteq \mathbf{C}$,

$$\dim_{\mathrm{H}}(X) \leq \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

Por otro lado, Staiger [87] demostró que la igualdad no era siempre posible. Es decir, existen conjuntos $X \subseteq \mathbf{C}$ para los cuales

$$\dim_{\mathrm{H}}(X) < \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

Con el desarrollo de la versión constructiva de la dimensión de Hausdorff, Mayordomo [77] y Lutz [68] demostraron que se conseguía una caracterización completa de la dimensión constructiva:

$$\mathrm{cdim}(X) = \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n},$$

para todo conjunto $X \subseteq \mathbf{C}$.

En el Capítulo 2 se amplia este resultado a la dimensión constructiva con escala, obteniendo que

$$\mathrm{cdim}_g(A) = \liminf_{n \to \infty} f_g^{n+1}(K(A[0..n-1])),$$

donde $f_g^{|w|+1}$ es la función inversa de g, siendo g la correspondiente escala (ver Sección 1.4.5). Para llegar a este resultado, se define la dimensión con escala de secuencias finitas, generalizando la defnición de dimensión en $\{0,1\}^*$ proporcionada por Lutz en [68]. Estos resultados han sido publicados en [60].

En cuanto a los casos de dimensión en espacio polinómico y dimensión calculable, Hitchcock [33] demostró que también es posible una caracterización completa para el caso no escalado. En este caso es necesario utilizar la versión de complejidad de Kolmogorov con recursos de espacio acotados. Más concretamente se tiene que, para todo conjunto $X \subseteq \mathbf{C}$,

$$\dim_{\mathrm{pspace}}(X) = \inf_{\mathrm{s} \in \mathrm{pspace}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \tag{1.2.1}$$

$$\dim_{\mathrm{comp}}(X) = \inf_{\mathrm{s} \in \mathrm{comp}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^{\mathrm{s}}(A[0..n-1])}{n}. \tag{1.2.2}$$

Para demostrar estos resultados, las técnicas utilizadas por Hitchock pasan por caracterizar dimensión con otro concepto bien conocido de Teoría de la Información: la entropía. Esta, a su vez, se relaciona con la complejidad de Kolmogorov, obteniéndose así las igualdades (1.2.1) y (1.2.2).

La idea de Hitchcock se basa en el trabajo de Staiger [87, 88], donde se define un tipo de entropía que caracteriza la dimensión clásica de Hausdorff. A partir de la versión calculable y la versión en espacio polinómico de esta entropía, Hitchcok demostró que era posible caracterizar las dimensiones calculable y en espacio polinómico respectivamente [33].

En el Capítulo 3 se extienden los resultados obtenidos por Hitchcock para el caso de dimensión con escala, caracterizándola mediante ambos conceptos: la complejidad de Kolmogorov y la entropía. Por último, se utiliza esta caracterización para estudiar el comportamiento de las reducciones P/poly-Turing en la clase ESPACE. Estos resultados han sido publicados junto con John Hitchcock y Elvira Mayordomo en [41] y posteriormente en una versión extendida de revista en [38].

1.2.2. Relaciones entre dimensión y compresión

Tal como se comenta en la anterior subsección, en los casos de dimensión constructiva, dimensión calculable y dimensión en espacio polinómico se consiguen caracterizaciones a partir de la complejidad de Kolmogorov clásica y acotada en espacio. Sin embargo, en el caso de la dimensión en tiempo polinómico no parecen existir posibles caracterizaciones de ese tipo [40]. Esto no es extraño puesto que computar (incluso de manera aproximada) la complejidad de Kolmogorov acotada en tiempo parece que requiere una búsqueda exponencial (bajo las hipótesis habituales en Complejidad Computacional). La principal diferencia entre complejidad de Kolmogorov acotada en tiempo y en espacio es la reversividad. En el caso de cotas en espacio, la fase de codificación puede hacerse con cotas de espacio similares a la de decodificación.

En el Capítulo 4 se utiliza la noción usual de algoritmo de compresión para secuencias finitas para caracterizar la dimensión en tiempo polinómico. Gracias a esta nueva caracterización, varios resultados conocidos para dimensión en tiempo polinómico se pueden interpretar como resultados de compresión. Por ejemplo, los lenguajes de una clase de p-dimensión 1 no pueden comprimirse (i.o.) en más que una cantidad sublineal. Así se obtienen resultados sobre la compresibilidad de lenguajes completos y autoreducibles. Estos resultados han sido publicados junto con Elvira Mayordomo en [61].

Dentro de los algoritmos de compresión estudiados en el Capítulo 4, sin duda uno de los más conocidos es el algoritmo de Lempel-Ziv [96]. Este algoritmo es actualmente el método de compresión universal más utilizado en el mundo (por ejemplo, suele comprimir textos largos en inglés a la mitad de su tamaño original). No sólo eso, el algoritmo de Lempel-Ziv también se utiliza en los formatos de imágenes TIFF y GIF, y dentro del software de Adobe Acrobat, entre otros.

El motivo de su éxito es la universalidad del algoritmo frente a los algoritmos de compresión de estados finitos. Más concretamente, en [96] se demuestra de un modo teórico que LZ_{78} tiene un ratio de compresión mejor que el de cualquier compresor de estados finitos. Este tipo de demostración teórica asegura, no sólo la conveniencia de usar el algoritmo de Lempel-Ziv frente a los algoritmos conocidos hasta el momento, sino frente a los algoritmos de compresión de estados finitos que pudan surgir en un futuro.

En este momento, Lempel-Ziv sigue siendo uno de los algoritmos más estudiados en el mundo y

pese a ello, todavía quedan cuestiones abiertas sobre el comportamiento del algoritmo. En el Capítulo 5 se estudia el ratio de compresión de Lempel-Ziv desde el punto de vista de la dimensión y se tratán algunas de estas cuestiones, como por ejemplo, el problema abierto de la catástrofe del bit. Estos resultados están publicados en [59].

1.2.3. Relaciones entre dimensión y aprendizaje

Las primeras relaciones entre medida de recursos acotados y teoría de aprendizaje fueron establecidas por Watanabe y otros autores en [57] donde se investiga la medida de recursos acotados de clases que son aprendibles con un algoritmo PAC o con un algoritmo basado en preguntas de equivalencia. Mas concretamente, se prueba en su trabajo que: i) Las subclases de P/poly que pueden aprenderse con un algoritmo PAC tiene medida polinómica 0 si EXP $\not\subseteq$ AM; y ii) las clases P/poly que pueden aprenderse con preguntas de equivalencia tienen medida polinómica 0. De estos resultados, se obtienen hipótesis en medidas de recursos acotados que implicarían el no aprendizaje de la clase de circuitos Booleanos en tiempo polinómico. Por otro lado, en el contexto de dimensión efectiva, Hitchcock exploró en [34] la relación de dimensión con la logarithmic loss unpredictability.

En el Capítulo 6 se proporcionan nuevos resultados en línea con los citados anteriormente. Primero, en relación con el aprendizaje on-line, se obtiene una cota superior de la dimensión en tiempo polinómico de clases de conceptos que pueden aprenderse con algoritmos on-line en tiempo exponencial y con $\alpha 2^n$ errores. Es más, se demuestra que esta cota superior es óptima (en el sentido de que no se puede mejorar). Basándose en los resultados obtenidos en esta tesis, Hitchcock ha investigado en [37] el caso de tener un número de errores subexponencial y dimensión cero, con interesantes aplicaciones que desarrolla en [27].

En segundo lugar, en relación con el aprendizaje PAC, se demuestra en el Capítulo 6 que la dimensión en espacio polinómico de clases de conceptos que son PAC-aprendibles es cero. Esto proporciona una hipótesis basada en dimensión efectiva que implica la impredicibilidad inherente de una clase de conceptos (la no predictivilidad es una propiedad muy interesante en aprendizaje computacional que hace referencia a las clases que usando cualquier hipótesis no son PAC aprendibles). Más aún, existen conexiones entre resultados de aprendizaje PAC y construcciones en el campo de la criptografía [48] que se pueden reescribir con hipótesis de dimensión efectiva.

Finalmente, se estudia la dimensión de clases que se pueden aprender con algoritmos basados en preguntas de pertenencia. El principal resultado demuestra que la dimensión en espacio polinómico de clases de conceptos que pueden aprenderse con un algoritmo basado en preguntas de pertenencia es cero. Esto puede usarse para demostrar que, para clases que son complejas en el sentido de dimensión, es necesaria una representación compleja para poder aprender eficientemente esa clase con preguntas de pertenencia. Estos resultados han sido publicados junto con Ricard Gavalda, Elvira

Mayordomo y Vinodchandran N. Variyam en [25].

1.3. Preliminares

En esta sección se fija en primer lugar la notación relacionada con secuencias y lenguajes, clases de complejidad, y funciones y series. Además, se da una breve descripción de distintos conceptos y herramientas que se utilizan a lo largo de la tesis dentro del área de Complejidad Computacional, Teoría de la Información, Compresión de Datos y Teoría de Aprendizaje. Esta es una sección prescindible para aquellos lectores que tengan conocimientos sobre estos conceptos y herramientas.

1.3.1. Notación básica

Secuencias y lenguajes

Se denotará por $\{0,1\}^*$ el conjunto de todas las secuencias finitas de ceros y unos. λ denotará la palabra vacía. $s_0, s_1, s_2 \ldots$ denotará la *enumeración estándar de* $\{0,1\}^*$, es decir,

$$s_0 = \lambda,\ s_1 = 0,\ s_2 = 1,\ s_3 = 00,\ s_4 = 01,\ s_5 = 10,\ s_6 = 11,\ s_7 = 000 \ldots .$$

En general, w denotará una secuencia finita de ceros y unos, es decir, $w \in \{0,1\}^*$. Se denotará la longitud de w como $|w|$.

Dado $n \in \mathbb{N}$, w^n denotará la concatenación de n veces w, es decir, recursivamente,

$$w^0 = \lambda \ \text{ y } \ w^n = w^{n-1} \cdot w.$$

Dado $n \in \mathbb{N}$ y R una relación binaria en $\{=, \leq, \geq, <, >\}$, se denotará

$$\{0,1\}^{Rn} = \{w \in \{0,1\}^* \,|\, |w| R n\}.$$

En particular, será de especial importancia el conjunto

$$\{0,1\}^{=n} = \{w \in \{0,1\}^* \,|\, |w| = n\},$$

y se denotará por $s_0^n, \ldots s_{2^n-1}^n$ la enumeración estándar de sus elementos, es decir,

$$s_0^n = 0^n,\ s_1^n = 0^{n-1}1, \ldots, s_{2^n-1}^n = 1^n.$$

Se denotará por $\{0,1\}^\infty$ el conjunto de todas las secuencias binarias infinitas.

En general, x denotará una secuencia binaria, finita o infinita, es decir $x \in \{0,1\}^* \cup \{0,1\}^\infty$.

Dados x y w como antes, $w \sqsubseteq x$ significará que w es un prefijo de x y $w \sqsubset x$ significará que w es un prefijo propio de x.

Se define el *cilindro generado por* $w \in \{0,1\}^*$ como $C_w = \{x \in \{0,1\}^\infty \,|\, w \sqsubset x\}$. Notar que $C_\lambda = \{0,1\}^\infty$.

Dados $i, j \in \mathbb{N}$ con $i \leq j$, se denotará por $x[i \dots j]$ la secuencia finita de ceros y unos que consiste en la subsecuencia de x desde el i-esimo bit hasta el j-esimo bit.

Un *lenguaje*, o *problema decisional*, es un conjunto $A \subseteq \{0,1\}^*$.

Dado $n \in \mathbb{N}$ y R una relación binaria en $\{=, \leq, \geq, <, >\}$, se denotará

$$A^{Rn} = A \cap \{0,1\}^{Rn}.$$

En particular, serán de especial importancia los conjuntos

$$A^{=n} = A \cap \{0,1\}^{=n} \text{ y } A^{\leq n} = A \cap \{0,1\}^{\leq n}.$$

Se define el lenguaje A^{io} como,

$$A^{io} = \{\ S \in \{0,1\}^\infty |\ \exists^\infty n \text{ con } S[0 \dots n] \in A\ \}.$$

Se dice que A es un *conjunto prefijo* si ningún elemento de A es un prefijo propio de otro elemento de A.

Cada lenguaje A se puede identificar con su secuencia característica en $\{0,1\}^\infty$:

$$\chi_A = [\![s_0 \in A]\!][\![s_1 \in A]\!][\![s_2 \in A]\!][\![s_3 \in A]\!] \dots$$

donde

$$[\![s_i \in A]\!] = \begin{cases} 1 & \text{si } s_i \in A, \\ 0 & \text{si } s_i \notin A. \end{cases}$$

El *espacio de Cantor* $\mathbf{C}$ es el conjunto $\{0,1\}^\infty$ de todas las secuencias binarias infinitas. Con la identificación anterior, el espacio de Cantor es el espacio de todos los lenguajes.

Dados $A, B \in \mathbf{C}$, se dice que A es *many-one reducible* a B ($A \leq_m B$) si existe una función calculable f tal que $x \in A$ sí y sólo sí $f(x) \in B$.

Clases de complejidad

Se necesitarán las siguientes clases de funciones para definir las clases de complejidad que se usarán posteriormente.

Para cada $i \in \mathbb{N}$ se define la *clase* G_i de funciones $f : \mathbb{N} \to \mathbb{N}$ recursivamente como sigue:

$$\begin{aligned} G_0 &= \{f | (\exists k)(\forall^\infty n) f(n) \leq kn\}, \\ G_{i+1} &= \{f | (\exists g \in G_i)(\forall^\infty n) f(n) \leq 2^{g(\log n)}\}. \end{aligned}$$

Las funciones de estas clases se usarán como cotas de crecimiento. En particular, G_0 contiene todas las funciones acotadas linealmente y G_1 contiene las funciones acotadas polinomialmente.

Se definen las funciones $\widehat{g}_i \in G_i$ como:

$$\begin{aligned} \widehat{g}_0(n) &= 2n, \\ \widehat{g}_{i+1}(n) &= 2^{\widehat{g}_i(\log n)}. \end{aligned}$$

Cada G_i es cerrado bajo composición, cada $f \in G_i$ es $o(\widehat{g}_{i+1})$ y cada $\widehat{g}_i$ es $o(2^n)$. Así pues, G_i contiene cotas superpolinomiales para todo $i > 1$, pero la jerarquía G_i es siempre subexponencial.

Dentro de la clase de todos los lenguajes decidibles (DEC), serán especialmente importantes en esta tesis las clases de complejidad exponenciales:

$$\begin{aligned} \mathrm{E}_i &= \mathrm{DTIME}(2^{G_{i-1}}) \qquad (i \geq 1), \\ \mathrm{E}_i\mathrm{SPACE} &= \mathrm{DSPACE}(2^{G_{i-1}}) \qquad (i \geq 1). \end{aligned}$$

En particular, se denota

$$\begin{aligned} \mathrm{E} &= \mathrm{E}_1 = \mathrm{DTIME}(2^{\mathrm{lineal}}), \\ \mathrm{EXP} &= \mathrm{E}_2 = \mathrm{DTIME}(2^{\mathrm{polinomial}}), \\ \mathrm{ESPACE} &= \mathrm{E}_1\mathrm{SPACE} = \mathrm{DSPACE}(2^{\mathrm{lineal}}), \\ \mathrm{EXPSPACE} &= \mathrm{E}_2\mathrm{SPACE} = \mathrm{DSPACE}(2^{\mathrm{polinomial}}). \end{aligned}$$

Otro modo de definir estas clases de complejidad es utilizando constructores:

Un *constructor* es una función $\delta : \{0,1\}^* \to \{0,1\}^*$ que satisface $w \sqsubseteq \delta(w)$ para todo $w \in \{0,1\}^*$. El resultado de un constructor δ (es decir, el lenguaje construido por δ) es el único lenguaje $R(\delta)$ tal que $\delta^n(\lambda) \sqsubseteq R(\delta)$ para todo $n \in \mathbb{N}$.

Las siguientes familias de funciones servirán para dar una definición equivalente de las clases vistas anteriormente:

$$\begin{aligned} \mathrm{all} &= \{f \mid f : \{0,1\}^* \to \{0,1\}^*\}, \qquad (1.3.1) \\ \mathrm{comp} &= \{f \in \mathrm{all} \mid f \text{ es calculable}\}, \\ \mathrm{p}_i &= \{f \in \mathrm{all} \mid f \text{ es calculable en tiempo } G_i\} \quad (i \geq 1), \\ \mathrm{p}_i\mathrm{space} &= \{f \in \mathrm{all} \mid f \text{ es calculable en espacio } G_i\} \quad (i \geq 1), \\ \mathrm{plogon} &= \{f : \{0,1\}^* \to \{0,1\}^* \mid f \text{ es calculable por una máquina on-line con espacio de} \\ &\qquad \text{trabajo y de salida polilogarítmico en el tamaño de la entrada}\}. \end{aligned}$$

La longitud de la salida se incluye como parte del espacio usado en la computación de f. Se denota p para p_1 y pspace para $\mathrm{p}_1$space. En general, Δ denotará una de las clases anteriores: comp, p_i, $\mathrm{p}_i$space $(i \geq 1)$.

Se tiene entonces que [62]

$$\begin{aligned} R(\text{all}) &= \mathbf{C}, \\ R(\text{comp}) &= DEC, \\ R(\text{p}_i) &= E_i \ \text{ para todo } i \geq 1, \\ R(\text{p}_i\text{space}) &= E_i SPACE \ \text{ para todo } i \geq 1. \end{aligned}$$

En el caso de la clase de funciones plogon, sólo si consideramos constructores que cumplen la definición de plogon (excepto con respecto al espacio de salida), obtenemos:

$$R(\text{plogon}) = PSPACE \text{ [76]}.$$

Ver [62, 67] para una introducción más detallada de los contenidos de esta subsección y la siguiente.

1.3.2. Funciones y series

En este apartado se define el significado de calculabilidad y convergencia bajo ciertas restricciones en los recursos de cálculo.

Sea D un dominio discreto y $f : D \to \mathbb{R}$ una función.

1. Se dice que f es Δ-*calculable* si existe una función $\widehat{f} : \mathbb{N} \times D \to \mathbb{Q}$ tal que:

   i) $|\widehat{f}(r, x) - f(x)| \leq 2^{-r}$ para todo $r \in \mathbb{N}$ y $x \in D$.

   ii) $\widehat{f} \in \Delta$, donde r se codifica en unario y la salida en binario.

2. Se dice que f es *exactamente* Δ-*calculable* si $f : D \to \mathbb{Q}$ y $f \in \Delta$.

3. Se dice que f es *semicalculable por debajo o constructiva* si existe una función calculable $\widehat{f} : D \times \mathbb{N} \to \mathbb{Q}$ tal que:

   i) Para todo $(x, t) \in D \times \mathbb{N}$,
   $$\widehat{f}(x, t) \leq \widehat{f}(x, t + 1) < f(x).$$

   ii) Para todo $x \in D$,
   $$\lim_{t \to \infty} \widehat{f}(x, t) = f(x).$$

Sea una serie de números reales no negativos, $\sum_{n=0}^{\infty} a_n$, entonces se dice que la serie es Δ-*convergente* si existe una función $h : \mathbb{N} \to \mathbb{N}$ tal que $h \in \Delta$ y

$$\sum_{n=h(r)}^{\infty} a_n \leq 2^{-r}$$

para todo $r \in \mathbb{N}$. Dicha función h se denominará Δ-*módulo de convergencia*.

Sea una secuencia de series de numeros reales no negativos,

$$\sum_{n=0}^{\infty} a_{j,n} \qquad (j = 0, 1, 2, \ldots),$$

entonces se dice que la secuencia es *uniformemente* Δ-*convergente* si existe una función $f : \mathbb{N}^2 \to \mathbb{N}$ tal que $f \in \Delta$ y f_j es un módulo de convergencia de la serie $\sum_{n=0}^{\infty} a_{j,n}$ para cada $j \in \mathbb{N}$ (donde $f_j(n) = f(j, n)$ para cada $j, n \in \mathbb{N}$).

1.3.3. Complejidad no uniforme: funciones consejo, clase P/poly y reducciones $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$

Las clases de complejidad introducidas en la sección anterior se definen exclusivamente restringiendo recursos de cálculo en los algoritmos (máquinas de Turing) encargados de aceptar o no un lenguaje. Este tipo de definición tiene sentido cuando se consideran lenguajes infinitos, pero no resulta útil para clasificar lenguajes finitos (puesto que estos pueden reconocerse en tiempo y espacio constante). Para esto último se mide el *tamaño* de algoritmos (circuitos booleanos) que aceptan conjuntos finitos.

Esta forma de medir recursos de cálculo se puede extender a los lenguajes infinitos asociando para cada $A \in \mathbf{C}$ una función que describa el crecimiento de los tamaños de los circuitos que aceptan cada $A^{=n}$. Formalmente:

Definición 1.3.1. Sea $A \in \mathbf{C}$.

1. Para cada $n \in N$, se define la *complejidad de circuito de A en la longitud n*, $c_A(n)$, como el tamaño del menor circuito booleano que acepta $A^{=n}$.

2. Se define la *complejidad de circuito de A* como la función $c_A : \mathbb{N} \to \mathbb{N}$ que para cada n nos devuelve $c_A(n)$.

3. Dada $\alpha : \mathbb{N} \to \mathbb{N}$ definimos la clase $\mathrm{SIZE}(\alpha(n))$ como

$$\mathrm{SIZE}(\alpha(n)) = \{A \mid \forall n \; c_A(n) \leq \alpha(n)\}.$$

Para establecer una conexión entre estas dos maneras de enfocar la complejidad: uniforme (definida en la subsección 1.3.1) y no uniforme (la definida en esta subsección), se introduce el concepto de *función consejo (advice)*. Estas funciones son las funciones del tipo $f : \mathbb{N} \to \{0, 1\}^*$. A partir de una clase $\mathcal{F}$ de funciones consejo y una clase de complejidad $\mathcal{C}$, se define la clase $\mathcal{C}/\mathcal{F}$ como todos los conjuntos B para los cuales existe un lenguaje $A \in \mathbf{C}$ y una función $f \in \mathcal{F}$ tal que

$$B = \{w \mid \langle w, f(|w|)\rangle \in A\}.$$

Intuitivamente, cuando $\mathcal{C}$ es una clase de complejidad como las definidas en la subsección 1.3.1, $\mathcal{C}/\mathcal{F}$ es la clase de todos los conjuntos B tal que alguna función de $\mathcal{F}$ proporciona información suficiente para aceptar B dentro de las cotas de recursos especificados en la definición de $\mathcal{C}$.

Como ejemplo particular de este tipo de clases, se tiene la *clase P/poly*, donde poly denota la clase de funciones $f : \mathbb{N} \to \{0,1\}^*$ de forma que para algún polinomio p, $|f(n)| \leq p(n)$ para cada $n \in \mathbb{N}$.

Es decir, P/poly es la clase de conjuntos $B = \{w \,|\, \langle w, f(|w|)\rangle \in A\}$, donde $A \in$ P y para algún polinomio p y todo n, $|f(n)| \leq p(n)$.

Esta clase se relaciona con la complejidad no uniforme mediante el siguiente teorema.

Teorema 1.3.2. Sea $A \in \mathbf{C}$. Entonces, $A \in$ P/poly si y solo si existe un polinomio p tal que para todo n, $c_A(n) \leq p(n)$.

Utilizando este teorema las reducciones $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$ (reducciones Turing que se computan por una familia no uniforme de circuitos de tamaño polinomial) se pueden definir formalmente como sigue.

Definición 1.3.3. [9] Dados $A, B \in \mathbf{C}$, se dice que A es $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$-reducible a B si existen M máquina de Turing con oráculo en tiempo polinómico y $h \in$ poly tales que

$$A = \{w \in \{0,1\}^* \,|\, M^B \text{ acepta } \langle w, h(|w|)\rangle\}.$$

Existen varios textos básicos de complejidad computacional donde se desarrollan en detalle estos conceptos, por ejemplo [9] contiene complejidad no uniforme.

1.3.4. Complejidad de Kolmogorov y medidas de probabilidad en $\{0,1\}^*$

La complejidad de Kolmogorov de una secuencia finita de ceros y unos es la longitud de la descripción más corta de dicha secuencia. La idea intuitiva que persigue esta definición es el proporcionar una medida de la información intrínseca que tiene una secuencia. Por ejemplo, fijándose en las siguientes secuencias de longitud 30:

$$\begin{aligned} A &= 010101010101010101010101010101, \\ B &= 101100101011001011111110010101, \end{aligned}$$

la secuencia A se puede describir como "15 repeticiones de 01", mientras que, a simple vista, la secuencia B no parece tener una descripción más corta. Así pues, intuitivamente, la secuencia A parece más sencilla que la secuencia B y por lo tanto, parece tener una complejidad de Kolmogorov más pequeña que la secuencia B. La definición formal de complejidad de Kolmogorov es la siguiente.

Definición 1.3.4. Dada $w \in \{0,1\}^*$ y M una maquina de Turing, se define la *complejidad de Kolmogorov de w con respecto a M* como

$$K_M(w) = \text{mín}\{|x| \mid x \in \{0,1\}^* \text{ tal que } M(x) = w\}.$$

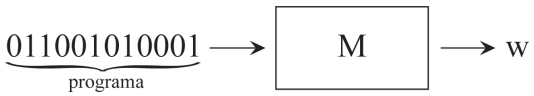

Figura 1.3: Maquina de Turing que genera w a partir de un programa.

Es decir, la complejidad de Kolmogorov de w relativa a una máquina fija M es el programa más corto que al ejecutarlo en M devuelve w (figura 1.3).

Esta definición depende de la máquina M utilizada, sin embargo su dependencia es sólo por una constante aditiva cuando se retringe a máquinas universales, tal como se ve en el siguiente teorema.

Teorema 1.3.5 (Invarianza de la complejidad de Kolmogorov)**.** Si U es una máquina de Turing universal, entonces para cualquier otra máquina de Turing M,

$$K_U(w) \leq K_M(w) + c_M,$$

para toda secuencia $w \in \{0,1\}^*$, donde la constante c_M no depende de w.

Así pues, al hablar de complejidad de Kolmogorov, se fijará una máquina de Turing universal y se denotará simplemente por $K(w)$.

Además del concepto de complejidad de Kolmogorov, donde los recursos para calcular w son ilimitados, se puede definir complejidad de Kolmogorov con recursos acotados. La idea es restringir el poder de la máquina que se usa para calcular w. En general, se usará $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ para cotas de espacio y $\mathrm{t} : \mathbb{N} \to \mathbb{N}$ para cotas de tiempo.

Definición 1.3.6. Sea M una máquina de Turing, $w \in \{0,1\}^*$ y $\mathrm{s}, \mathrm{t} : N \to \mathbb{N}$ cotas de espacio y tiempo.

1. La *complejidad de Kolmogorov acotada en espacio* s *relativa a* M se define como

$$KS^{\mathrm{s}}_M(w) = \text{mín}\{|x| \mid x \in \{0,1\}^* \text{ tal que } M(x) = w \text{ en espacio } \leq \mathrm{s}(|w|)\}.$$

2. La *complejidad de Kolmogorov acotada en tiempo* t *relativa a* M se define como

$$K^{\mathrm{t}}_M(w) = \text{mín}\{|x| \mid x \in \{0,1\}^* \text{ tal que } M(x) = w \text{ en tiempo } \leq \mathrm{t}(|w|)\}.$$

Al igual que en el caso de complejidad de Kolmogorov clásica, para complejidad de Kolmogorov con recursos acotados también existe un teorema de invarianza. Sin embargo, al añadir cotas en los recursos de cálculo, la propiedad de invarianza es significativamente más débil.

Teorema 1.3.7 (Invarianza de la complejidad de Kolmogorov con recursos acotados)**.** Si U es una máquina de Turing universal y $\mathrm{s}, \mathrm{t} : \mathbb{N} \to \mathbb{N}$ son cotas de espacio y tiempo, entonces para cualquier otra máquina de Turing M,

$$\begin{aligned} KS^{c\mathrm{s}}_U(w) &\leq KS^{\mathrm{s}}_M(w) + c, \\ K^{c'\mathrm{t}\log\mathrm{t}}_U(w) &\leq K^{\mathrm{t}}_M(w) + c', \end{aligned}$$

para toda secuencia $w \in \{0,1\}^*$, donde las constantes c y c' no dependen de w.

Al hablar de complejidad de Kolmogorov con recursos acotados, se fijará una máquina de Turing universal y se denotará simplemente $KS^{\mathrm{s}}(w)$ y $K^{\mathrm{t}}(w)$.

Dado un lenguaje A se utilizará la siguiente notación para referirse a la complejidad de Kolmogorov en espacio de la secuencia de 2^n bits que caracteriza $A^{=n}$ o de la secuencia de $2^{n+1}-1$ bits que caracteriza $A^{\leq n}$.

Notación 1.3.8. Sean U una máquina de Turing universal, $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ una cota de espacio, A un lenguaje y n un número natural,

1. La *complejidad de Kolmogorov acotada en espacio* $\mathrm{s}(n)$ *de* $A^{=n}$ se define como
$$KS^{\mathrm{s}}(A^{=n}) = \mín\{|x| \mid x \in \{0,1\}^* \text{ tal que } U(x) = A[2^n - 1 \ldots 2^{n+1} - 2] \text{ en espacio } \leq \mathrm{s}(n)\}.$$
2. La *complejidad de Kolmogorov acotada en espacio* $\mathrm{s}(n)$ *de* $A^{\leq n}$ se define como
$$KS^{\mathrm{s}}(A^{\leq n}) = \mín\{|x| \mid x \in \{0,1\}^* \text{ tal que } U(x) = A[0 \ldots 2^{n+1} - 2] \text{ en espacio } \leq \mathrm{s}(n)\}.$$

Uno de los resultados más importantes sobre complejidad de Kolmogorov es el que la relaciona con medidas de subprobabilidad.

Definición 1.3.9.
1. Una *medida de subprobabilidad en* $\{0,1\}^*$ es una función $p : \{0,1\}^* \to [0,1]$ tal que verifica
$$\sum_{w \in \{0,1\}^*} p(w) \leq 1. \tag{1.3.2}$$
2. Una *medida de probabilidad en* $\{0,1\}^*$ es una medida de subprobabilidad en $\{0,1\}^*$ que verifica la condición (1.3.2) con igualdad.
3. Una medida de subprobabilidad en $\{0,1\}^*$ es *constructiva* si es semicalculable por debajo.
4. Una medida de subprobabilidad p en $\{0,1\}^*$ es *constructiva óptima* si para cada medida de subprobabilidad constructiva p' existe una constante real $\alpha > 0$ tal que:
$$p(w) > \alpha p'(w),$$
para todo $w \in \{0,1\}^*$.

El Teorema de Levin asegura la existencia de una medida de subprobabilidad óptima.

Teorema 1.3.10. (Levin [97]) Existe una medida de subprobabilidad constructiva óptima en $\{0,1\}^*$. Se denotará por $\mathbf{m}$.

El siguiente teorema es una caracterización de la complejidad de Kolmogorov en términos de $\mathbf{m}$. Este resultado se debe a Levin [53, 54] y Chaitin [17].

Teorema 1.3.11. Existe una constante $c \in \mathbb{N}$ tal que, para todo $w \in \{0,1\}^*$,
$$\left| K(w) - \log \frac{1}{\mathbf{m}(w)} \right| \leq c.$$

Para mayores detalles sobre complejidad de Kolmogorov y las demostraciones de los teoremas enunciados en esta sección, se remite al lector a [55].

1.3.5. Medidas de probabilidad en C y entropía

Sea $\mathcal{F}$ la σ-álgebra generada por los cilindros en $\mathbf{C}$. Sea $\nu : \mathcal{F} \to [0,1]$ una medida positiva con masa total 1. Entonces, $(\mathbf{C}, \mathcal{F}, \nu)$ es un *espacio de probabilidad* y ν es una *medida de probabilidad* en $\mathbf{C}$.

Ahora bien, al trabajar en $\mathbf{C}$, cada medida de probabilidad ν se puede identificar con una función $\mu : \{0,1\}^* \to [0,1]$ definida como $\mu(w) = \nu(C_w)$. Esta función μ verifica las siguientes condiciones de consistencia de Kolmogorov:

$$(i)\ \ \mu(\lambda) = 1.$$
$$(ii)\ \mu(w0) + \mu(w1) = \mu(w)\ \text{ para todo } w \in \{0,1\}^*.$$

Por otro lado, por el Teorema de existencia de Kolmogorov [11], para cada $\mu : \{0,1\}^* \to [0,1]$ que satisface las condiciones anteriores, existe una única medida de probabilidad ν_μ en $\mathbf{C}$ tal que $\nu_\mu(C_w) = \mu(w)$.

Así pues, por simplicidad, una función μ verificando (i) y (ii) se llamará también medida de probabilidad en $\mathbf{C}$. A partir de esta idea se tienen las siguientes definiciones.

Definición 1.3.12. 1. Una *supermedida de subprobabilidad en* $\mathbf{C}$ es una función $\mu : \{0,1\}^* \to [0,1]$ tal que verifica

$$(i)\ \ \mu(\lambda) \leq 1. \tag{1.3.3}$$
$$(ii)\ \mu(w) \geq \mu(w0) + \mu(w1)\ \text{ para todo }\ w \in \{0,1\}^*. \tag{1.3.4}$$

2. Una *medida de subprobabilidad en* $\mathbf{C}$ es una supermedida de subprobabilidad que verifica la condición (1.3.4) con igualdad para todo $w \in \{0,1\}^*$.

3. Una *medida de probabilidad en* C es una medida de subprobabilidad que satisface la condición (1.3.3) con igualdad.

Intuitivamente, si μ es una medida de probabilidad en C y $w \in \{0,1\}^*$, entonces $\mu(w)$ es la probabilidad de que $w \sqsubseteq A$ cuando la secuencia $A \in \mathbf{C}$ está "construida" de acuerdo con la medida de probabilidad ν_μ.

Así como en Complejidad Computacional se interpreta la complejidad de Kolmogorov de una secuencia como la medida de información intrínseca que dicha secuencia tiene, en Teoría de la Información, para cada medida de probabilidad, se define el concepto de entropía de modo que represente la incertidumbre que proporciona dicha probabilidad. En el caso particular de una medida de probabilidad en $\mathbf{C}$, se define:

Definición 1.3.13. Sea μ una medida de probabilidad en $\mathbf{C}$. La *entropía de* μ se define como,

$$\mathcal{H}(\mu) = \lim_n \frac{\mathcal{H}_n(\mu)}{n},$$

donde

$$\mathcal{H}_n(\mu) = \sum_{w \in \{0,1\}^n} \mu(w) \log \frac{1}{\mu(w)}.$$

Sea $0 < \alpha < 1$, la entropía binaria de Shannon $\mathcal{H}(\alpha)$ se define como,

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

Para obtener más información sobre esta sección y las demostraciones de los teoremas enunciados, se remite al lector a [11, 20].

1.3.6. Compresores de estados finitos

Los compresores de estados finitos fueron introducidos por Huffman [42] y desde entonces han sido extensamente investigados (ver por ejemplo [49, 50]). Se tratan de compresores que han evolucionado a los conocidos algoritmos de compresión de Lempel-Ziv (ver Sección 1.3.8).

Definición 1.3.14. Un *compresor de estados finitos o FSC (finite-state compressor)* es una 4-tupla $C = (Q, \delta, \nu, q_0)$ donde

i) Q es el conjunto de *estados*. Se trata de un conjunto no vacío y finito.

ii) $\delta : Q \times \{0,1\}^* \to Q$ es la *función de transición*.

iii) $\nu : Q \times \{0,1\} \to \{0,1\}^*$ es la *función de salida*.

iv) q_0 es el *estado inicial*.

En particular se define:

1. Dado q un estado en Q y $w \in \{0,1\}^*$, la *salida desde el estado q en la entrada w* es la secuencia finita $\nu(q, w)$ definida mediante la siguiente recursión:

$$\begin{aligned} \nu(q, \lambda) &= \lambda, \\ \nu(q, vb) &= \nu(q, v)\nu(\delta^*(q, v), b), \end{aligned}$$

   para todo $v \in \{0,1\}^*$ y $b \in \{0,1\}$, siendo la función $\delta^* : Q \times \{0,1\}^* \to Q$ definida mediante la recursión:

$$\begin{aligned} \delta^*(q, \lambda) &= q, \\ \delta^*(q, vb) &= \delta(\delta^*(q, v), b), \end{aligned}$$

   para todo $q \in Q$, $v \in \{0,1\}^*$ y $b \in \{0,1\}$.

   Abusando de notación, se escribirá δ por δ^* y para cada $w \in \{0,1\}^*$ se denotará con $\delta(w)$ el valor $\delta^*(q_0, w)$. Informalmente, $\delta(w)$ es el estado en el que el compresor termina de trabajar cuando se le ha proporcionado como entrada w.

2. La *salida de C en la entrada $w \in \{0,1\}^*$* es la secuencia finita

$$C(w) = \nu(q_0, w).$$

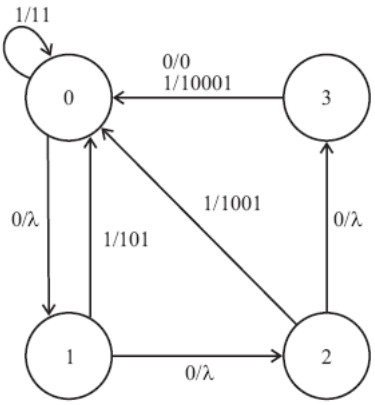

Figura 1.4: Diagrama de un ILFSC de 4 nodos.

Definición 1.3.15. Un compresor de estados finitos $C = (Q, \delta, \nu, q_0)$ se dice que *no tiene perdida de información* si la función

$$\begin{array}{ccc} \{0,1\}^* & \to & \{0,1\}^* \times Q \\ w & \mapsto & (C(w), \delta(w)) \end{array}$$

es inyectiva.

Un *compresor de estados finitos sin perdida de información* se denotará por *ILFSC (information-lossless finite-state compressor).*

Ejemplo 1.3.16. [82] La figura 1.4 representa un ILFSC de 4 estados $G = (Q, \delta, \beta, 0)$, donde $Q = (0, 1, 2, 3)$. Cada estado aparece representado como un círculo, etiquetado en su interior. La función de transición viene determinada por el primer valor de las aristas y la función de salida por el segundo valor. Así pues, el ILFSC que nos determina la figura 1.4 es aquel en el que los valores de la función de transición son los siguientes:

$$\begin{array}{cccc} \delta(0,0) = 1, & \delta(1,0) = 2, & \delta(2,0) = 3, & \delta(3,0) = 0, \\ \delta(0,1) = 0, & \delta(1,1) = 0, & \delta(2,1) = 0, & \delta(3,1) = 0. \end{array}$$

Los valores de la función de salida son los siguientes:

$$\begin{array}{cccc} \nu(0,0) = \lambda, & \nu(1,0) = \lambda, & \nu(2,0) = \lambda, & \nu(3,0) = 0, \\ \nu(0,1) = 11, & \nu(1,1) = 101, & \nu(2,1) = 1001, & \nu(3,1) = 10001. \end{array}$$

Y, por ejemplo, si $w = 0100$, se tiene que $C(w) = \nu(0, 0100) = 101$ y $\delta(w) = 2$.

Además, se puede ver que C es IL. Por ejemplo, si $C(w) = 110$ y $\delta(w) = 0$, entonces w debe tomar el valor $w = 10000$.

1.3.7. Apostadores de estados finitos

En esta sección se introducen unos modelos de computación que, esencialmente, son autómatas de estados finitos pero que, en vez de aceptar o rechazar una secuencia finita, "apuestan" en los

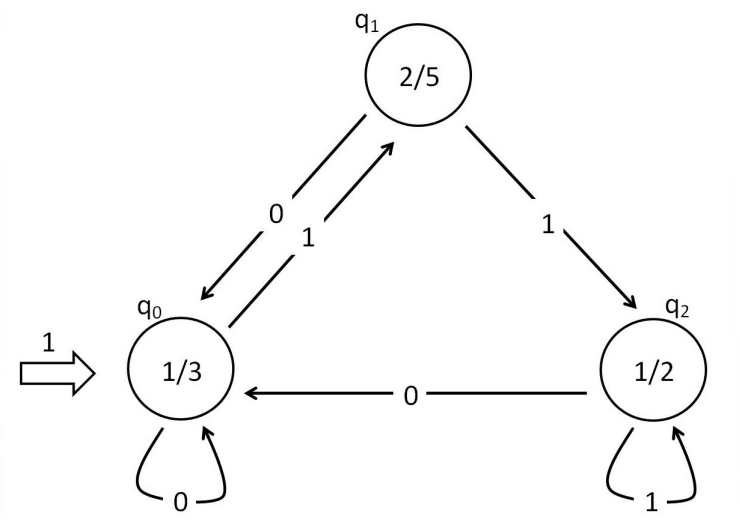

Figura 1.5: Diagrama de un FSG de 3 nodos.

sucesivos bits de una secuencia infinita [21]. Más formalmente,

Definición 1.3.17. Un *FSG (finite state gambler)* es una 5-tupla $G = (Q, \delta, \beta, q_0, c_0)$ donde

1. Q es el conjunto de *estados*. Se trata de un conjunto finito y no vacío.
2. $\delta : Q \times \{0,1\} \to Q$ es la *función de transición*.
3. $\beta : Q \to \mathbb{Q} \cap [0,1]$ es la *función de apuestas*.
4. q_0 es el *estado inicial*.
5. c_0 es el *capital inicial*.

Nota 1.3.18.

1. En general, se considerará $c_0 = 1$ y se suprimirá de la definición.
2. Los FSG se definieron originalmente utilizando k cuentas para trabajar [21]. Sin embargo, para las aplicaciones que interesan en esta tesis, es suficiente considerar una sola cuenta, lo cual da una definición equivalente de compresión, aunque pueda aumentar exponencialmente el número de estados necesarios.
3. Al igual que se hace en la definición de los compresores de estados finitos (ver Sección 1.3.14), se puede extender la función de transición $\delta : Q \times \{0,1\} \to Q$ a la función $\delta^* : Q \times \{0,1\}^* \to Q$. Igualmente, se denotará por $\delta(w)$ a $\delta^*(q_0, w)$.

Ejemplo 1.3.19. La figura 1.5 representa un FSG de 3 estados $G = (Q, \delta, \beta, q_0)$, donde $Q = (q_0, q_1, q_2)$. Cada estado aparece representado como un círculo y los valores en el interior de cada estado corresponden a la función de apuesta $\beta(q)$. La función de transición viene determinada por las flechas. Así pues, el FSG que nos determina la figura 1.5 es aquel en el que los valores de la función de apuestas son los siguientes:

$$\beta(q_0) = 1/3, \quad \beta(q_1) = 2/5, \quad \beta(q_2) = 1/2.$$

La función de transición por su parte toma los siguientes valores:

$$\begin{array}{lll} \delta(q_0,0)=q_0, & \delta(q_1,0)=q_0, & \delta(q_2,0)=q_0, \\ \delta(q_0,1)=q_1, & \delta(q_1,1)=q_2, & \delta(q_2,1)=q_2. \end{array}$$

En la Sección 1.4 (Definición 1.4.7) se formalizará la forma de apostar de un FSG. Intuitivamente, suponer que se tiene un FSG (Q,δ,β,q_0,c_0) en un estado q y con un capital c y se realiza la apuesta binaria $\beta(q)$. Si el juego es limpio, después de la apuesta si el siguiente bit resulta ser b, G estará en el estado $\delta(q,b)$ y el capital será

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

1.3.8. El algoritmo de compresión de datos de Lempel-Ziv

El algoritmo de compresión de datos de Lempel-Ziv se puede interpretar como una universalización de los compresores de estados finitos, ya que el algoritmo es óptimo respecto a todos estos compresores. Hoy en día se trata de uno de los algoritmos de compresión más estudiados y utilizados (por ejemplo, se utiliza en los formatos de imágenes TIFF y GIF, y dentro del software de Adobe Acrobat).

Las diferentes versiones del algoritmo de compresión de datos de Lempel-Ziv codifican una secuencia finita utilizando distintas particiones de ésta. En este contexto una frase es simplemente una secuencia finita.

Definición 1.3.20. Sea una secuencia finita $w \in \{0,1\}^*$,

1. Un *análisis (parsing)* de w se define como una partición de w en *frases* $w_1, w_2, \ldots, w_n$ de modo que $w_1 w_2 \ldots w_n = w$.

2. Un *análisis único (distinct parsing)* de w se define como un análisis de w de modo que ninguna frase, excepto quizá la última, es la misma que una frase anterior.

3. Un *análsis único válido (valid distinct parsing)* de $w \in \{0,1\}^*$ se define como un análisis único de w de modo que, si w_i es una frase en la secuencia w, entonces cada prefijo de w_i aparece antes de w_i en el análisis único. Notar que cada secuencia finita w sólo tiene un análisis único válido.

En el siguiente ejemplo se ve la diferencia entre análisis, análisis único y análisis único válido.

Ejemplo 1.3.21. Sea $w = 01001100010010$.

1. La siguiente partición es un análisis de w pero no es un análisis único puesto que existen dos frases iguales: w_1 y w_4.

$$\begin{array}{ccccc} 01 & 0011 & 00 & 01 & 0010 \\ w_1 & w_2 & w_3 & w_4 & w_5 \end{array}$$

2. La siguiente partición es un análisis único de w pero no es su análisis único válido puesto que existen frases (w_1, w_2 y w_3) cuyos prefijos no son frases anteriores.

$$\begin{array}{ccccc} 01 & 001 & 100 & 010 & 010 \\ w_1 & w_2 & w_3 & w_4 & w_5 \end{array}$$

3. La siguiente partición es el análisis único válido de w.

$$\begin{array}{cccccccc} 0 & 1 & 00 & 11 & 000 & 10 & 01 & 0 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 \end{array}$$

El algoritmo de compresión de datos de Lempel-Ziv 78 (LZ_{78}) reemplaza frases de un análisis por apuntadores a frases previas. Más concretamente,

Definición 1.3.22. [96] El *algoritmo de compresión de datos LZ_{78}* codifica una secuencia finita $w \in \{0,1\}^*$ utilizando su análisis único válido $w_1 \dots w_n$. Para ello, LZ_{78} codifica cada frase w_i con una tupla (j, b), donde j representa un puntero y b un bit. El puntero j indica el número del prefijo propio más largo de la frase y el bit b es el último bit de la frase, es decir, $w_i = w_j b$. Ambos especifican completamente la frase que está siendo codificada. Por ejemplo, la frase 4 del Ejemplo 1.3.21 (apartado 3) se codifica como $w_4 = (2, 1)$ (por convención, $w_0 = \lambda$).

Cada puntero a w_i se representa utilizando $\lceil \log i + 1 \rceil$ bits. De este modo, cada frase del análisis único válido se codifica utilizando $\lceil \log i + 1 \rceil + 1$ bits. La salida del algoritmo es una secuencia que se denota por $\mathrm{LZ}_{78}(w)$ y tiene longitud

$$|\mathrm{LZ}_{78}(w)| = \sum_{i=1}^{t(w)} (\lceil \log i + 1 \rceil + 1) \leq t(w)[\log t(w) + 1],$$

donde $t(w)$ es el número de frases en el análisis único válido (ver [82, 96] para más detalles).

En la siguiente definición se explica la partición utilizada en otra versión del algoritmo de compresión de Lempel-Ziv (LZ_{77}) [95].

Definición 1.3.23. Sea π el operador que borra el bit final de una secuencia finita w, es decir, $\pi(w) = w[0 \dots |w| - 2]$.

1. Una *historia de w* es un análisis $w = w_1 \dots w_n$ que tiene las propiedades:

   i) $w_1 \in \{0, 1\}$.

   ii) $\pi(w_i)$ es un trozo de la secuencia $\pi^2(w_1 \dots w_i)$, para todo $2 \leq i \leq n$.

   Es decir, cualquier nueva frase w_i, exceptuando su último bit, aparece anteriormente en la secuencia.

2. Una historia se dice *exhaustiva* si ninguno de los w_i, con $2 \leq i \leq n-1$ aparece antes. En otras palabras, el nuevo factor w_i no aparece antes en la palabra, aunque sí que lo hacen todos sus prefijos propios. Notar que cada secuencia w tiene una historia exhaustiva única y además, la historia exhaustiva es la más corta de todas las historias.

Utilizando la misma secuencia que en el Ejemplo 1.3.21 se ve la diferencia entre una historia y su historia exhaustiva.

Ejemplo 1.3.24. Sea $w = 01001100010010$.

1. La siguiente partición es una historia de w que no es una historia exhaustiva.

$$\begin{array}{ccccccc} 0 & 1 & 00 & 11 & 000 & 100 & 10 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{array}$$

   En efecto, cada w_i salvo su último bit aparece anteriormente en la secuencia.

2. La siguiente partición es la historia exhaustiva de w

$$\begin{array}{cccccc} 0 & 1 & 00 & 11 & 000 & 10010 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{array}$$

El algoritmo de compresión de datos de LZ_{77} reemplaza frases de la historia exhaustiva por apuntadores a un bit anterior y la longitud que debemos contar a partir de ese bit. Más concretamente,

Definición 1.3.25. [95] El *algoritmo de compresión de datos LZ_{77} (también llamado* unrestricted *LZ ó ULZ)* codifica una secuencia finita w utilizando su historia exhaustiva $w_1 \dots w_n$. Para ello LZ_{77} codifica cada frase w_i de la historia exhaustiva con una terna (p, l, b), donde p es la posición donde empieza la anterior ocurrencia en $w_i[0 \dots |w_i| - 2]$, l es la longitud de la ocurrencia (es decir, $l = |w_i| - 1$) y b es el siguiente carácter. Por ejemplo, la frase 6 del Ejemplo 1.3.24 (apartado 2) se codifica como $w_6 = (2, 4, 0)$.

Para representar cada frase de la historia son necesarios $2\lceil \log |w| \rceil + 1$ bits y por lo tanto,

$$|\mathrm{LZ}_{77}(w)| = \sum_{i=1}^{t_{77}(w)} (2\lceil \log |w| \rceil + 1) = t_{77}(w)(2\lceil \log |w| \rceil + 1),$$

donde $t_{77}(w)$ representa el número de frases de la historia exhaustiva de w.

Ejemplo 1.3.26. Sea $w = 0100110001001$.

1. Para aplicar el algoritmo de compresión LZ_{78} a w, primero encontramos el *valid distinct parsing* de w,

$$\begin{array}{ccccccc} 0 & 1 & 00 & 11 & 000 & 10 & 01 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{array}$$

   Ahora codificamos $w_i = (j, b)$ donde $w_i = w_j b$. Así pues, $w_1 = (0, 0)$, $w_2 = (0, 1)$, $w_3 = (1, 0)$, $w_4 = (2, 1)$, $w_5 = (3, 0)$, $w_6 = (2, 0)$ y $w_7 = (1, 1)$.

2. Para aplicar el algoritmo de compresión LZ_{77} a w, primero encontramos la historia exhaustiva de w,

$$\begin{array}{cccccc} 0 & 1 & 00 & 11 & 000 & 1001 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{array}$$

   Ahora codificamos $w_i = (p, l, b)$ donde $w_i = w[p - 1 \dots p - 2 + l]b$. Así pues, $w_1 = (0, 0, 0)$, $w_2 = (0, 0, 1)$, $w_3 = (1, 1, 0)$, $w_4 = (2, 1, 1)$, $w_5 = (3, 2, 0)$ y $w_6 = (2, 4, \lambda)$.

1.3.9. Modelos de aprendizaje

Los modelos de aprendizaje tratan de establecer los límites de lo que puede o no ser aprendido. Para definir un modelo de aprendizaje es necesario establecer un protocolo de aprendizaje y un procedimiento de deducción. Por un lado, el protocolo de aprendizaje especifica la manera en que se obtiene la información del mundo exterior. Por otro lado, el procedimiento de deducción es el mecanismo mediante el cual se deduce un algoritmo de reconocimiento del concepto que queremos aprender. Este mecanismo se denomina algoritmo de aprendizaje.

El primer paso para establecer un modelo de aprendizaje es establecer de qué manera codificamos la información del exterior. En esta tesis, la información consistirá en el valor de $n \in \mathbb{N}$ atributos booleanos. Se codificará esta información como una secuencia finita en $\{0,1\}^n$ que se denominará *instancia*. Así pues, el conjunto $\{0,1\}^n$ se denominará *espacio de instancias*. Un *concepto* c será un subconjunto de $\{0,1\}^n$ o, equivalentemente, una función booleana $c : \{0,1\}^n \to \{0,1\}$ donde $c(x) = 0$ si $x \notin c$ y $c(x) = 1$ si $x \in c$.

Sea $\mathcal{C}_n$ un subconjunto de conceptos en el espacio de instancias $\{0,1\}^n$. Una *representación* de $\mathcal{C}_n$ consiste en un conjunto de secuencias L_n y una aplicación σ_n de L_n en $\mathcal{C}_n$ que asocia cada secuencia en L_n con un concepto en $\mathcal{C}_n$. Una medida de complejidad para $\mathcal{C}_n$ es una aplicación $\mathbf{size}_n$ de $\mathcal{C}_n$ en $\mathbb{N}$ (normalmente se toma $\mathbf{size}_n(c)$ la longitud mínima de las secuencias que representan c en la representación (L_n, σ_n)).

Para cada $n \in \mathbb{N}$, sea $\mathcal{C}_n$ un conjunto de conceptos en $\{0,1\}^n$, L_n y σ_n una representación para $\mathcal{C}_n$, y $\mathbf{size}_n$ una medida de complejidad para $\mathcal{C}_n$. Entonces se dice que $\mathcal{C} = \{\mathcal{C}_n\}_n$ es una clase de conceptos y $\{(\mathcal{C}_n, L_n, \sigma_n, \mathbf{size}_n)\}_{n\in\mathbb{N}}$ es *la clase de representación de* $\mathcal{C}$. Normalmente tanto la representación como la medida de complejidad se sobrentienden del contexto.

Ejemplo 1.3.27. Consideremos la clase de conceptos k-CNF (para algún k fijo) del artículo de Valiant [94]. Aquí la representación L_n consiste en todas las fórmulas CNF de n variables $(x_1 \dots x_n)$ que tienen a lo mas k literales por cláusula, $\mathcal{C}_n$ consiste en todos los $c \subseteq \{0,1\}^n$ tal que c es el conjunto de asignaciones que satisfacen una de estas expresiones, σ_n manda una fórmula CNF al conjunto de asignaciones que la satisfacen, y $\mathbf{size}_n(c)$ es en este caso, el número de literales en la menor representación k-CNF de c.

A continuación introduciremos los tres modelos de aprendizaje que utilizaremos en esta tesis.

Aprendizaje aproximadamente correcto (PAC learning)

El modelo de aprendizaje aproximadamente correcto (PAC learning) formaliza el proceso de aprendizaje mediante ejemplos. En particular, las máquinas de soporte vectorial, las redes neuronales y los árboles de decisión están basados en este modelo teórico.

En el modelo PAC el algoritmo de aprendizaje tiene acceso a un conjunto de ejemplos (positivos y negativos) de un concepto objetivo c desconocido. Este concepto pertenece a una clase de conceptos

fija $\mathcal{C}$ que nos es conocida. El algoritmo de aprendizaje debe aproximar el concepto objetivo a partir de los ejemplos que va viendo. Esta idea fue desarrollada por Valiant en [94].
Más formalmente, sea $\mathcal{C}$ y $\mathcal{H}$ dos clases de conceptos. Un *algoritmo PAC* que aprende $\mathcal{C}$ mediante $\mathcal{H}$ es un algoritmo que, cuando se le da unos ejemplos de algún concepto $c \in \mathcal{C}$ devuelve como salida (la representación) de algún concepto $h \in \mathcal{H}$ que es una aproximación de c en el sentido que precisaremos más adelante. La clase $\mathcal{C}$ se llama *clase objetivo* y $\mathcal{H}$ se llama *clase de hipótesis*. De un modo equivalente, c se llama *concepto objetivo* y h *hipótesis del algoritmo*.

Definición 1.3.28. Sea $\{0,1\}^n$ un conjunto de instancias, sea D una distribución en el conjunto de instancias y sea c el concepto objetivo. El *error de h con respecto el concepto c y la distribución D* se define como

$$\text{error}_D(h,c) = Pr_{x \in D}[h(x) \neq c(x)].$$

Es decir, $\text{error}_D(h,c)$ es la probabilidad de que h y c no coincidan en una instancia elegida de un modo aleatorio siguiendo la distribución D. Intuitivamente, h será una buena aprozimación del concepto objetivo c si $\text{error}_D(h,c)$ es pequeño.

Definición 1.3.29. Sea $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ una clase de conceptos y sea $\mathcal{H} = \{\mathcal{H}_n\}_{n \in \mathbb{N}}$ una clase de hipótesis. $\mathcal{C}$ es *PAC aprendible en términos de $\mathcal{H}$* si existe algún algoritmo PAC A tal que,

- para todo $n \in \mathbb{N}$,
- para todo concepto objetivo $c \in \mathcal{C}_n$,
- para cada distribución de probabilidad D en el espacio de instancias $\{0,1\}^n$,
- para todo ϵ y todo δ, donde $0 < \epsilon, \delta < 1$,

si le damos al algoritmo A en una entrada (n, ϵ, δ) ejemplos aleatorios independientes de $\{0,1\}^n$ obtenidos de acuerdo con la distribución D y, le damos la información de si cada ejemplo está o no en c, entonces, con probabilidad al menos $1 - \delta$, A devuelve una hipótesis $h \in \mathcal{H}_n$ de modo que $\text{error}_D(h,c) \leq \epsilon$.
Más aún, el tiempo de ejecución de A estará acotado por un polinomio en n, $1/\epsilon$, $1/\delta$ y $\mathbf{size}_n(c)$.
$\mathcal{C}$ se dice *PAC aprendible* si $\mathcal{C}$ se puede aprender en términos de alguna clase $\mathcal{H}$. Además, se dice que $\mathcal{C}$ es *propiamente PAC aprendible* si $\mathcal{C}$ es aprendible en términos de $\mathcal{C}$.

La idea detrás de esta definición es que los algoritmos de aprendizaje de tipo PAC deben procesar los ejemplos en tiempo polinómico, es decir, deben ser computacionalmente eficientes y deben ser capaces de devolver una buena aproximación del concepto objetivo con alta probabilidad usando sólo un número razonable de ejemplos.

Notar que la eficiencia del algoritmo PAC se mide en función de sus parámetros relevantes: tamaño de los ejemplos (n), tamaños del concepto objetivo ($\mathbf{size}_n$), $1/\epsilon$, y $1/\delta$ (ver [30, 7, 31] para más detalles).

Nota 1.3.30. Es importante resaltar que la definición anterior lleva involucrada la representación. Es claro que para el mismo concepto puede haber representaciones muy diferentes y en particular, estas

representaciones pueden ser de tamaño muy diferentes. Esto significa que usar una representación u otra puede significar usar más tiempo. Como el tiempo de ejecución de los algoritmos PAC son polinómicos en el tamaño de la representación, conceptos con representaciones largas como $\Theta(2^n)$ son trivialmente aprendibles en la mayoría de los casos.

Ejemplo 1.3.31. [30] Existen un buen número de resultados en torno a la noción de propiamente PAC aprendible y PAC aprendible. A continuación se presenta un resumen de algunos de estos resultados. Los resultados negativos se han basado en la hipótesis de que RP$\neq$ NP [51].

1. Conceptos en forma de conjunciones son propiamente PAC aprendibles [94], sin embargo la clase de conceptos en forma de disyunciones de dos conjunciones no es propiamente PAC aprendible [51], y tampoco la clase de conceptos en forma de existenciales de conjunciones.
2. Los conceptos de umbral lineal (perceptrones) son propiamente PAC aprendibles [12], pero la clase de conceptos en forma de conjunción de dos umbrales lineales no es PAC aprendible. [1]. Del mismo modo, también los umbrales lineales de umbrales lineales (es decir, perceptrones multicapa con unidades ocultas) son propiamente PAC aprendibles, sin embargo la clase de conceptos en forma de disyucción de dos de ellos no es PAC aprendible. Por último, si los pesos se restringen a 1 y 0 (pero el umbral es arbitrario), entonces los conceptos de umbral lineal no son PAC aprendibles [51].
3. Las clases de k-DNF, k-CNF y k-listas decisionales son propiamente PAC aprendibles para cada k fijo [93, 79].

La mayoría de las dificultades a la hora de demostrar que algo es propiamente PAC aprendible es debido a la dificultad computacional de encontrar una hipótesis en la forma particular especificada por la clase objetivo. Por ejemplo, mientras que las funciones umbral booleanas con pesos $0-1$ no son propiamente PAC aprendibles (a menos que RP=NP), sí que son PAC aprendibles por funciones umbral booleanas generales. De modo similar se pueden extender las clases de hipótesis de los ejemplos mencionados en 1, que no eran PAC aprendibles, convirtiéndose en clases PAC aprendibles [51, 29].

Aprendizaje basado en preguntas

En el modelo de aprendizaje PAC, el algoritmo de aprendizaje puede verse como pasivo en el sentido de que no puede decidir qué ejemplos verá durante la fase de entrenamiento (los ejemplos vienen dados de modo aleatorio de acuerdo con alguna distribución de probabilidad fija). Sin embargo, podría ser interesante permitir que el algoritmo de aprendizaje seleccionara algún ejemplo particular y preguntará si está o no en el concepto objetivo. Esta es la idea del aprendizaje basado en preguntas de pertenencia, que fue introducida por Valiant en [94] y que se define formalmente del siguiente modo:

- *Preguntas de pertenencia*, Mem(x): la entrada es una secuencia $x \in \{0,1\}^n$ y la salida es el valor del concepto objetivo c evaluado en x.

Sin embargo, este no es el único tipo de preguntas que podría hacer el algoritmo de aprendizaje. Existen otro tipo de preguntas que también podrían ser útiles para el algoritmo como las introducidas por Angluin en su modelo (modelo basado en preguntas) [6]. En la definición original, el

algoritmo de aprendizaje tiene acceso a un conjunto fijo de oráculos (expertos) que contestarán a algunos tipos de preguntas especificas sobre el concepto objetivo c. Estas preguntas pueden ser del tipo: pertenencia, equivalencia, subconjunto, superconjunto, disjunción y exahustividad. Esta tesis se centrará exclusivamente en preguntas de pertenencia y preguntas de equivalencia, estando estas últimas definidas del siguiente modo:

- *Preguntas de equivalencia*, Equ(h): la entrada es una representación de un concepto $h \in \mathcal{H}_n$ y la salida es SÍ, en caso de que h sea equivalente al concepto objetivo c, o NO en caso de que no sean equivalentes. En este último caso, se devolverá también un *contraejemplo* x que cumpla $c(x) \neq h(x)$.

Por lo tanto, el aprendizaje mediante preguntas de pertenencia o de equivalencia se define formalmente del siguiente modo.

Definición 1.3.32. Sea $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ una clase de conceptos y sea $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$ la clase de hipótesis. $\mathcal{C}$ es *aprendible en términos de* $\mathcal{H}$ *con preguntas de pertenencia (equivalencia)* si existe un algoritmo A tal que, para cada n y cada concepto $c \in \mathcal{C}_n$, A hace preguntas de pertenencia (equivalencia) sobre c y cuando para, devuelve una hipótesis $h \in \mathcal{H}_n$ que es equivalente al concepto objetivo, es decir, para todo x, $c(x) = h(x)$.

$\mathcal{C}$ se dice *aprendible de un modo eficiente mediante preguntas de pertenencia o equivalencia* si el tiempo de ejecución y el número total de preguntas hechas por A están acotados por un polinomio en n y en $\mathbf{size}_n(c)$.

Notar de nuevo que la elección de la representación es muy relevante para el aprendizaje basado en preguntas, en particular es importante el tamaño de la representación. Un concepto con representaciones del tamaño $\Theta(2^n)$ es trivialmente aprendible en la mayoría de los casos.

Aprendizaje on-line

Este modelo de aprendizaje es el *on-line mistake-bound model* propuesto por Littlestone [58], que considera el aprendizaje con ejemplos en una situación en la cual el objetivo es hacer el menor número posible de errores. En este modelo, no hay un conjunto separado de ejemplos de entrenamiento. El algoritmo de aprendizaje debe predecir la respuesta apropiada para cada ejemplo (es decir, predecir si es un ejemplo positivo o negativo), empezando con el primer ejemplo recibido. Después de hacer su predicción, al algoritmo se le facilita la información de si ésta ha sido correcta o no, de modo que pueda usar esta información para mejorar sus hipótesis. El algoritmo de aprendizaje continuará aprendiendo mientras siga recibiendo ejemplos; es decir, continuará examinando la información recibida en un esfuerzo por mejorar sus hipótesis. La evaluación del comportamiento del algoritmo se hace contando el número de errores que cometerá el algoritmo en el peor caso cuando esté aprendiendo un concepto de una clase de conceptos fijada. Esto corresponde con el aprendizaje frente a un adversario que fija el orden en que los ejemplos vienen dados. Podemos definir este modelo más formalmente como sigue.

Definición 1.3.33. Sea $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ una clase de conceptos.

1. Un *algoritmo de aprendizaje on-line* A es un algoritmo que recibe entradas de la forma (n, y, h) y devuelve 0 ó 1, donde $|y| = n$ es el ejemplo a predecir y $h = ((x_1, b_1) \dots (x_r, b_r))$ es la historia de los ejemplos recibidos previamente con sus respectivas respuestas.

2. Sea $n \in \mathbb{N}$, y un concepto $c \in \mathcal{C}_n$, el *número de errores cometidos por* A *en* c se define como sigue,
$$\text{Mist}(n, c, A) = \max_h \#\{x \mid |x| = n, A(n, x, h) \neq c(x)\}.$$

3. El *peor caso de número de errores cometidos por* A *en* $\mathcal{C}_n$ se define como
$$\text{Mist}(n, \mathcal{C}_n, A) = \max_{c\in\mathcal{C}_n} \text{Mist}(n, c, A).$$

4. La clase de conceptos $\mathcal{C}$ es *on-line aprendible con* $f(n)$ *errores* si existe un algoritmo de aprendizaje on-line A, que se ejecuta en tiempo polinómico en la longitud de la entrada y, para un numero infinito de n's, $\text{Mist}(n, \mathcal{C}_n, A) \leq f(n)$.

El siguiente resultado relaciona el aprendizaje on-line con el aprendizaje basado en preguntas de equivalencia y será de utilidad en el capítulo 6 de esta tesis.

Proposición 1.3.34. [58] Si $\mathcal{C}$ es aprendible mediante $f(n)$ preguntas de equivalencia, entonces $\mathcal{C}$ es on-line aprendible con $f(n)$ errores.

Notar que, en la proposición anterior, las cotas de tiempo en los algoritmos de aprendizaje son relevantes.

1.4. Dimensión y dimensión con escala en C

El concepto de dimensión en matemáticas tiene un significado muy amplio. La idea más intuitiva es la de dimensión topológica [43]. Básicamente, la dimensión topológica de un conjunto se refiere al número de parámetros necesarios para definir un punto dentro del conjunto. Por ejemplo, la dimensión de un plano es dos puesto que un punto del plano se puede definir usando sus dos coordenadas cartesianas.

Sin embargo, cuando se trata de conjuntos altamente irregulares (fractales) la dimensión topológica se convierte en un concepto contraintuitivo. Un ejemplo clásico de esto son las curvas de von Koch (figura 1.6). Estas curvas son conjuntos con dimensión topológica uno pero que pueden llegar a rellenar una porción del plano. Por tanto, una noción de dimensión que fuera acorde con la idea intuitiva de dimensión debería dar un valor más cercano al dos.

Esta nueva noción de dimensión empezó a ser formalizada por Mandelbrot [71, 72], que defendía la existencia de dimensiones fraccionarias que midieran, en algún sentido, el grado de complejidad e irregularidad de un objeto (o visto de otro modo, su eficacia para ocupar espacio). Utilizando herramientas matemáticas desarrolladas a principios del siglo XX, Mandelbrot fue capaz de establecer la

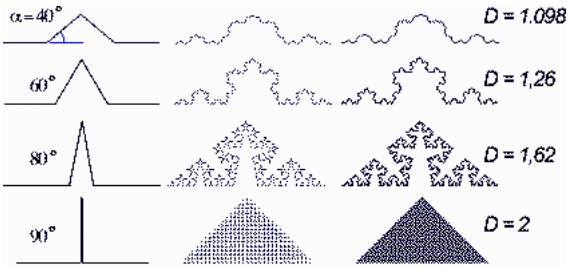

Figura 1.6: Dimensión de Hausdorff de las curvas de von Koch, dependiendo del ángulo.

manera idónea de medir fractales: utilizando la dimensión de Hausdorff (también llamada dimensión fractal) [28]. Desde entonces, ésta se ha convertido en la herramienta ideal de medición en geometría fractal y representa intuitivamente la complejidad del objeto que mide.

Desde un punto de vista formal, la definición de dimensión de Hausdorff se hace a través de la teoría de la medida y es posible definirla en cualquier espacio métrico y no sólo en el espacio Euclídeo. En particular, es posible definir dimensión de Hausdorff en el espacio de las secuencias infinitas de ceros y unos (espacio de Cantor). Tal como se ve en la Sección 1.3, es posible identificar las secuencias del espacio de Cantor con problemas decisionales (lenguajes) y por lo tanto, se puede utilizar la dimensión de Hausdorff como una herramienta para trabajar en Complejidad Computacional.

Sin embargo, quizá lo más interesante de la dimensión es que, tal como ocurre en geometría fractal, proporciona una idea de complejidad, en este caso, complejidad en el espacio de Cantor.

En esta sección se definirá la herramienta fundamental de esta tesis: la dimensión de recursos acotados. Más concretamente, se definirán diversas dimensiones dentro del espacio de Cantor C, conservando la idea intuitiva de la dimensión de Hausdorff, pero permitiendo su cálculo con distintos recursos [67, 21, 8]. Dependiendo de estos recursos de cálculo, se obtendrán dimensiones acotadas, por ejemplo, en tiempo o en espacio. Además se definirá la dimensión con escala, [39], como un refinamiento de la dimensión que permitirá estudiar más profundamente algunas clases de complejidad y su estructura.

1.4.1. Dimensión en C

La definición original de la dimensión de Hausdorff tiene sentido en cualquier espacio métrico y se trata de una definición muy sofisticada (ver, por ejemplo, [24]). Esto hace que el manejo, e incluso el cálculo, de la dimensión pueda resultar muy complejo en determinados casos. Sin embargo, en 2000, Lutz encontró una caracterización alternativa de la dimensión de Hausdorff, más sencilla e intuitiva, que puede ser aplicada tanto en el espacio de Cantor [67] como en el Euclídeo [70]. Esta definición será la que se utilizará en esta tesis.

La definición de dimensión proporcionada por Lutz está basada en la noción de martingala. Las

martingalas fueron utilizadas por primera vez en Informática Teórica en los años 70 por Schnorr [83, 84, 85, 86] en su investigaciones sobre la noción de aleatoriedad proporcionada por Martin-Löf [73]. Posteriormente fueron utilizadas por Lutz para definir la medida de recursos acotados [62, 66] y por último se generalizaron para definir dimensión.

El concepto de martingala está íntimamente ligado a los juegos de azar: la idea fundamental de una martingala no es otra que la de un juego limpio, tal como se verá más adelante.

Definición 1.4.1. Una *martingala* es una función $d : \{0,1\}^* \to [0,\infty)$ que verifica, para todo $w \in \{0,1\}^*$, la siguiente condición:

$$\frac{d(w0) + d(w1)}{2} = d(w). \tag{1.4.1}$$

Una forma intuitiva de entender este concepto sería plantear un escenario de un juego con un único jugador contra la banca. El jugador tiene una cierta estrategia para apostar sobre los sucesivos bits de una secuencia $A \in \mathbf{C}$. El valor de la martingala $d(w)$ representa la cantidad de dinero obtenido con dicha estrategia después de haber apostado sobre el prefijo w de A.

Intuitivamente, la noción de martingala representa un "juego limpio" en cuanto a la manera en que la banca realiza los pagos (doble o nada). Esto se deduce de la condición 1.4.1 puesto que la parte izquierda de la ecuación representa exactamente la esperanza matemática de $d(wb)$ cuando el siguiente bit de A se elige aleatoriamente según la distribución uniforme. Es decir, la fórmula dice que si A es aleatoria, no se espera ni ganar ni perder: la cantidad de dinero que se tiene es exactamente la cantidad de dinero que se espera tener en la siguiente jugada.

Basándose en esta misma idea, se define el concepto de s-gala, donde el parámetro s representará una medida de lo "limpio" que es el juego en cuanto a los pagos.

Definición 1.4.2. [67] Sea $s \in [0,\infty)$. Una *s-gala* es una función $d : \{0,1\}^* \to [0,\infty)$ que verifica, para todo $w \in \{0,1\}^*$, la siguiente condición:

$$\frac{d(w0) + d(w1)}{2^s} = d(w). \tag{1.4.2}$$

En el caso de que $s < 1$ (y este es el caso que va a resultar interesante) el pago será en favor a la banca. En efecto, al fijarse en la condición 1.4.2

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

se deduce que la cantidad de dinero que se tiene es mayor que la cantidad que se espera tener tras la siguiente apuesta. Además, cuanto más pequeña sea la s más injustos serán los pagos por parte de la banca.

Aun así, según como sea la secuencia A, es posible ganar dinero aunque el pago de la banca no sea justo. Por ejemplo, si se sabe de antemano que la secuencia A es una secuencia infinita de ceros,

se puede apostar todo el capital al cero y siempre se ganará dinero, por muy injusto que sea el pago de la banca.

Relacionada con la noción de ganancia está la siguiente definición.

Definición 1.4.3. [67] Sea d una s-gala, donde $s \in [0, \infty)$. Decimos que

1. *d tiene éxito en un lenguaje $A \in \mathbf{C}$* cuando
$$\limsup_{n\to\infty} d(A[0 \dots n-1]) = \infty.$$
2. El *conjunto de éxito de d* es
$$S^\infty[d] = \{A \in \mathbf{C} \mid d \text{ tiene éxito en } A\}.$$

Es decir, dada una s-gala, su conjunto de éxito son aquellas secuencias para las cuales se llega a ganar dinero ilimitado tras apostar "infinitas" veces, pese a que el pago de la banca no sea justo.

Con todo esto, se está en disposición de definir dimensión de Hausdorff y dar una interpretación de su significado. Se utilizará directamente como definición la caracterización de Lutz [67].

Definición 1.4.4. La *dimensión de Hausdorff de un conjunto $X \subseteq \mathbf{C}$* se define como
$$\dim_{\mathrm{H}}(X) = \text{ínf}\{s \in [0, \infty) \mid \exists\, d\, s\text{-gala tal que } X \subseteq S^\infty[d]\}.$$

Así pues, la dimensión de Hausdorff no es más que la medida de hasta que punto pueden ser injustos los pagos de la banca y aun así poder ganar "infinito" dinero con las secuencias de la clase X.

Esta interpretación esta relacionada con la idea intuitiva que en matemática clásica tiene la dimensión de un conjunto: su complejidad. Intuitivamente, si las secuencias de la clase no son muy complejas (dimensión pequeña), es posible predecir cuáles son sus bits y por tanto elaborar una estrategia que permita apostar correctamente y ganar dinero por muy desfavorables que sean los pagos.

1.4.2. Dimensión efectiva

Hasta el momento, se ha visto que la dimensión de Hausdorff en el espacio de Cantor dependía de la existencia de una estrategia que permita ganar dinero, aún con pagos desfavorables por parte de la banca. Ahora bien, que exista una estrategia no significa que ésta sea sencilla de calcular.

En la caracterización de dimensión de Hausdorff proporcionada por Lutz no existe ninguna restricción sobre las estrategias para ganar dinero (es decir, sobre las s-galas). Sin embargo, al fijarse en secuencias calculables, o incluso eficientemente calculables, parece natural exigir que también las estrategias para apostar puedan ser calculadas de un modo efectivo (más aún cuando la dimensión de Hausdorff es siempre trivial para conjuntos contables como el de las secuencias calculables). De esta idea surge la definición de dimensión efectiva.

Definición 1.4.5. Sea X un subconjunto de $\mathbf{C}$.

1. La *dimensión constructiva* de X (introducida en [68]) se define como

$$\mathrm{cdim}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala constructiva tal que } X \subseteq S^{\infty}[d]\},$$

   donde por s-gala constructiva se entiende que d sea semicalculable por debajo (ver 1.3.2).

2. Sea Δ cualquiera de las cotas de recursos definidos en 1.3.1. La Δ-*dimensión* de X (introducida en [67]) se define como

$$\dim_{\Delta}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala } \Delta\text{-calculable tal que } X \subseteq S^{\infty}[d]\}.$$

   La Δ-dimensión es especialmente útil para estudiar clases de complejidad dentro de $R(\Delta)$. Se define la *dimensión de X en $R(\Delta)$* como

$$\dim(X|R(\Delta)) = \dim_{\Delta}(X \cap R(\Delta)).$$

Nota 1.4.6. Es bien conocido que la dimensión de Hausdorff y las distintas definiciones de dimensión efectiva admiten otras definiciones equivalentes:

1. Se pueden sustituir en la definición las s-galas por s-supergalas [67, 35]. Una s-supergala no es más que una función $d : \{0,1\}^* \to [0,\infty)$ tal que verifica la siguiente condición:

$$d(w) \geq \frac{d(w0) + d(w1)}{2^s}. \tag{1.4.3}$$

2. También es equivalente definir la dimensión sustituyendo el conjunto $S^{\infty}[d]$ por el conjunto

$$S^{1}[d] = \{A \in \mathbf{C} \mid (\exists^{\infty} n)\, d(A[0 \ldots n-1]) \geq 1\}.$$

3. Tal como se verá en secciones posteriores, la dimensión se puede caracterizar mediante predicción y otros conceptos basados en teoría de la información.

Se utilizará una u otra definición según interese.

1.4.3. Dimensión de estados finitos

Desde 2000 numerosos autores han utilizado las diferentes dimensiones de la Definición 1.4.5 obteniendo interesantes resultados en Complejidad Computacional y Teoría de la Información (ver [78] para un resumen reciente). Además, se han definido también (utilizando otras restricciones sobre las s-galas) nuevas dimensiones como las que se ven por ejemplo en [21, 23, 2, 22]. Entre todas éstas, se ha utilizado en el desarrollo de esta tesis la dimensión de estados finitos [21], que requiere recursos de cálculo mucho menores que las dimensiones definidas en la Definición 1.4.5.

En la dimensión de estados finitos, las s-galas se definirán a partir de martingalas que se puedan calcular utilizando las máquinas apostadoras de estados finitos (FSG) vistas en la Subsección 1.3.7.

Definición 1.4.7. Sea $G = (Q, \delta, \beta, q_0)$ un FSG.

1. La *martingala de* G es la función $d_G : \{0,1\}^* \to [0, \infty)$ definida mediante la siguiente recursión:

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

   para todo $w \in \{0,1\}^*$ y $b \in \{0,1\}$.

2. Dado $s \in [0, \infty)$, una *s-gala de estados finitos* d es una s-gala para la cual existe un FSG, G, tal que

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

Así pues, se define la dimensión de estados finitos utilizando la restricción a s-galas de estados finitos.

Definición 1.4.8. [21] Sea X un subconjunto de $\mathbf{C}$. La *dimensión de estados finitos* de X se define como

$$\dim_{\mathrm{FS}}(X) = \text{ínf}\{s \in [0, \infty) \mid \exists\, d\, s\text{-gala de estados finitos tal que } X \subseteq S^\infty[d]\}.$$

1.4.4. Dimensión fuerte

La dimensión fuerte o empaquetadora (*Packing*) se desarrolló independientemente por Tricot [90] y Sullivan [89] como una definición alternativa a la idea intuitiva de dimensión y, junto con la dimensión de Hausdorff, es una de las herramientas más potentes para trabajar en geometría fractal.

Sorprendentemente, aunque la definición formal de la dimensión *Packing* es todavía más compleja que la definición clásica de Hausdorff, es posible obtener una caracterización dual a la caracterización de la dimensión de Hausdorff e igual de sencilla [8] (compárense las definiciones correspondientes). Más concretamente,

Definición 1.4.9. Sea d una s-gala, donde $s \in [0, \infty)$. Decimos que

1. *d tiene éxito fuertemente en un lenguaje* $A \in \mathbf{C}$ cuando

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

2. El *conjunto de éxito fuerte de* d es

$$S^\infty_{\mathrm{str}}[d] = \{A \in \mathbf{C} \mid d \text{ tiene éxito-fuertemente en } A\}.$$

Utilizando estos conceptos se puede caracterizar la dimensión *Packing* [21] y, tal como se hace en la Definición 1.4.5 y en la Definición 1.4.8, definir lo que se denominará *dimensión* Packing *efectiva* y *dimensión* Packing *de estados finitos*. Se utilizará directamente como definición la caracterización de dimensión *Packing* de [21].

Definición 1.4.10. Sea X un subconjunto de $\mathbf{C}$.

1. La dimensión *Packing* de X se define como

$$\mathrm{Dim}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala tal que } X \subseteq S^{\infty}_{\mathrm{str}}[d]\}.$$

2. La *dimensión* Packing *constructiva* de X se define como

$$\mathrm{cDim}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala constructiva tal que } X \subseteq S^{\infty}_{\mathrm{str}}[d]\}.$$

3. Sea Δ cualquiera de las cotas de recursos definidos en 1.3.1. La Δ-*dimensión* Packing de X se define como

$$\mathrm{Dim}_{\Delta}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala } \Delta\text{-calculable tal que } X \subseteq S^{\infty}_{\mathrm{str}}[d]\}.$$

   Se define la *dimensión* Packing *de* X *en* $R(\Delta)$ como

$$\mathrm{Dim}(X|R(\Delta)) = \mathrm{Dim}_{\Delta}(X \cap R(\Delta)).$$

4. La *dimensión* Packing *de estados finitos* de X se define como

$$\mathrm{Dim}_{\mathrm{FS}}(X) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s\text{-gala de estados finitos tal que } X \subseteq S^{\infty}_{\mathrm{str}}[d]\}.$$

1.4.5. Dimensión con escala en C

Las distintas definiciones de dimensión vistas hasta el momento proporcionan un amplio abanico de herramientas para diferenciar y estudiar múltiples clases de complejidad aunque, sin embargo, todavía existen clases para las cuales estas dimensiones no son útiles. El motivo es que muchas de las clases que aparecen de forma natural en complejidad computacional no son compatibles con la escala lineal que va implícita en la dimensión de Hausdorff clásica y en la dimensión con recursos acotados. La solución, al igual que en la teoría clásica de dimensión de Hausdorff, es la de introducir nuevas escalas que permitan atrapar clases que estén parametrizadas de un modo no lineal [39]. Por ejemplo, las clases de circuitos de tamaño acotado por $2^{\alpha n}$ con $0 < \alpha < 1$ (interesantes en criptografía), tienen todos idéntica dimensión 0 en ESPACE sin embargo, con una dimensión con escala, se consigue reconocer la información del parámetro α. Para definir la dimensión con escala se utilizarán las llamadas funciones escala que se definen a continuación.

Definición 1.4.11. [39] Una *función escala* es una función continua $g : H \times [0,\infty) \to \mathbb{R}$ que verifica las siguientes propiedades:

1. $H = (a,\infty)$ para algún $a \in \mathbb{R} \cap \{-\infty\}$.
2. $g(m,1) = m$ para todo $m \in H$.
3. $g(m,0) = g(m',0) \geq 0$ para todos los $m, m' \in H$.

4. La función $s \mapsto g(m,s)$ es una función no negativa y estrictamente creciente para $m \in H$ suficientemente grande.

5. Para todos los $s' > s \geq 0$,
$$\lim_{m\to\infty}[g(m,s') - g(m,s)] = \infty.$$

Notación 1.4.12. Para cada función escala $g : H \times [0,\infty) \to \mathbb{R}$, se define la función $\Delta g : H \times [0,\infty) \to \mathbb{R}$ como
$$\Delta g(m,s) = g(m+1,s) - g(m,s).$$
Para $l \in \mathbb{N}$ se usa la notación extendida
$$\Delta^l g(m,s) = g(m+l,s) - g(m,s).$$

Existe una familia de funciones escala que por su definición son especialmente interesantes para trabajar con dimensión. Esta familia fue introducida en [39] y se utilizará a lo largo de esta tesis.

Definición 1.4.13. Para cada $k \in \mathbb{Z}$, definimos la *función escala de orden k*, $g_k : H_k \times [0,\infty) \to \mathbb{R}$ mediante la siguiente recursión:

i) $g_0(m,s) = ms$.

ii) Para $k \geq 0$,
$$g_{k+1}(m,s) = 2^{g_k(\log m, s)}.$$

iii) Para $k < 0$,
$$g_k(m,s) = \begin{cases} m + g_{-k}(m,0) - g_{-k}(m,1-s) & \text{si } 0 \leq s \leq 1, \\ g_{-k}(m,s) & \text{si } s \geq 1. \end{cases}$$

El dominio de g_k es de la forma $H_k = (a_{|k|},\infty)$, donde $a_0 = -\infty$ y $a_{|k|+1} = 2^{a_{|k|}}$.

Definiendo la dimensión a partir de las funciones escala se obtiene un mayor grado de libertad. La definición se hace más concretamente a través de las s^g-galas, una generalización de las s-galas vistas hasta el momento.

Definición 1.4.14. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y $s \in [0,\infty)$. Una *s-gala g-escalada* (s^g-gala) es una función $d : \{0,1\}^* \to [0,\infty)$ que satisface la siguiente condición
$$d(w) = \frac{d(w0) + d(w1)}{2^{\Delta g(|w|,s)}} \tag{1.4.4}$$
para todo $w \in \{0,1\}^*$ con $|w| \in H$.

La idea intuitiva que está detrás de la definición de s^g-gala es la misma que en el caso de las martingalas y s-galas. En este caso, $\Delta g(|w|,s)$ representará una medida de lo "limpio" que es el juego dependiendo de s y del tiempo que se lleve jugando, es más, una s-gala no es más que un caso particular de s^g-gala cuando se considera $g = g_0$.

Notación 1.4.15. Cuando $g = g_k$ ($k \in \mathbb{Z}$) se escribirá $s^{(k)}$-gala en vez de s^{g_k}-gala.

Se definirá la noción de éxito de una s^g-gala y las diversas nociones de dimensión de un modo análogo a como se hace en el caso no escalado.

Definición 1.4.16. Sea g una función escala, $s \in [0, \infty)$ y d una s^g-gala.

1. *d tiene éxito en una secuencia $A \in \mathbf{C}$* si

$$\limsup_{n\to\infty} d(A[0 \dots n]) = \infty.$$

2. El *conjunto de éxito de d* es

$$S^\infty[d] = \{A \in \mathbf{C} \mid d \text{ tiene éxito en } A\}.$$

Definición 1.4.17. Sea $X \subseteq \mathbf{C}$ y g una función escala.

1. La *dimensión constructiva con escala g de X* es

$$\mathrm{cdim}^g(X) = \text{ínf}\{s \in [0, \infty) \mid \exists\, d\, s^g\text{-gala constructiva tal que } X \subseteq S^\infty[d]\}.$$

2. La *Δ-dimensión con escala g de X* es

$$\dim^g_\Delta(X) = \text{ínf}\{s \in [0, \infty) \mid \exists\, d\, s^g\text{-gala } \Delta\text{-calculable tal que } X \subseteq S^\infty[d]\}.$$

   La Δ-dimensión con escala g, al igual que en el caso no escalado, es especialmente útil para trabajar en clases de complejidad dentro de $R(\Delta)$. Se define la *dimensión de X en $R(\Delta)$* como

$$\dim^g(X|R(\Delta)) = \dim^g_\Delta(X \cap R(\Delta)).$$

De un modo dual, cambiando la condición $X \subseteq S^\infty[d]$ por $X \subseteq S^\infty_{\mathrm{str}}[d]$, se obtienen las definiciones correspondientes de dimensiones *Packing* con escalas.

Notación 1.4.18. Cuando $g = g_k$ ($k \in \mathbb{Z}$) se escribirá $\mathrm{cdim}^{(k)}$ y $\dim^{(k)}_\Delta$ en vez de cdim^{g_k} y $\dim^{g_k}_\Delta$.

Nota 1.4.19. Al igual que en el caso de dimensión sin escala, es bien conocido que estas definiciones admiten definiciones equivalentes en el siguiente sentido:

1. Se puede sustituir en la definición las s^g-galas por s^g-supergalas. Una s^g-supergala no es más que una función $d : \{0,1\}^* \to [0, \infty)$ tal que verifica la siguiente condición:

$$d(w) \geq \frac{d(w0) + d(w1)}{2^{\Delta g(|w|,s)}}. \tag{1.4.5}$$

2. Se puede sustituir en la definición el conjunto $S^\infty[d]$ por el conjunto

$$S^1[d] = \{A \in \mathbf{C} \mid (\exists^\infty n) d(A[0 \dots n-1]) \geq 1\}.$$

En esta tesis se utilizará una u otra definición según interese.

El siguiente diagrama nos da una visión general de las diferentes definiciones de dimensión vistas hasta el momento.

Los siguientes resultados sobre s^g-supergalas (s^g-galas) son bien conocidos (para más detalles ver [39]). Serán necesarios en diversos puntos de la tesis.

Lema 1.4.20. [39] Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y $m = \min(H \cap \mathbb{N})$. Sean $s \in [0,\infty)$ y $\alpha_k \in [0,\infty)$. Para cada $k \in \mathbb{N}$, sea d_k una s^g-gala.

i) Para cada $n \in \mathbb{N}$, $\sum_{k=0}^{n} \alpha_k d_k$ es una s^g-gala.

ii) Si $\sum_{k=0}^{\infty} \alpha_k d_k(w) < \infty$ para cada $w \in \{0,1\}^*$ con $|w| = m$, entonces $\sum_{k=0}^{\infty} \alpha_k d_k$ es una s-gala.

Lema 1.4.21. [39] Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y sea $s \in [0,\infty)$. Si d una s^g-supergala y $B \subseteq \{0,1\}^*$ es un conjunto prefijo, entonces para todo $w \in \{0,1\}^*$ con $|w| \in H$,

$$\sum_{u \in B} 2^{-\Delta^{|u|} g(|w|,s)} d(wu) \leq d(w).$$

Lema 1.4.22. [39] Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala, sea $s \in [0,\infty)$ y d una s^g-supergala. Para todo $w, u \in \{0,1\}^*$ con $|w| \in H$,

$$d(wu) \leq 2^{\Delta^{|u|} g(|w|,s)} d(w).$$

Lema 1.4.23. [39] Sean g, g' funciones escala y d una s^g-gala. Entonces la función

$$d'(w) = 2^{g'(|w|,s) - g(|w|,s)} d(w)$$

es una $s^{g'}$-gala.

1.5. Funciones escala: lemas técnicos

En esta sección se fijará la notación y algunos lemas técnicos de las funciones escala que serán necesarios en capítulos posteriores de la tesis. Se introducirán las funciones escala regulares que permiten demostrar propiedades más robustas de su correspondiente dimensión con escala e incluyen la mayoría de las funciones escala interesantes (por ejemplo la familia $\{g_k\}$).

Definición 1.5.1. Una *función escala regular* es una función escala calculable, $g : H \times [0, \infty) \to \mathbb{R}$, tal que verifica

1. g es continua y derivable en la segunda variable.
2. Existe una función $\alpha : H \to \mathbb{R}$ tal que,

   i) $\frac{\partial g}{\partial s}(m, s') \geq \alpha(m)$, para todo $s' \in [0, 1]$ y $m \in H$.

   ii) $\alpha(m)$ tiende a infinito cuando m tiende a infinito.
3. Para todo $m \in H$ suficientemente grande y $s' > s$ se verifica que

$$\triangle g(m, s') - \triangle g(m, s) > 0.$$

Además, se dice que una escala $g : H \times [0, \infty) \to \mathbb{R}$ es *estrictamente regular* si es función escala regular y verifica la siguiente propiedad:

4. Para todo $s' > s \geq 0$, con $m_g = \text{mín}(H \cap \mathbb{N})$,

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')} < \infty.$$

Notación 1.5.2. Cuando $g = g_k$ $(k \in \mathbb{Z})$ se escribirá m_k en vez de m_{g_k}.

Nota 1.5.3. La condición 3. en la definición anterior es equivalente a la siguiente condición:

3'. Para todo $m \in H$ suficientemente grande se verifica

$$g(m + 1, s') - g(m + 1, s) > g(m, s') - g(m, s).$$

Ejemplo 1.5.4. La función escala g_0 es una función escala estrictamente regular.

En efecto, es claro que g_0 es una función calculable y además

1. g_0 es derivable en la segunda variable con

$$\frac{\partial g_0}{\partial s}(m, s') = m.$$

2. $\alpha(m) = m$.

3. Para todo $m \in H$ suficientemente grande y $s' > s$ se verifica que
$$\triangle g_0(m, s') - \triangle g_0(m, s) = m(s' - s) > 0.$$

4. Para todo $s' > s \geq 0$,
$$\sum_{n=0}^{\infty} 2^{g_0(n,s)-g_0(n,s')} = \sum_{n=0}^{\infty} 2^{(s-s')n} < \infty.$$

Definición 1.5.5. Sea $g : H \times [0, \infty)$ una función escala. Se denotará por
$$f_g^m : [g(m, 0), \infty] \to [0, \infty)$$
a la función inversa de $s \mapsto g(m, s)$, es decir, la función definida como
$$f_g^m(x) = s \quad \text{si} \quad g(m, s) = x.$$

El siguiente ejemplo muestra la función inversa de $s \mapsto g_1(m, s)$, es decir $f_{g_1}^m$.

Ejemplo 1.5.6. Sea $g(m, s) = g_1(m, s) = m^s$, entonces $f_{g_1}^m(x) = \frac{\log x}{\log m}$. En efecto,
$$f_{g_1}^m(g_1(m, s)) = \frac{\log m^s}{\log m} = s.$$

Estas funciones estarán bien definidas para m suficientemente grandes, dado que $g(m, .)$ es estrictamente creciente en ese caso (ver Definición 1.4.11).

Las siguiente propiedades de las funciones f_g^m serán necesarias a lo largo de esta tesis.

Proposición 1.5.7. Sea $g : H \times [0, \infty) \to \mathbb{R}$ una función escala regular.

1. Para todo $m \in \mathbb{N}$ suficientemente grande, la función f_g^m es continua y estrictamente creciente.
2. La función f_g^m es derivable para todo $m \in H$.
3. Se cumple que
$$(f_g^m)'(x) \to_{m\to\infty} 0.$$

Demostración. 1. Al ser g una función escala, es continua y, para m's suficientemente grandes, la función $s \mapsto g(m, s)$ es estrictamente creciente. Así que, por el Teorema de continuidad de la función inversa, se tiene que f_g^m es continua y estrictamente creciente.

2. Al ser g función escala regular tenemos que es derivable en la segunda variable y por el Teorema de derivabilidad de la función inversa, f_g^m también es derivable.

3. Por ser g función escala regular, tenemos que, para m suficientemente grande y para todo $s' \in [0, 1]$, existe una función $\alpha : H \to \mathbb{R}$ tal que $\alpha(m) \to \infty$ cuando m tiende a infinito y que verifica $\frac{\partial g}{\partial s}(m, s') > \alpha(m)$.

   Así pues, para m suficientemente grande y para todo $s' \in [0, 1]$ tendremos que $\frac{\partial g}{\partial s}(m, s') \neq 0$. Por el Teorema de derivabilidad de la función inversa, si $s' = f_g^m(x)$, entonces
$$(f_g^m)'(x) = \frac{1}{\frac{\partial g}{\partial s}(m, s')} < \frac{1}{\alpha(m)},$$
   que tiende a 0 cuando m tiende a infinito. □

La familia $\{g_k\}_{k\in\mathbb{Z}}$ y propiedades

A continuación se muestran algunas de las propiedades de la familia de funciones $\{g_k\}_{k\in\mathbb{Z}}$ introducida en la sección anterior.

Proposición 1.5.8. [39] Las funciones de la familia $\{g_k\}_{k\in\mathbb{Z}}$ son funciones escalas.

Proposición 1.5.9. Las funciones de la familia $\{g_k\}_{k\in\mathbb{Z}}$ son funciones escalas regulares.

Demostración. Por la Proposición 1.5.8 ya sabemos que g_k, para $k \in \mathbb{Z}$, es función escala. Veamos que también es regular. Es claro que las funciones g_k son calculables.

Primero comprobaremos por inducción sobre i que g_i cumple las propiedades de función escala regular $\forall i \in \mathbb{N}$.

a) Para $i = 0$ ya se ha visto que g_0 es función escala regular (Ejemplo 1.5.4).

b) Suponer que g_i es regular, veamos que g_{i+1} también es regular.

1. Por ser g_i función escala regular, se tiene que g_i es derivable en la segunda variable. Sea $m \in H_{i+1}$, entonces $\log m \in H_i$ y aplicando la regla de la cadena:

$$\frac{\partial g_{i+1}}{\partial s}(m, s') = \frac{\partial g_i}{\partial s}(\log m, s')2^{g_i(\log m, s')} \ln 2 = \frac{\partial g_i}{\partial s}(\log m, s')g_{i+1}(m, s') \ln 2 \qquad (1.5.1)$$

   por lo que g_{i+1} es derivable en la segunda variable.

2. Veamos primero que la función $s' \mapsto \frac{\partial g_i}{\partial s}(m, s')$ es no decreciente para todo $m \in H_i$. Por inducción sobre i,

   i) para $i = 0$, $\frac{\partial g_0}{\partial s}(m, s') = m$, luego constante y por lo tanto no decreciente.

   ii) Suponer que es cierto para g_i, entonces para todo $m \in H_{i+1}$ se tiene que $\log m \in H_i$ y por tanto la función $s \mapsto \frac{\partial g_i}{\partial s}(\log m, s)$ es no decreciente. Entonces, por la igualdad (1.5.1) la función

$$s \mapsto \frac{\partial g_{i+1}}{\partial s}(m, s)$$

   es también no decreciente.

   Así pues, al ser $s' \mapsto \frac{\partial g_i}{\partial s}(m, s')$ no decreciente, tenemos que

$$\frac{\partial g_i}{\partial s}(m, s') \geq \frac{\partial g_i}{\partial s}(m, 0).$$

   Sólo queda ver que $\frac{\partial g_i}{\partial s}(m, 0)$ tiende a infinito cuando m tiende a infinito.

   De nuevo por inducción,

   i) Para $i = 0$, $\frac{\partial g_0}{\partial s}(m, 0) = m$, que tiende a infinito cuando $m \to \infty$.

   ii) Suponer que es cierto para g_i, entonces la función $m \mapsto \frac{\partial g_i}{\partial s}(\log m, 0)$ tiende a infinito cuando m tiende a infinito. Por la igualdad (1.5.1) y por ser g_{i+1} función escala,

$$m \mapsto \frac{\partial g_{i+1}}{\partial s}(m, 0) = \frac{\partial g_i}{\partial s}(\log m, 0)g_{i+1}(m, 0) \ln 2$$

   tiende a infinito cuando m tiende a infinito.

3. Veamos que se verifica la condición equivalente de la Nota 1.5.3. Para ello probaremos algo más fuerte, que la función $g_i(m,s') - g_i(m,s)$ es estrictamente creciente cuando $m > M_i$ para algún $M_i \in H_i$ suficientemente grande.

   Por inducción sobre i,

   i) Para $i = 0$,
   $$g_0(m,s') - g_0(m,s) = m(s'-s),$$
   luego es una función estrictamente creciente para todo $m > 0$.

   ii) Suponer que es cierto para g_i, sea M_{i+1} suficientemente grande para que la función $g_i(m,s') - g_i(m,s)$ sea estrictamente creciente para todo $m > \log M_{i+1}$, entonces
   $$\begin{aligned} g_{i+1}(m,s') - g_{i+1}(m,s) &= 2^{g_i(\log m, s')} - 2^{g_i(\log m, s)} \\ &= 2^{g_i(\log m, s)}[2^{g_i(\log m, s') - g_i(\log m, s)} - 1], \end{aligned}$$
   luego $g_{i+1}(m,s') - g_{i+1}(m,s)$ es una función estrictamente creciente en m para $m > M_{i+1}$.

   Así pues, en particular tenemos que para m's suficientemente grandes,
   $$g_{i+1}(m+1,s') - g_{i+1}(m+1,s) > g_{i+1}(m,s') - g_{i+1}(m,s).$$

Luego hemos demostrado que cuando $i \in \mathbb{N}$, las funciones g_i son funciones escalas regulares. Esto se utilizará a continuación para demostrar que, para los k negativos, g_k también es función escala regular. Sea $i \in \mathbb{N}$, veamos que g_{-i} cumple las condiciones de función escala regular.

1. Por ser g_i función escala regular, se tiene que g_i es derivable en la segunda coordenada. Sea $m \in H_{-i} = H_i$, entonces

   i) Si $s > 1$, entonces $g_{-i} = g_i$ y se tiene que
   $$\frac{\partial g_{-i}}{\partial s}(m,s) = \frac{\partial g_i}{\partial s}(m,s).$$

   ii) Si $0 \leq s \leq 1$ entonces por la regla de la cadena tenemos que
   $$\frac{\partial g_{-i}}{\partial s}(m,s) = \frac{\partial g_i}{\partial s}(m,1-s) \tag{1.5.2}$$

2. Para $0 \leq s \leq 1$ se tiene por las desigualdades 1.5.2 y 2 del caso $i \in \mathbb{N}$, que la función $s \mapsto \frac{\partial g_{-i}}{\partial s}(m,s)$ es decreciente. Por lo tanto,
   $$\begin{aligned} \frac{\partial g_{-i}}{\partial s}(m,s) &\geq \frac{\partial g_{-i}}{\partial s}(m,1) \\ &= \frac{\partial g_i}{\partial s}(m,0), \end{aligned}$$
   que tal como se ha visto en 2 del caso $i \in \mathbb{N}$, tiende a infinito cuando $m \to \infty$.

3. Distinguimos tres casos:

i) Si $s' > s \geq 1$, entonces

$$\triangle g_{-i}(m, s') - \triangle g_{-i}(m, s) = \triangle g_i(m, s') - \triangle g_i(m, s) > 0.$$

ii) Si $1 \geq s' > s$ entonces

$$\begin{aligned} \triangle g_{-i}(m, s') - \triangle g_{-i}(m, s) &= -g_i(m+1, 1-s') + g_i(m, 1-s') \\ &+ g_i(m+1, 1-s) - g_i(m, 1-s) \\ &= -\Delta g_i(m, 1-s') + \Delta g_i(m, 1-s) \\ &> 0. \end{aligned}$$

iii) Si $s' \geq 1$ y $s < 1$ entonces

$$\begin{aligned} \triangle g_{-i}(m, s') - \triangle g_{-i}(m, s) &= \triangle g_i(m, s') - (m+1) - g_i(m+1, 0) \\ &+ g_i(m+1, 1-s) + m + g_i(m, 0) - g_i(m, 1-s) \\ &= \triangle g_i(m, s') + \triangle g_i(m, 1-s) - 1 \\ &> 2\triangle g_i(m, 1) - 1 \\ &= 1. \end{aligned}$$

□

Por último se demostrará que g_k es estrictamente regular para todo $k \in \mathbb{Z}$. Es decir, que verifica que para todo $s' > s \geq 0$, con $m_k = \text{mín}(H_k \cap \mathbb{N})$,

$$\sum_{n=m_k}^{\infty} 2^{g_k(n,s) - g_k(n,s')} < \infty.$$

Para demostrarlo se necesitan las siguientes propiedades que también serán útiles más adelante. Se utilizará la siguiente notación.

Notación 1.5.10. Denotaremos por $g_k(., s) : H_k \to \mathbb{R}$ a la función $m \mapsto g_k(m, s)$.

Proposición 1.5.11. Sea $s > 0$. Entonces,

1. Para toda escala $i \geq 0$ se tiene que,
   $g_{i+1}(., s) \in o(g_i(., s))$ si $s < 1$,
   $g_i(., s) \in o(g_{i+1}(., s))$ si $s > 1$.
2. Para toda escala $k \in \mathbb{Z}$ y para todo $p > 0$, $(\log m)^p \in o(g_k(., s))$.
3. Para toda escala $i \geq 0$ y para todo $s \in (0, \infty)$ y $m \geq m_i$

$$g_i(g_i(m, s), \frac{1}{s}) = m.$$

Demostración. La demostración se hace por inducción sobre $i \in \mathbb{N}$.

1. Tenemos que distinguir dos casos:

a) Si $s < 1$ veamos que $g_{i+1}(.,s) \in o(g_i(.,s))$.

i) Cierto para $i = 0$. En efecto, como $s < 1$

$$\lim_{m\to\infty} \frac{g_1(m,s)}{g_0(m,s)} = \lim_{m\to\infty} \frac{m^s}{ms} = 0.$$

ii) Cierto para $i \in \mathbb{N}$ implica que es cierto para $i+1$. En efecto, como $g_{i+1}(.,s) \in o(g_i(.,s))$, tenemos que para todo $\epsilon > 0$ y para todo m suficientemente grande

$$g_{i+1}(\log m, s) < \epsilon g_i(\log m, s)$$

luego,

$$\begin{aligned} \frac{g_{i+2}(m,s)}{g_{i+1}(m,s)} &= 2^{g_{i+1}(\log m,s) - g_i(\log(m,s)} \\ &< 2^{(\epsilon-1)g_i(\log m,s)} \end{aligned}$$

que tiende a 0 cuando m tiende a infinito.

b) Si $s > 1$ veamos que $g_i(.,s) \in o(g_{i+1}(.,s))$.

i) Cierto para $i = 0$. En efecto, como $s > 1$,

$$\lim_{m\to\infty} \frac{g_0(m,s)}{g_1(m,s)} = \lim_{m\to\infty} \frac{ms}{m^s} = 0.$$

ii) Cierto para $i \in \mathbb{N}$ implica que es cierto para $i+1$. En efecto, como $g_i(.,s) \in o(g_{i+1}(.,s))$ tenemos que para $\epsilon > 0$ y para todo m suficientemente grande

$$g_i(\log m, s) < \epsilon g_{i+1}(\log m, s).$$

Luego,

$$\begin{aligned} \frac{g_{i+1}(m,s)}{g_{i+2}(m,s)} &= 2^{g_i(\log m,s) - g_{i+1}(\log m,s)} \\ &< 2^{(\epsilon-1)g_{i+1}(\log m,s)}, \end{aligned}$$

que tiende a 0 cuando m tiende a infinito.

2. Sea $i \geq 0$. Veremos dos casos: escalas positivas (g_i) y escalas negativas (g_{-i}).

a) En el caso de escalas positivas, veamos que para todo $p > 0$, $(\log m)^p \in o(g_i(.,s))$.

i) Cierto para $i = 0$. En efecto, para todo $p > 0$,

$$\lim_{m\to\infty} \frac{(\log m)^p}{ms} = 0.$$

ii) En efecto, sea $\epsilon > 0$ y m suficientemente grande tal que

$$(\log(\log m))^p < \epsilon g_i(\log m, s)$$

$$\begin{aligned}\frac{(\log m)^p}{g_{i+1}(m,s)} &= \frac{2^{\log(\log m)^p}}{2^{g_i(\log m,s)}} \\ &\leq 2^{(\log(\log m))^p - g_i(\log m,p)} \\ &< 2^{(\epsilon-1)g_i(\log m,p)}\end{aligned}$$

que tiende a 0 cuando m tiende a infinito.

b) En el caso de escalas negativas, veamos que para todo $p > 0$, $(\log m)^p \in o(g_{-i}(.,s))$. Cuando $s > 1$ es claro, puesto que $g_{-i}(m,s) = g_i(m,s)$.
Veamos que ocurre cuando $s \leq 1$. Utilizaremos para ello inducción sobre i.

i) Es claro que se cumple para $i = 0$.

ii) Cierto para i implica que es cierto para $i+1$. En efecto, notar que como

$$g_{i+1}(.,1-s) = o(g_i(.,1-s)),$$

se tiene que para m suficientemente grande, $g_{i+1}(m,1-s) \leq g_i(m,1-s)$, así pues,

$$\begin{aligned}\frac{(\log m)^p}{g_{-(i+1)}(m,s)} &= \frac{(\log m)^p}{m + c_{i+1} - g_{i+1}(m,1-s)} \\ &\leq \frac{(\log m)^p}{m + c_{i+1} - g_i(m,1-s)} \\ &= \frac{(\log m)^p}{g_{-i}(m,s) + c},\end{aligned}$$

que tiende a cero por la hipótesis de inducción.

3. Veamos que para todo $s \in (0,\infty)$ y $m \geq m_i$

$$g_i(g_i(m,s),\frac{1}{s}) = m.$$

i) Cierto para $i = 0$. En efecto,

$$g_0(g_0(m,s),\frac{1}{s}) = g_0(m,s)\frac{1}{s} = \frac{ms}{s} = m.$$

ii) Cierto para i implica cierto para $i+1$. En efecto,

$$\begin{aligned}g_{i+1}(g_{i+1}(m,s),\frac{1}{s}) &= 2^{g_i(\log g_{i+1}(m,s),\frac{1}{s})} \\ &= 2^{g_i(g_i(\log m,s),\frac{1}{s})} \\ &= 2^{\log m} = m.\end{aligned}$$

□

Lema 1.5.12. Para todo $k \in \mathbb{Z}$, $s > 0$ y $c > 0$,

$$\sum_{n=m_k}^{\infty} 2^{-cg_k(n,s)} < \infty$$

Demostración. Por la Proposición 1.5.11, $(\log m)^p \in o(g_k(.,s))$ para todo $p > 0$. Esto implica que

$$2^{-cg_k(m,s)} \in o(2^{-c(\log m)^p}).$$

Así que, es suficiente probar que, para algún $p > 0$, la serie

$$\sum_{n=m_k}^{\infty} 2^{-c(\log n)^p}$$

converge.

Ahora bien, como $2^{-c(\log n)^p} \leq n^{-cp}$ para todo $p > 1$, basta fijar $p > 1$ tal que $cp > 1$ para asegurar la convergencia de la serie. □

Proposición 1.5.13. $i \geq 0$ y $0 < s < s'$

1. $g_i(n, s') - g_i(n, s)$ es estrictamente creciente para todo $n \in H_i$ suficientemente grande.
2. $\frac{g_i(n,s')}{g_i(n,s)}$ es no decreciente en $n \in H_i$ suficientemente grande.
3. Existe una constante $0 < C = C(s', s) < 1$ tal que, para todo $n \in H_i$ suficientemente grande,

$$g_i(n, s') - g_i(n, s) \geq C \cdot g_i(n, s').$$

Demostración.

1. Este punto se ve en detalle en el punto 3 de la demostración de la Proposición 1.5.9.
2. Para el caso $i = 0$,

$$\frac{g_0(n, s')}{g_0(n, s)} = \frac{ns'}{ns} = \frac{s'}{s},$$

constante, luego no decreciente.

Para el caso $i > 0$ tenemos que

$$\frac{g_i(n, s')}{g_i(n, s)} = 2^{g_{i-1}(\log n, s') - g_{i-1}(\log n, s)},$$

que es estrictamente creciente por el apartado 1.

3. Utilizando el punto anterior,

$$\begin{aligned} g_i(n, s') - g_i(n, s) &= g_i(n, s')[1 - \frac{g_i(n, s)}{g_i(n, s')}] \\ &\geq g_i(n, s')[1 - \frac{g_i(m_i, s)}{g_i(m_i, s')}]. \end{aligned}$$

Luego $C(s', s) = 1 - \frac{g_i(m_i,s)}{g_i(m_i,s')}$. Notar que $0 < \frac{g_i(m_i,s)}{g_i(m_i,s')} < 1$, luego $0 < C < 1$.

□

Proposición 1.5.14. Sea $k \in \mathbb{Z}$. La función g_k es función escala estrictamente regular. Es decir, para todo $s' > s \geq 0$,

$$\sum_{n=m_k}^{\infty} 2^{g_k(n,s) - g_k(n,s')} < \infty.$$

Demostración. Veamos primero que se cumple para las escalas positivas. Sea $i > 0$. Por la Proposición 1.5.13 se tiene que existe M tal que, para todo $n > M$

$$g_i(n, s') - g_i(n, s) \geq g_i(n, s).$$

Por lo tanto

$$\sum_{n=m_k}^{\infty} 2^{g_k(n,s)-g_k(n,s')} = \sum_{n=m_k}^{M} 2^{g_k(n,s)-g_k(n,s')} + \sum_{n=M+1}^{\infty} 2^{-C\cdot g_k(n,s')},$$

que converge por el Lema 1.5.12.

Veamos ahora el caso negativo, sea $i \geq 0$

$$\begin{aligned}\sum_{n=m_i}^{\infty} 2^{g_{-i}(n,s)-g_{-i}(n,s')} &= \sum_{n=m_i}^{\infty} 2^{n+g_i(n,0)-g_i(n,1-s)-n-g_i(n,0)+g_i(n,1-s')} \\ &= \sum_{n=m_i}^{\infty} 2^{g_i(n,1-s')-g_i(n,1-s)} < \infty\end{aligned}$$

puesto que al ser $s < s'$, tenemos que $1 - s' < 1 - s$ y aplicando el resultado para escalas positivas tenemos la convergencia. □

La siguientes propiedades sobre las funciones escalas serán necesarias más adelante. Se utilizará la siguiente notación para mayor claridad en las demostraciones.

Notación 1.5.15. Sea $0 \leq s \leq 1$ fijo. Denotamos $g_i(m) = g_i(m, s)$ y $g_i'(.) = \frac{\partial g_i(m,s)}{\partial m}(.)$.

Proposición 1.5.16. Sea $0 \leq s \leq 1$. Entonces, para todo $i \in \mathbb{N}$, la función $g_i(., s)$ es cóncava para valores suficientemente grandes.

Demostración. Utilizando inducción sobre i:

i) Cierto para $i = 0$. En efecto, para ello basta ver que la segunda derivada es menor o igual que cero. En efecto, como $g_0'(m) = s$, entonces $g_0''(m) = 0$.

ii) Cierto para i implica cierto para $i + 1$. En efecto,

$$g_{i+1}'(m) = 2^{g_i(\log m)} g_i'(\log m) \frac{\ln 2}{m \ln 2},$$

luego

$$\begin{aligned} g_{i+1}''(m) &= \frac{2^{g_i(\log m)}}{m^2}[g_i'(\log m)^2 + \frac{g_i''(\log m)}{\ln 2} - g_i'(\log m)] \\ &= \frac{2^{g_i(\log m)}}{m^2}\left[g_i'(\log m)(g_i'(\log m) - 1) + \frac{g_i''(\log m)}{\ln 2}\right]. \end{aligned}$$

Por hipótesis de inducción se tiene que el signo de la segunda derivada es negativo. En efecto: $\frac{2^{g_i(\log m)}}{m^2}$ es positivo y los sumandos $g_i'(\log m)(g_i'(\log m) - 1)$ y $\frac{g_i''(\log m)}{\ln 2}$ son negativos cuando m es suficientemente grande. Así que $g_{i+1}''(m) \leq 0$ para todo m suficientemente grande y se tiene el resultado.

□

Proposición 1.5.17. Sean $a, b \in \mathbb{N}$ suficientemente grandes y $s \in [0,1]$. Entonces, para todo $i \in \mathbb{N}$, se tiene que,

$$g_i(a+b,s) \leq g_i(a,s) + g_i(b,s).$$

Demostración. Tal como se hace en la anterior proposición, utilizaremos la notación simplificada $g_n(m) = g_n(m,s)$. Suponer $a \leq b$ suficientemente grandes. Veamos que se cumple por inducción:

i) Cierto para $n = 0$.

$$g_0(a) + g_0(b) = as + bs = (a+b)s = g_0(a+b).$$

ii) Cierto para n implica que es cierto para $n+1$. Por el Teorema del Valor Medio:

$$g_{n+1}(a+b,s) - g_{n+1}(b,s) = \frac{\partial g_{n+1}}{\partial m}(\xi, s)[(a+b) - b],$$

con $\xi \in [b, a+b]$. Ahora bien, tal como hemos visto en la Proposición 1.5.16, se tiene que g''_{n+1} es negativa, luego g'_{n+1} es decreciente. Luego

$$g'_{n+1}(\xi) \leq g'_{n+1}(b) \leq g'_{n+1}(a),$$

y por lo tanto

$$\begin{aligned} g_{n+1}(a+b) - g_{n+1}(b) &\leq g'_{n+1}(a)a \leq [g_{n+1}(a)g'_n(\log a)\frac{1}{a}]a \\ &\leq g_{n+1}(a), \end{aligned}$$

donde la última desigualdad se deduce de que $g'_n(\log a) \leq 1$ cuando a es grande.

□

Proposición 1.5.18. Sean $0 \leq s < s' \leq 1$ y sea $i \in \mathbb{N}$. Entonces, $\forall \tilde{s} \leq s' - s$ se tiene que

$$g_i(n,s') - g_i(n,s) \geq g_i(n,\tilde{s}),$$

para n suficientemente grande.

Demostración. Veamos la demostración por inducción.

i) Para $i = 0$, es claro.

ii) Que sea cierto para i implica que es cierto para $i+1$. Sea N tal que $\forall n > N$ se tiene que

$$g_i(n,s') - g_i(n,s) \geq g_i(n,\tilde{s}).$$

En ese caso, para todo $n > 2^N$ se tiene que, $\log n > N$ y aplicando la hipótesis de inducción:

$$\begin{aligned} g_{i+1}(n,s') &= 2^{g_i(\log n, s')} \geq 2^{g_i(\log n, s) + g_i(\log n, \tilde{s})} \\ &\geq 2^{g_i(\log n, s)} + 2^{g_i(\log n, \tilde{s})} = g_{i+1}(n,s) + g_{i+1}(n,\tilde{s}), \end{aligned}$$

para n suficientemente grande.

□

Notación 1.5.19. Se denotará por f_k^m la inversa de $g_k(m,.)$ según la definición 1.5.5

Proposición 1.5.20. Dado $m \in H_k$, el valor de f_k^m es

i) Para $k \geq 0$,

$$f_k^m(s) = \frac{\log(\log(\overset{k}{\ldots}\log(s)..))}{\log(\log(\overset{k}{\ldots}(\log(m))..))}.$$

ii) Para $k < 0$,

$$\begin{aligned} f_k^m(s) &= 1 - \frac{\log(\log(\overset{|k|}{\ldots}\log(m + g_{|k|}(m,0) - s)..))}{\log(\log(\overset{|k|}{\ldots}(\log(m))..))} \qquad \text{si } 0 \leq s \leq 1, \\ f_k^m(s) &= \frac{\log(\log(\overset{|k|}{\ldots}\log(s)..))}{\log(\log(\overset{|k|}{\ldots}(\log(m))..))} \qquad \text{si } s > 1. \end{aligned}$$

Demostración. La demostración es simplemente una comprobación de que $f_k^m(g_k(m,s)) = s$ en todos los casos, ya que $g_k(m,.)$ es estrictamente creciente. □

Capítulo 2

Dimensión efectiva con escala en $\{0,1\}^*$

La dimensión de Hausdorff de un conjunto es intuitivamente una medida de la complejidad de dicho conjunto pero, ¿hasta que punto están relacionados las diversas dimensiones con recursos acotados y con escala y otras medidas de complejidad bien conocidas en la Teoría de la Información? En el Capítulo 3 se estudiará con detalle estas relaciones, sin embargo en el presente capítulo llegaremos a resultados previos para la dimensión con escala constructiva definiendo para ello la dimensión con escala de secuencias finitas.

La dimensión para secuencias finitas (dimensión en $\{0,1\}^*$) fue introducida por Lutz en [68]. Esta definición es una modificación de la definición de dimensión en $\mathbf{C}$ de forma que proporciona una medida de complejidad en $\{0,1\}^*$. Tal como se ha visto en el capítulo anterior, la dimensión en $\mathbf{C}$ se define a partir de las s-galas, que pueden interpretarse como la ganancia obtenida a partir de estrategias de juego para apostar en los diferentes bits (0 ó 1) de una secuencia infinita. En el caso de dimensión en $\{0,1\}^*$ la idea es utilizar termgalas, funciones similares a las s-galas que en este caso tienen como posibilidades para apostar el 0, el 1 ó bien apostar a que la secuencia termina. Lutz demuestra en [68] que la dimensión de una secuencia finita está intimamente relacionada con la complejidad de Kolmogorv, más concretamente demuestra que existe una constante c tal que para todo $w \in \{0,1\}^*$,

$$|K(w) - |w|\dim(w)| \leq c.$$

Además, demuestra que la dimensión constructiva de una secuencia infinita A no es más que el límite inferior de las dimensiones de sus prefijos, por lo que obtiene una demostración alternativa al teorema demostrado por Mayordomo [77] que relaciona dimensión constructiva en $\mathbf{C}$ y complejidad de Kolmogorv, estableciendo que para toda secuencia $A \in \mathbf{C}$,

$$\mathrm{cdim}(A) = \liminf_{n\to\infty} \frac{K(A[0\ldots n-1])}{n}.$$

El objetivo del presente capítulo es extender estos resultados al caso de dimensión con escala, de modo que se obtenga una relación entre dimensión constructiva con escala y complejidad de Kolmogorov, todo ello a través de una definción de dimensión con escala para secuencias finitas. Para adaptar la definición de dimensión con escala constructiva de una secuencia infinita A, es decir,

$$\mathrm{cdim}^{(g)}(A) = \text{ínf}\{s \in [0,\infty) \mid \exists\, d\, s^{(g)}\text{-gala constructiva tal que } A \subseteq S^{\infty}[d]\},$$

a una secuencia finita, se encuentran los siguientes problemas:

i) Se debe permitir apostar a que la secuencia termina. Este problema se soluciona trabajando en $\mathcal{T}$ (las secuencias finitas acabadas y los prefijos de estas) y sustituyendo la noción de s^g-gala por un concepto similar: las g-supertermgalas.

ii) Se debe reemplazar la condición "$d(A[0\ldots n-1])$ no esta acotada cuando $n \to \infty$" (es decir, $A \in S^{\infty}[d]$) por una condición adaptada al nuevo entorno finito.

iii) Se debe conseguir una definición consistente y esto será posible gracias a la existencia de una g-supertermgala constructiva óptima.

En este capítulo se resuelven estos problemas de un modo análogo a como se hace en [68] y se proporciona una definición de dimensión con escala en $\{0,1\}^*$ consistente que generaliza la definición de dimensión en $\{0,1\}^*$. También se amplian los resultados de [68] que relacionan la dimensión en $\{0,1\}^*$, la complejidad de Kolmogorov y la dimensión constructiva en C, a los casos de dimensión con escala. Por último, se define el concepto de termpredictor añadiendo la habilidad de predecir el final de una secuencia finita frente a la predicción estándar de los algoritmos on-line. Es decir, un termpredictor predice tanto el próximo bit como el punto final de la secuencia finita. A partir de esta definición es posible relacionar la dimensión con escala de secuencias finitas con la predicción "on-line", extendiendo parcialmente al caso de escalas los resultados de Hitchcock en [34], aunque en principio no parece posible alcanzar una caracterización exacta.

Los resultados de este capítulo se encuentran publicados en [60] y, conjuntamente con John Hitchcock y Elvira Mayordomo, en una versión extendida en [38].

2.1. Supertermgalas escaladas

Al igual que ocurre en la dimensión en $\mathbf{C}$, podemos extender la dimensión en $\{0,1\}^*$ a su versión con escalas. Para ello será necesario un concepto análogo al de $s^{(g)}$-gala, pero en este caso permitiendo apostar en 0, en 1 ó bien apostar a que la secuencia termina. El símbolo $\Box$ se utilizará para denotar esto último. El conjunto consistente en todas las secuencias acabadas (los elementos de $\{0,1\}^*\Box$) y

los prefijos de estas se denotará por $\mathcal{T}$,

$$\mathcal{T} = \{0,1\}^* \cup \{0,1\}^*\square.$$

Definición 2.1.1. Sea $s \in [0,\infty)$ y $g : H \times [0,\infty) \to \mathbb{R}$ una función escala. Una *s-supertermgala g-escalada* (ó s^g-supertermgala) es una función $d_g : \mathcal{T} \to [0,\infty)$ tal que,

i) $d_g(x) \leq 1$ para todo $x \in \mathcal{T}$ con $|x| \notin H$.

ii) Para todo $w \in \{0,1\}^*$ con $|w| \in H$,

$$d_g(w) \geq \frac{d_g(w0) + d_g(w1) + d_g(w\square)}{2^{\triangle g(|w|,s)}}. \tag{2.1.1}$$

Una $s^{(g)}$-supertermgala representa las ganancias de una estrategia que permite apostar en los sucesivos bits de una secuencia finita y también apostar a que la secuencia termina. A diferencia de las $s^{(g)}$-galas, donde el juego es infinitamente largo, el capital final de una $s^{(g)}$-supertermgala es $d(w\square)$. Por otro lado, lo "limpio" que es el juego depende del valor de s y del tiempo que se lleve jugando (dependencia en g y $|w|$).

Lema 2.1.2. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala, $d_g, d'_g : \mathcal{T} \to [0,\infty)$ y $s, s' \in [0,\infty)$. Si se verifica

$$2^{-g(|x|,s)} d_g(x) = 2^{-g(|x|,s')} d'_g(x)$$

para todo $x \in \mathcal{T}$ con $|x| \in H$, entonces

$$d_g \text{ es una } s^g\text{-supertermgala} \Leftrightarrow d'_g \text{ es una } (s')^g\text{-supertermgala.}$$

Como consecuencia del Lema 2.1.2, una 0^g-supertermgala determina una familia entera de s^g-supertermgalas y en esto se basa la siguiente definición.

Definición 2.1.3. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala.

1. Una *g-supertermgala* es una familia

$$d_g = \{d_g^{(s)} \mid s \in [0,\infty)\},$$

de modo que cada $d_g^{(s)}$ es una s^g-supertermgala y

$$2^{-g(|x|,s)} d_g^{(s)}(x) = 2^{-g(|x|,s')} d_g^{(s')}(x) \tag{2.1.2}$$

para todo $s, s' \in [0,\infty)$ y $x \in \mathcal{T}$ con $|x| \in H$.

2. Se dice que d_g es *constructiva* si $d_g^{(0)}$ es constructiva.

3. Una g-supertermgala constructiva $\tilde{d_g}$ es *óptima* si para cada g-supertermgala constructiva d_g existe una constante $\alpha > 0$ tal que

$$\tilde{d_g}^{(s)}(w\square) > \alpha\, d_g^{(s)}(w\square)$$

para todo $s \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$.

Para que la definición de dimensión con escala en $\{0,1\}^*$ proporcionada en la siguiente sección sea robusta se demuestra la existencia de una g-supertermgala constructiva óptima. Para ello son necesarios la siguiente definición y resultados.

Definición 2.1.4. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y p una medida de subprobabilidad en $\{0,1\}^*$. La *g-supertermgala inducida por p* es la familia

$$d_g[p] = \{d_g^{(s)}[p] \mid s \in [0,\infty)\},$$

donde cada $d_g^{(s)}[p]$ se define como

$$d_g^{(s)}[p](x) = 2^{g(|x|,s)} \sum_{\substack{v \in \{0,1\}^* \\ x \sqsubseteq v\square}} p(v)$$

para todo $s \in [0,\infty)$ y $x \in \mathcal{T}$ con $|x| \in H$.

El siguiente lema demuestra que $d_g[p]$ es una g-supertermgala.

Lema 2.1.5. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala. La g-supertermgala inducida por una medida de subprobabilidad p en $\{0,1\}^*$, $d_g[p]$, es una g-supertermgala. Además, si g es calculable y p es constructiva, $d_g[p]$ es constructiva.

Demostración. Sea $w \in \{0,1\}^*$ con $|w| \in H$ y $s \in [0,\infty)$, entonces

$$\begin{aligned}
& \left[d_g^{(s)}[p](w0) + d_g^{(s)}[p](w1) + d_g^{(s)}[p](w\square)\right] 2^{-\Delta g(|w|,s)} \\
= \;& 2^{g(|w|+1,s)-\Delta g(|w|,s)} \left[\sum_{\substack{v \in \{0,1\}^* \\ w0 \sqsubseteq v\square}} p(v) + \sum_{\substack{v \in \{0,1\}^* \\ w1 \sqsubseteq v\square}} p(v) + \sum_{\substack{v \in \{0,1\}^* \\ w\square \sqsubseteq v\square}} p(v)\right] \\
= \;& 2^{g(|w|,s)} \left[\sum_{\substack{v \in \{0,1\}^* \\ w \sqsubseteq v\square}} p(v)\right] \\
= \;& d_g^{(s)}[p](w).
\end{aligned}$$

Es claro por la definición de $d_g[p]$ que si g es calculable y p es constructiva, $d_g[p]$ es constructiva. □

Lema 2.1.6. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y d_g una 0^g-supertermgala. Para todo $u \in \{0,1\}^*$,

$$\sum_{w\in\{0,1\}^*} d_g(uw\square) \leq d_g(u).$$

Demostración. Veamos por inducción sobre n que

$$\sum_{w\in\{0,1\}^{<n}} d_g(uw\square) + \sum_{w\in\{0,1\}^{=n}} d_g(uw) \leq d_g(u) \tag{2.1.3}$$

para todo $n \in \mathbb{N}$.

i) Para $n = 0$ es trivial.

ii) Asumir que se cumple para n, veamos que se cumple para $n+1$:

$$\begin{aligned}
& \sum_{w\in\{0,1\}^{<n+1}} d_g(uw\square) + \sum_{w\in\{0,1\}^{n+1}} d_g(uw) \\
\leq\ & \sum_{w\in\{0,1\}^{<n}} d_g(uw\square) + \sum_{w\in\{0,1\}^{n}} d_g(uw\square) + \sum_{w\in\{0,1\}^{n}} [d_g(uw0) + d_g(uw1)] \\
\leq\ & \sum_{w\in\{0,1\}^{<n}} d_g(uw\square) + \sum_{w\in\{0,1\}^{n}} d_g(uw) \\
\leq\ & d_g(u).
\end{aligned}$$

Luego la desigualdad (2.1.3) se verifica para todo $n \in \mathbb{N}$. Ahora bien, al ser d_g una 0^g-supertermgala se tiene que

$$d_g(uw0) + d_g(uw1) + d_g(uw\square) \leq d_g(uw),$$

y en particular $d_g(uw\square) \leq d_g(uw)$. Así que podemos deducir de la desigualdad (2.1.3) que

$$\sum_{w\in\{0,1\}^{\leq n}} d_g(uw\square) \leq d_g(u)$$

para todo $n \in \mathbb{N}$ y por tanto

$$\sum_{w\in\{0,1\}^{*}} d_g(uw\square) \leq d_g(u).$$

□

Corolario 2.1.7. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala y $m_g = \text{mín}(H \cap \mathbb{N})$. Sea d_g una s^g-supertermgala. Para todo $u \in \{0,1\}^*$ y todo $n \geq m_g$,

$$\sum_{w\in\{0,1\}^n} d_g(uw\square) \leq 2^{\Delta^{n+1}g(|u|,s)} d_g(u).$$

Demostración. Por el Lema 2.1.2, tenemos que $d'_g(x) = 2^{-g(|x|,s)+g(|x|,0)} d_g(x)$ es una 0^g-termgala, así que por el lema anterior,

$$\begin{aligned}
\sum_{w\in\{0,1\}^n} d_g(uw\square) &= 2^{g(|u|+n+1,s)-g(|u|+n+1,0)} \sum_{w\in\{0,1\}^n} d'_g(uw\square) \\
&\leq 2^{g(|u|+n+1,s)-g(|u|+n+1,0)} d'_g(u) \\
&= 2^{g(|u|+n+1,s)-g(|u|+n+1,0)} 2^{-g(|u|,s)+g(|u|,0)} d_g(u) \\
&= 2^{\Delta^{n+1}g(|u|,s)} d_g(u).
\end{aligned}$$

□

El siguiente teorema demuestra que, debido a la existencia de una medida de subprobabilidad constructiva óptima, es posible encontrar una supertermgala constructiva óptima.

Teorema 2.1.8. Sea $\tilde{p}$ una medida de subprobabilidad constructiva óptima en $\{0,1\}^*$ y $g : H \times [0,\infty) \to \mathbb{R}$ una función escala calculable. Entonces, la g-termgala inducida por $\tilde{p}$ es una g-supertermgala constructiva óptima.

Demostración. Por el Lema 2.1.5, $d_g[\tilde{p}]$ es una g-supertermgala constructiva. Veamos que $d_g[\tilde{p}]$ es óptima, es decir, si consideramos $d_g = \{d_g^{(s)} \mid s \in [0,\infty)\}$ una g-supertermgala constructiva arbitraria, debemos encontrar una constante real $\alpha > 0$ tal que

$$d_g^{(s)}[\tilde{p}](w\square) > \alpha d_g^{(s)}(w\square)/N$$

para todo $s \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$.

Definimos $p : \{0,1\}^* \to [0,\infty)$ como

$$p(w) = d_g^{(0)}(w\square)/N$$

para todo $w \in \{0,1\}^*$, donde $N \in \mathbb{N}$ cumple $d_g^{(0)}(\lambda) \leq N$. Por el Lema 2.1.6 (considerando $u = \lambda$), p es una medida de subprobabilidad en $\{0,1\}^*$ y además p es constructiva por serlo d_g. Así pues,

como $\tilde{p}$ es óptima, se tiene que existe una constante real $\alpha' > 0$ tal que

$$\tilde{p}(w) > \alpha' p(w)$$

para todo $w \in \{0,1\}^*$.

Tenemos entonces que, para todo $s \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$,

$$\begin{aligned} d_g^{(s)}[\tilde{p}](w\square) &= 2^{g(|w|+1,s)}\tilde{p}(w) \\ &> 2^{g(|w|+1,s)}\alpha'\, p(w) \\ &= 2^{g(|w|+1,s)}\alpha'\, d_g^{(0)}(w\square)/N. \end{aligned}$$

Ahora, por el Lema 2.1.2,

$$d_g^{(0)}(w\square) = 2^{g(|w|+1,0)-g(|w|+1,s)} d_g^{(s)}(w\square)$$

y tenemos que

$$d_g^{(s)}[\tilde{p}](w\square) > 2^{g(|w|+1,0)}\alpha' d_g^{(s)}(w\square)/N.$$

Como para todo $n \in \mathbb{N}$, $g(n,0)$ es una constante tenemos que $d[\tilde{p}]$ es óptima (tomando $\alpha = 2^{g(n,0)}\alpha'/N$). □

El siguiente resultado es una adaptación del Lema 1.4.20 visto para s^g-galas y será útil en las secciones posteriores.

Lema 2.1.9. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala calculable, sea $m_g = \text{mín}(H \cap \mathbb{N})$. Para cada $n \in \mathbb{N}$, sea d_n una g-supertermgala constructiva y sea $\alpha_n \in [0,\infty)$. Entonces,

1. Para cada $m \in \mathbb{N}$, $\sum_{n=0}^{m-1} \alpha_n d_n$ es una g-supertermgala constructiva.
2. Si $\sum_{n=0}^{\infty} \alpha_n d_n(w) < \infty$ para cada $w \in \{0,1\}^{m_g}$, entonces $\sum_{n=0}^{\infty} \alpha_n d_n$ es una g-supertermgala constructiva.

Demostración.

1. Como la suma es finita, es trivial que se cumple la condición de g-supertermgala y que es constructiva.
2. En el caso de la suma infinita, la condición de g-supertermgala se cumple, pero también es necesario comprobar que la serie converge para todo $x \in \mathcal{T}$. Sin embargo, como cada d_n es

g-supertermgala constructiva, tenemos que para todo $w \in \{0,1\}^*$ y $b \in \{0,1,\square\}$,

$$d_n(wb) \leq 2^{\Delta g(|w|,s)} d_n(w).$$

Haciendo la recursión, para todo $u \in \mathcal{T}$, se tiene que

$$d_n(wu) \leq 2^{\sum_{l=0}^{|u|-1} \Delta g(|w|+l,s)} d_n(w).$$

Así que, si $x \in \mathcal{T}$ es tal que $x = wu$, con $w \in \{0,1\}^{m_g}$ y $u \in \mathcal{T}$, tenemos que

$$\begin{aligned} d(x) &= \sum_{n=0}^{\infty} \alpha_n d_n(wu) \\ &\leq \sum_{n=0}^{\infty} 2^{\sum_{l=0}^{|u|-1} \Delta g(|w|+l,s)} \alpha_n d_n(w) \\ &= 2^{\sum_{l=0}^{|u|-1} \Delta g(|w|+l,s)} \sum_{n=0}^{\infty} \alpha_n d_n(w). \end{aligned}$$

Luego, como $\sum_{n=0}^{\infty} \alpha_n d_n(w) < \infty$ para cada $w \in \{0,1\}^{m_g}$, entonces $\sum_{n=0}^{\infty} \alpha_n d_n$ es una g-supertermgala constructiva.

□

2.2. Dimensión con escala en $\{0,1\}^*$

En esta sección se define la dimensión con escala de una secuencia finita. Primero se definirá la dimensión dependiendo de una supertermgala constructiva fija. Posteriormente se demostrará que al considerar en la defincón una supertermgala constructiva óptima, esta definición es consistente.

Definición 2.2.1. Sea $g : H \times [0,\infty] \to \mathbb{R}$ una función escala y $w \in \{0,1\}^*$ con $|w| \in H$. Si d_g es una g-supertermgala constructiva, entonces la *g-dimensión de w relativa a d_g* es

$$\dim_{d_g}(w) = \text{ínf}\{s \in [0,\infty) \mid d_g^s(w\square) > 1\}.$$

Esta definición depende de la supertermgala que se considere, los siguientes resultados preparan para una definición de g-dimensión que no dependa de la supertermgala.

Proposición 2.2.2. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala regular, sea d_g una g-supertermgala constructiva arbitraria y $\tilde{d}_g$ una g-supertermgala constructiva óptima. Entonces existe $C > 0$ y una función $\beta : H \to \mathbb{R}$ verificando que $\beta(m)$ tiende a ∞ cuando $m \to \infty$, tales que

$$\dim_{\tilde{d}_g}(w) \leq \dim_{d_g}(w) + \frac{C}{\beta(|w|+1)}$$

para todo $w \in \{0,1\}^*$.

Demostración. Al ser $\tilde{d_g}$ una g-supertermgala constructiva óptima existe una constante real $\alpha > 0$ tal que

$$\tilde{d}_g^s(w\square) > \alpha\, d_g^s(w\square)$$

para todo $s \in [0,\infty)$, $w \in \{0,1\}^*$ con $|w| \in H$. Por otro lado, al ser g una función escala existe $m_0 \in \mathbb{N}$ tal que $\forall m > m_0$ las funciones $g(m,\cdot)$ son continuas y estrictamente crecientes. Definimos entonces $h(w) = s'$, donde s' es tal que

$$g(|w|+1, \dim_{d_g}(w) + s') - g(|w|+1, \dim_{d_g}(w)) = \log \frac{1}{\alpha}.$$

Veamos que

$$\dim_{\tilde{d_g}}(w) \leq \dim_{d_g}(w) + h(w) \tag{2.2.1}$$

para todo w con $|w| > m_0$. Para ello, sea $s = \dim_{d_g}(w) + h(w)$. Basta probar que $\tilde{d_g}^{(s)}(w\square) > 1$. En efecto,

$$\begin{aligned}
\tilde{d_g}^{(s)}(w\square) &> \alpha\, d_g^{(s)}(w\square) \\
&= \alpha\, 2^{g(|w|+1,s) - g(|w|+1, \dim_{d_g}(w))} d_g^{(\dim_{d_g}(w))}(w\square) \\
&\geq \alpha\, 2^{g(|w|+1,s) - g(|w|+1, \dim_{d_g}(w))} \\
&= 1.
\end{aligned}$$

Por otro lado, dado que g es una función escala regular, podemos aplicar el Teorema del Valor Medio. Es decir, existe $s' \in [\dim_{d_g}(w), \dim_{d_g}(w) + h(w)]$ tal que

$$g(|w|+1, \dim_{d_g}(w) + h(w)) - g(|w|+1, \dim_{d_g}(w)) = \frac{\partial g}{\partial s}(|w|+1, s')h(w). \tag{2.2.2}$$

Así que de 2.2.1 y 2.2.2 se tiene

$$h(w) = \frac{\log \frac{1}{\alpha}}{\frac{\partial g}{\partial s}(|w|+1, s')} \leq \frac{\log \frac{1}{\alpha}}{\beta(|w|+1)},$$

donde en la última desigualdad se aplica que al ser g funcion escala regular, existe β verificando las condiciones del enunciado y tal que para todo $0 \leq s' \leq 1$,

$$\frac{\partial g}{\partial s}(|w|+1, s') \geq \beta(|w|+1).$$

□

Corolario 2.2.3. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala regular. Sean $\tilde{d}_1$ y $\tilde{d}_2$ g-supertermgalas constructivas óptimas. Entonces existe una constante $C > 0$ y $\alpha : H \to \mathbb{R}$ verificando que $\alpha(m)$ tiende a infinito cuando $m \to \infty$, tales que

$$|\dim_{\tilde{d}_1}(w) - \dim_{\tilde{d}_2}(w)| \leq \frac{C}{\alpha(|w|+1)},$$

para todo $w \in \{0,1\}^*$.

Nota 2.2.4. En el Corolario 2.2.3 como función $\alpha(m)$ puede tomarse la función que aparece en la definición de función escala regular (Definición 1.5.1, apartado 2).

Como g es una función escala regular, $\alpha(m)$ tiende a ∞ cuando $m \to \infty$, y el Corolario 2.2.3 dice que si la definición de g-dimensión se basa en una g-supertermgala constructiva optima $\tilde{d}_g$, la elección particular de esta $\tilde{d}_g$ sólo contribuye $O(\frac{1}{|w|})$ al valor que toma la dimensión $\dim_{\tilde{d}_g}(w)$.

Así pues, se fijará una g-supertermgala construciva óptima $d_{g\,\square}$ y se definirá la g-dimensión de secuencias finitas como sigue.

Definición 2.2.5. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala regular. La *g-dimensión de una secuencia* $w \in \{0,1\}^*$ con $|w| \in H$ se define como

$$\dim_g(w) = \dim_{d_{g\,\square}}(w).$$

La idea intuitiva de esta definición es que la dimensión con escala mide hasta que punto pueden ser injustos los pagos en un juego y, aún así, que el capital final sea mayor que el capital inicial. Como en el caso de la dimensión constructiva en **C**, podemos utilizar una estrategia fija para apostar que viene dada por una g-supertermgala óptima. Elegir una g-supertermgala óptima es necesario para asegurar la consistencia de la definición y permite asegurar que el valor de la dimensión no cambiará sustancialmente aunque se utilice otra estrategia óptima para apostar.

2.3. Relación entre dimensión con escala en $\{0,1\}^*$ y en C

En esta sección se demuestra que la dimensión constructiva con escala de una secuencia infinita se caracteriza por la dimensión con escala de sus prefijos. Este resultado generaliza el obtenido en [68] donde se establece que la dimensión de una secuencia infinita se caracteriza por la dimensión de sus prefijos.

Teorema 2.3.1. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala regular y sea $A \in C$. Entonces,

$$\mathrm{cdim}_g(A) = \liminf_{n\to\infty} \dim_g(A[0\ldots n-1]).$$

Demostración. Para ver que

$$\mathrm{cdim}_g(A) \leq s_0 = \liminf_{n\to\infty} \dim_g(A[0\ldots n-1]),$$

sean s y s' números racionales cualesquiera tales que $s' > s > s_0$. Será suficiente probar que $\mathrm{cdim}_g(A) \leq s'$. Por la elección de $s > s_0$, existe un conjunto infinito $J \subseteq \mathbb{N}$ tal que para todo $n \in J$,

$$\dim_g(A[0\ldots n-1]) < s,$$

luego, por la definición de dimensión de una secuencia finita, para todo $n \in J$,

$$d_{g\Box}^{(s)}(A[0\ldots n-1]\Box) > 1.$$

Sea $d'_g : \{0,1\}^* \to [0,\infty)$ definida como

$$d'_g(w) = d_{g\Box}^{(s')}(w) + \frac{1}{2}\, d_{g\Box}^{(s')}(u\Box),$$

donde $u = w[0\ldots|w|-2]$.

Entonces, d'_g es constructiva por serlo $d_{g\Box}$ y además d'_g es una $(s')^g$-supergala:

$$\begin{aligned}
[d'_g(w0) + d'_g(w1)]2^{-\Delta g(|w|,s')} &= [d_{g\Box}^{(s')}(w0) + d_{g\Box}^{(s')}(w1) + d_{g\Box}^{(s')}(w\Box)]2^{-\Delta g(|w|,s')} \\
&\leq d_{g\Box}^{(s')}(w) \\
&\leq d'_g(w).
\end{aligned}$$

Ahora, para todo $n \in J$,

$$\begin{aligned}
d'_g(A[0\ldots n]) &= \tilde{d}_{g\Box}^{(s')}(A[0\ldots n]) + \frac{1}{2}\tilde{d}_{g\Box}^{(s')}(A[0\ldots n-1]\Box) \\
&\geq \frac{1}{2}\,\tilde{d}_{g\Box}^{(s')}(A[0\ldots n-1]\Box) \\
&= \frac{1}{2}\,2^{g(n,s')-g(n,s)}\tilde{d}_{g\Box}^{(s)}(A[0\ldots n-1]\Box) \\
&> \frac{1}{2}\,2^{g(n,s')-g(n,s)}
\end{aligned}$$

Como J es infinito, esto implica que $A \in S^\infty[d'_g]$, luego $\dim_g(A) \leq s'$.

Para ver que

$$\mathrm{cdim}_g(A) \geq \liminf_{n\to\infty} \dim_g(A[0\ldots n-1]),$$

sean s' y s números racionales cualesquiera tales que $s' > s > \dim_g(A)$. Basta ver que existen infinitos $n \in N$ para los cuales

$$\dim_g(A[0 \dots n-1]) \leq s'.$$

Como $s > \dim_g(A)$, existe una s^g-supergala constructiva, d_g, tal que $A \in S^\infty[d_g]$. Definimos $d'_g : \mathcal{T} \to [0,\infty)$ como

$$\begin{aligned} d'_g(w) &= d_g(w), \\ d'_g(w\square) &= [2^{\Delta g(|w|,s')} - 2^{\Delta g(|w|,s)}]d_g(w), \end{aligned}$$

para todo $w \in \{0,1\}^*$. Entonces d'_g es constructiva por serlo d_g y además es una $(s')^g$-termgala. En efecto,

$$\begin{aligned} [d'_g(w0) + d'_g(w1) + d'_g(w\square)]2^{-\Delta g(|w|,s')} &= [d_g(w0) + d_g(w1)]2^{-\Delta g(|w|,s)}2^{\Delta g(|w|,s)-\Delta g(|w|,s')} \\ &+ [2^{\Delta g(|w|,s')} - 2^{\Delta g(|w|,s)}]d_g(w)2^{-\Delta g(|w|,s')} \\ &\leq 2^{\Delta g(|w|,s)-\Delta g(|w|,s')}d_g(w) \\ &+ 2^{\Delta g(|w|,s')-\Delta g(|w|,s')}d_g(w) \\ &- 2^{\Delta g(|w|,s)-\Delta g(|w|,s')}d_g(w) \\ &= d_g(w). \end{aligned}$$

Luego, como d'_g es una $(s')^g$-supertermgala constructiva, si definimos para cada $t \in [0,\infty)$ las funciones $\tilde{d_g}^{(t)} : \mathcal{T} \to [0,\infty)$ como

$$\tilde{d_g}^{(t)}(x) = 2^{g(|x|,t)-g(|x|,s')}d'_g(x),$$

serán t^g-supertermgalas y la familia

$$\tilde{d_g} = \{\tilde{d_g}^{(t)} \mid t \in [0,\infty)\}$$

será una g-supertermgala constructiva.

Ahora, debido a la optimalidad de $d_{g\square}$, existe una constante real $\alpha > 0$ tal que para todo $t \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$,

$$d_{g\square}^{(t)}(w\square) > \alpha\, \tilde{d_g}^{(t)}(w\square).$$

Además, como $A \in S^{\infty}[d_g]$, existen infinitos $n \in N$ tales que

$$\alpha\,[2^{\triangle g(n,s')} - 2^{\triangle g(n,s)}]d_g(A[0\ldots n-1]) > 1.$$

Notar para esto último que, al ser g función escala regular, se tiene

$$2^{\triangle g(n,s')} - 2^{\triangle g(n,s)} = 2^{\Delta g(n,s)}2^{\triangle g(n,s')-\triangle g(n,s)} - 1 > 0.$$

Y por lo tanto, para infinitos $n \in H$ se tiene

$$\begin{aligned} d_{g\Box}^{(s')}(A[0\ldots n-1]\Box) &> \alpha\,\tilde{d_g}^{(s')}(A[0\ldots n-1]\Box) \\ &= \alpha\,[2^{\triangle g(n,s')} - 2^{\triangle g(n,s)}]d_g(A[0\ldots n-1]) > 1, \end{aligned}$$

luego, tal como queríamos demostrar,

$$\dim_g(A[0\ldots n-1]) \leq s'$$

para infinitos $n \in \mathbb{N}$.

□

2.4. Dimensión con escala en $\{0,1\}^*$ y complejidad de Kolmogorov

Al igual que en el caso de dimensión sin escala [68], es posible obtener una relación entre entre dimensión con escala de una secuencia finita y su complejidad de Kolmogorov. Este resultado permite dar una nueva caracterización de la dimensión con escala constructiva de una secuencia infinita en términos de la complejidad de Kolomogorov de sus prefijos, extendiendo así el resultado de [77].

Teorema 2.4.1. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala regular. Entonces existe $C > 0$ y una función $\alpha : H \to \mathbb{R}$ verificando que $\alpha(m)$ tiende a infinito cuando $m \to \infty$, tales que para todo $w \in \{0,1\}^*$ con $|w|$ suficientemente grande,

$$\left| f_g^{|w|+1}(K(w)) - \dim_g(w) \right| \leq \frac{C}{\alpha(|w|+1)},$$

donde $f_g^{|w|+1}$ es la función inversa de g definida en la Definición 1.5.5.

Demostración. Por el Teorema 1.3.11 existe una medida $\mathbf{m}$ de subprobabilidad óptima en $\{0,1\}^*$ tal que

$$\left| K(w) - \log \frac{1}{\mathbf{m}(w)} \right| \leq c, \tag{2.4.1}$$

para todo $w \in \{0,1\}^*$

Consideremos la g-supertermgala $d_g[\mathbf{m}]$ como en la Definición 2.1.4. Se tiene entonces que, para todo $s \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$,

$$\begin{aligned} d_g[\mathbf{m}]^{(s)}(w\square) > 1 &\Leftrightarrow 2^{g(|w|+1,s)}\mathbf{m}(w) > 1 \\ &\Leftrightarrow g(|w|+1,s) > \log \frac{1}{\mathbf{m}(w)}. \end{aligned} \tag{2.4.2}$$

Si $|w|$ es suficientemente grande, $\dim_g(w) < +\infty$ y existe $s_w \in [0,\infty)$ tal que

$$g(|w|+1, s_w) = \log \frac{1}{\mathbf{m}(w)}. \tag{2.4.3}$$

Por la definición de dimensión y por las ecuaciones (2.4.2), (2.4.3) se tiene que

$$\dim_{d_g[\mathbf{m}]}(w) = s_w. \tag{2.4.4}$$

Además, por las ecuaciones (2.4.1) y (2.4.3),

$$|K(w) - g(|w|+1, s_w)| = \left| K(w) - \log \frac{1}{\mathbf{m}(w)} \right| \leq c. \tag{2.4.5}$$

Por otro lado, aplicando primero que $g(m, f_g^m(x)) = x$ y seguidamente el Teorema del Valor Medio, tenemos que

$$\begin{aligned} |K(w) - g(|w|+1, s_w)| &= |g(|w|+1, f_g^{|w|+1}(K(w))) - g(|w|+1, s_w)| \\ &= \frac{\partial g}{\partial s}(|w|+1, s'_w)|f_g^{|w|+1}(K(w)) - s_w|, \end{aligned} \tag{2.4.6}$$

donde

$$\text{mín}\{f_g^{|w|+1}(K(w)), s_w\} \leq s'_w \leq \text{máx}\{f_g^{|w|+1}(K(w)), s_w\}.$$

Entonces, por la igualdad (2.4.4) y las ecuaciones (2.4.5) y (2.4.6),

$$|f_g^{|w|+1}(K(w)) - \dim_{d_g[\mathbf{m}]}(w)| \leq \frac{c}{\frac{\partial g}{\partial s}(|w|+1, s'_w)} \leq \frac{c}{\alpha(|w|+1)}, \tag{2.4.7}$$

donde, en la última desigualdad, α es la función que aparece en la definición de función escala regular.

Por último, el Corolario 2.2.3 nos dice que

$$|\dim_{d_g[\mathbf{m}]}(w) - \dim_g(w)| \leq \frac{c'}{\alpha(|w|+1)}. \tag{2.4.8}$$

Así que, de las desigualdades (2.4.7) y (2.4.8) tenemos que, tal como queríamos demostrar,

$$\begin{aligned} |f_g^{|w|+1}(K(w)) - \dim_g(w)| &\leq |f_g^{|w|+1}(K(w)) - \dim_{d_g[\mathbf{m}]}(w)| \\ &+ |\dim_{d_g[\mathbf{m}]}(w) - \dim_g(w)| \\ &\leq \frac{c+c'}{\alpha(|w|+1)}. \end{aligned}$$

□

Como corolario de los Teoremas 2.3.1 y 2.4.1 se tiene la relación entre la dimensión constructiva con escala y la complejidad de Kolmogorov de sus prefijos.

Corolario 2.4.2. Sea $A \in \mathbf{C}$ y sea g una función escala regular. Entonces, la dimensión constructiva de A es exactamente

$$\mathrm{cdim}_g(A) = \liminf_{n\to\infty} f_g^{n+1}(K(A[0..n-1])).$$

En particular, para cualquier función escala de la familia $\{g_k\}_{k\in\mathbb{Z}}$,

Corolario 2.4.3. Sea $k \in \mathbb{Z}$ y $A \in \mathbf{C}$, entonces

$$\mathrm{cdim}^{(k)}(A) = \liminf_{n\to\infty} f_k^{n+1}(K(A[0..n-1])).$$

En el caso particular de $i = 0$ se tiene el resultado para dimensión constructiva obtenido por Mayordomo en [77].

$$\mathrm{cdim}(A) = \liminf_{n\to\infty} \frac{K(A[0..n-1])}{n+1}.$$

2.5. Termgalas vs Supertermgalas

En [35] se demuestra que las galas constructivas y las supergalas constructivas son intercambiables a la hora de definir la dimensión constructiva en $\mathbf{C}$. Es bien conocido [39, 67] que también se pueden itercambiar supergalas y galas en las dimensiones con recursos acotados tanto en el caso clásico como en el caso escalado.

En esta sección se plantea la cuestión de si es posible intercambiar supertermgalas por termgalas en la definición de dimensión en $\{0,1\}^*$. El principal problema que ya se plantea en [68] es que, si bien

es posible demostrar la existencia de supertermgalas constructivas óptimas (tal como se ve con las g-supertermgalas), no se conoce la existencia de una termgala constructiva óptima. Esto impide una definición consistente de dimensión en $\{0,1\}^*$ basada en termgalas. Aún así, sí que resultará posible obtener una caracterización de dimensión constructiva en $\mathbf{C}$ utilizando únicamente g-termgalas.

Definición 2.5.1. Una *s-termgala g-escalada* (o s^g-termgala) es una función $d_g : \mathcal{T} \to [0,\infty)$ tal que,

i) $d_g(x) \leq 1$ para todo $x \in \mathcal{T}$ con $|x| \notin H$.

ii) Para todo $w \in \{0,1\}^*$ con $|w| \in H$,

$$d_g(w) = \frac{d_g(w0) + d_g(w1) + d_g(w\square)}{2^{\triangle g(|w|,s)}}.$$

Es decir, una s^g-termgala no es más que una s^g-supertermgala que satisface la condición (2.1.1) de la definición con igualdad para todo $w \in \{0,1\}^*$ con $|w| \in H$.

Además, el Lema 2.1.2 enunciado para s^g-supertermgalas se cumple también para s^g-termgalas y cada 0^g-termgala induce una familia entera de s^g-termgalas, dando sentido a la siguiente definición.

Definición 2.5.2. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala.

1. Una *g-termgala* es una familia

   $$d_g = \{d_g^{(s)} \mid s \in [0,\infty)\},$$

   de modo que cada $d_g^{(s)}$ es una s^g-termgala y

   $$2^{-g(|x|,s)} d_g^{(s)}(x) = 2^{-g(|x|,s')} d_g^{(s')}(x)$$

   para todo $s, s' \in [0,\infty)$ y $x \in \mathcal{T}$ con $|x| \in H$.

2. Se dice que d_g es *constructiva* si $d_g^{(0)}$ es constructiva.

3. Una g-termgala constructiva $\tilde{d_g}$ es *óptima* si para cada g-termgala constructiva d_g existe una constante $\alpha > 0$ tal que

   $$\tilde{d_g}^{(s)}(w\square) > \alpha\, d_g^{(s)}(w\square)$$

   para todo $s \in [0,\infty)$ y $w \in \{0,1\}^*$ con $|w| \in H$.

Es decir, una *g-termgala* es un caso particular de g-supertermgala. La cuestión de si existe o no una g-termgala constructiva óptima queda abierta.

El siguiente resultado estudia la propiedad $d(w\Box) > 1$ en la que se basa la definición de dimensión en $\{0,1\}^*$. Se demuestra en este teorema que es posible conservar esta propiedad intercambiando una s^g-supertermgala por una t^g-termgala con $t > s$ cualquiera.

Teorema 2.5.3. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala estrictamente regular, es decir, tal que para todo $s' > s \geq 0$, con $m_g = \min(H \cap \mathbb{N})$,

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')} < \infty.$$

Sean $t > s > 0$ reales calculables. Entonces, si $d_g^{(s)}$ una s^g-supertermgala constructiva, existe una t^g-termgala constructiva $\tilde{d}_g^{(t)}$ tal que, para $|w|$ suficientemente grande,

$$d_g^{(s)}(w\Box) > 1 \;\Leftrightarrow\; \tilde{d}_g^{(t)}(w\Box) > 1.$$

Demostración. Sin perdida de generalidad, podemos considerar $d_g^{(s)}(u) < 1$ para todo $u \in \{0,1\}^*$ con $|u| = m_g$. Definimos para cada $n \geq m_g$ las funciones

$$\tilde{d}_n^{(t)} : \mathcal{T} \to [0,\infty)$$

del siguiente modo:

$$\tilde{d}_n^{(t)}(x) = 2^{-g(n+1,t)+g(|x|,t)}\rho_n(x)$$

para cada $x \in \mathcal{T}$ con $|x| \geq m_g$, donde

$$\rho_n(x) = \#\{v \in \{0,1\}^n \mid x \sqsubseteq v\Box \text{ y } d_g^{(s)}(v\Box) > 1\}.$$

Notar que por el Corolario 2.1.7 (considerando $u = x[0 \ldots m_g - 1]$), para todo $n \geq m_g$ y todo $x \in \mathcal{T}$ se tiene que $\rho_n(x) < 2^{g(n+1,s)-g(m_g,s)}$. En efecto, como

$$\sum_{v\in\{0,1\}^{n-m_g}} d_g^{(s)}(x[0 \ldots m_g - 1]v\Box) \leq 2^{g(n+1,s)-g(m_g,s)},$$

se tiene que el número de v's que verifican $d_g^{(s)}(x[0 \ldots m_g - 1]v\Box) > 1$ debe ser menor que $2^{g(n+1,s)-g(m_g,s)}$ y por lo tanto, también el número de v's tales que $x \sqsubseteq v\Box$ y $d_g^{(s)}(v\Box) > 1$.

Veamos que $\tilde{d}_n^{(t)}$ es una t^g-termgala constructiva para cada n. En efecto, comprobemos primero que cumple la condición de t^g-termgala. Para ello distinguiremos tres casos,

i) Si $|w| > n$ entonces,

$$\rho_n(w0) = \rho_n(w1) = \rho_n(w\Box) = \rho_n(w) = 0,$$

por lo tanto,

$$[\tilde{d}_n^{(t)}(w0) + \tilde{d}_n^{(t)}(w1) + \tilde{d}_n^{(t)}(w\Box)]2^{-\Delta g(|w|,t)} = 0 = \tilde{d}_n^{(t)}(w).$$

ii) Si $|w| = n$, entonces

$$\begin{aligned} \rho_n(w0) = \rho_n(w1) &= 0, \\ \rho_n(w\Box) &= \rho_n(w), \end{aligned}$$

por lo tanto,

$$\begin{aligned} [\tilde{d}_n^{(t)}(w0) + \tilde{d}_n^{(t)}(w1) + \tilde{d}_n^{(t)}(w\Box)]2^{-\Delta g(|w|,t)} &= 2^{-g(n+1,t)+g(|w|,t)}[0 + 0 + \rho_n(w)] \\ &= \tilde{d}_n^{(t)}(w). \end{aligned}$$

iii) Si $m_g \leq |w| < n$, entonces

$$\begin{aligned} \rho_n(w\Box) &= 0, \\ \rho_n(w0) + \rho_n(w1) &= \rho_n(w), \end{aligned}$$

por lo tanto,

$$\begin{aligned} [\tilde{d}_n^{(t)}(w0) + \tilde{d}_n^{(t)}(w1) + \tilde{d}_n^{(t)}(w\Box)]2^{-\Delta g(|w|,t)} &= 2^{-g(n+1,t)+g(|w|,t)}[\rho_n(w0) + \rho_n(w1) + 0] \\ &= 2^{-g(n+1,t)+g(|w|,t)}\rho_n(w) \\ &= \tilde{d}_n^{(t)}(w). \end{aligned}$$

Así pues, $\tilde{d}_n^{(t)}$ es t^g-termgala. Por otro lado, al ser g calculable y $d_g^{(s)}$constructiva, $\tilde{d}_n^{(t)}$ también es constructiva.

Ahora, sea s' tal que $s < s' < t$. Definimos $\tilde{d}_g^{(t)} : \mathcal{T} \to [0, \infty)$ como

$$\tilde{d}_g^{(t)}(x) = \sum_{n=m_g}^{\infty} 2^{g(n+1,t)-g(n+1,s')}\tilde{d}_n^{(t)}(x),$$

para todo $x \in \mathcal{T}$ con $|x| \geq m_g$.

Por el Lema 2.1.9 (adaptado a t^g-termgalas), $\tilde{d}_g^{(t)}$ es una t^g-termgala constructiva. En efecto, $\{2^{g(n,t)-g(n,s')}\}_{n\in\mathbb{N}}$ son números reales positivos y $\tilde{d}_n^{(t)}$ son t^g-termgalas. Así que, sólo hay que demostrar que para todo $w \in \{0,1\}^{m_g}$,

$$\sum_{n=m_g}^{\infty} 2^{g(n+1,t)-g(n+1,s')}\tilde{d}_n^{(t)}(w) < \infty.$$

Veamos que esto es así, sea $w \in \{0,1\}^{m_g}$,

$$\begin{aligned}
\sum_{n=m_g}^{\infty} 2^{g(n+1,t)-g(n+1,s')}\tilde{d}_n^{(t)}(w) &= \sum_{n=m_g}^{\infty} 2^{g(n+1,t)-g(n+1,s')}2^{-g(n+1,t)+g(m_g,t)}\rho_n(w) \\
&< \sum_{n=m_g}^{\infty} 2^{g(n+1,t)-g(n+1,s')}2^{-g(n+1,t)+g(m_g,t)}2^{g(n+1,s)-g(m_g,s)} \\
&\leq \sum_{n=m_g}^{\infty} 2^{g(n+1,s)-g(n+1,s')} < \infty,
\end{aligned}$$

donde esta última desigualdad se cumple puesto que g es una función escala estrictamente regular.

Así pues, hemos definido una t^g-termgala constructiva $(\tilde{d}_g^{(t)})$ y únicamente falta ver que

$$d_g^{(s)}(w\Box) > 1 \;\Leftrightarrow\; \tilde{d}_g^{(t)}(w\Box) > 1.$$

Para ello, notar que

$$\begin{aligned}
\tilde{d}_g^{(t)}(w\Box) &= 2^{g(|w|+1,t)-g(|w|+1,s')}\tilde{d}_{|w|}^{(t)}(w\Box) \\
&= 2^{g(|w|+1,t)-g(|w|+1,s')}2^{g(|w|+1,t)-g(|w|+1,t)}\rho_{|w|}(w\Box).
\end{aligned}$$

Como $\rho_{|w|}(w\Box) = 1 \Leftrightarrow d_g^{(s)}(w\Box) > 1$, tenemos que $\tilde{d}_g^{(t)}(w\Box) > 1$ si $d_g^{(s)}(w\Box) > 1$ (para $|w|$ suficientemente grande).

Por otro lado, en el caso de que $d_g^{(s)}(w\Box) \leq 1$, entonces $\rho_{|w|}(w\Box) = 0$, y por lo tanto $\tilde{d}_g^{(t)}(w\Box) = 0$.

Así pues, tal como queríamos demostrar,

$$d_g^{(s)}(w\Box) > 1 \;\Leftrightarrow\; \tilde{d}_g^{(t)}(w\Box) > 1.$$

□

Si la definición de dimensión en $\{0,1\}^*$ no se basara en una estrategia fija, el teorema anterior sería suficiente para intercambiar supertermgalas por termgalas (tal como ocurre en dimensión en $\mathbf{C}$). Sin embargo, tal como está definida la dimensión en $\{0,1\}^*$, es necesario cambiar una g-supertermgala constructiva óptima $d_g = \{d_g^{(s)} \mid s \in [0,\infty)\}$ por una g-termgala $\tilde{d}_g = \{\tilde{d}_g^{(s)} \mid s \in [0,\infty)\}$ constructiva. El siguiente teorema demuestra una condición suficiente para realizar dicho cambio.

Teorema 2.5.4. Sea g una función escala estrictamente regular y sea una g-supertermgala $d_g = \{d_g^{(s)} \mid s \in [0,\infty)\}$ constructiva óptima. Suponer que existe una g-termgala constructiva $\tilde{d}_g = \{\tilde{d}_g^{(s)} \mid s \in [0,\infty)\}$ verificando la siguiente condición:

$$\begin{aligned} &\text{Si } s \in [0,\infty) \text{ y } w \in \{0,1\}^* \text{ suficientemente grande verifican:} \\ &\forall t > s, \quad d_g^{(s)}(w\square) > 1 \Rightarrow \tilde{d}_g^{(t)}(w\square) > 1, \end{aligned} \tag{2.5.1}$$

entonces:

Existiría una constante $C > 0$ y una función $\alpha : H \to \mathbb{R}$ verificando que $\alpha(m)$ tiende a infinito cuando $m \to \infty$, tales que

$$|\dim_g(w) - \dim_{\tilde{d}_g}(w)| \leq \frac{C}{\alpha(|w|+1)}.$$

Demostración. Al ser d_g una g-supertermgala constructiva óptima se tiene que, por el Corolario 2.2.3, existe una constante $C_1 > 0$ y una función $\alpha(m)$ que tiende a infinito cuando $m \to \infty$ tal que, $\forall w \in \{0,1\}^*$ con $|w|$ suficientemente grande,

$$|\dim_g(w) - \dim_{d_g}(w)| \leq \frac{C_1}{\alpha(|w|+1)}. \tag{2.5.2}$$

Por otro lado, por la definición de dimensión y por la condición (2.5.1) es claro que

$$\dim_{\tilde{d}_g}(w) \leq \dim_{d_g}(w). \tag{2.5.3}$$

Por último, como $\tilde{d}_g$ es g-termgala constructiva, en particular, g-supertermgala constructiva, se tiene que por la Proposición 2.2.2 existe una constante $C_2 > 0$ tal que

$$\dim_{d_g}(w) \leq \dim_{\tilde{d}_g}(w) + \frac{C_2}{\alpha(|w|+1)}. \tag{2.5.4}$$

Luego, si $C = \text{máx}\{C_1, C_2\}$, se tiene que:

i) Por la ecuación (2.5.4),

$$\dim_{d_g}(w) - \dim_{\tilde{d}_g}(w) \leq \frac{C}{\alpha(|w|+1)}.$$

ii) Por las ecuaciones (2.5.3) y (2.5.2),

$$\begin{aligned} \dim_{\tilde{d}_g}(w) - \dim_g(w) &\leq \dim_{d_g}(w) - \dim_g(w) \\ &\leq \frac{C}{\alpha(|w|+1)}. \end{aligned}$$

Y por lo tanto,

$$|\dim_g(w) - \dim_{\tilde{d}_g}(w)| \leq \frac{C}{\alpha(|w|+1)},$$

y se tiene el resultado. □

La existencia de una g-termgala constructiva que cumpla la condición (2.5.1) es quizá más débil que la cuestión planteada por Lutz sobre la existencia de una termgala constructiva óptima. Es claro que la condición de Lutz implica que existe una g-termgala constructiva como en el teorema anterior (en realidad, sería la propia g-termgala óptima). Sin embargo, la existencia de una g-termgala como en el Teorema 2.5.4 no parece implicar la existencia de una g-termgala constructiva óptima.

El siguiente resultado es una aproximación a encontrar una g-termgala que cumpla las condiciones del Teorema 2.5.4. Desgraciadamente, aunque la g-termgala definida cumple la condición (2.5.1), no parece ser constructiva.

Teorema 2.5.5. Sea g una función escala estrictamente regular y sea $d_g = \{d_g^{(s)} \mid s \in [0,\infty)\}$ una g-supertermgala constructiva. Entonces, existe una g-termgala $\tilde{d}_g = \{\tilde{d}_g^{(s)} \mid s \in [0,\infty)\}$ verificando que si $s \in [0,\infty)$ y $w \in \{0,1\}^*$ son tales que $d_g^{(s)}(w\square) > 1$, entonces $\tilde{d}_g^{(t)}(w\square) > 1$, para todo $t > s$, $|w|$ suficientemente grande.

Demostración. Sea $m_g = \min(H \cap \mathbb{N})$. Definimos para cada $w \in \{0,1\}^*$ con $|w| \geq m_g$ y $s \in [0,\infty)$ la función $\rho(w,s)$ recursivamente como sigue,

i) si $|w| = m_g$,

$$\rho(w,s) = 2^{\Delta g(|w|,s)} \frac{d_g^{(s)}(w)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(wb)},$$

ii) si $w = b_1 \dots b_n$ con $n > m_g$ y $a(w) = b_1 \dots b_{n-1}$ entonces

$$\rho(w,s) = 2^{\Delta g(|w|,s)} \frac{d_g^{(s)}(w)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(wb)} \rho(a(w),s).$$

Notar que la función $\rho(w,s)$ no depende de la elección de s. Esto se debe a que d_g es g-supertermgala y por lo tanto, $\forall t \in [0,\infty)$, podemos demostrar por inducción sobre $|w|$ que $\rho(w,s) = \rho(w,t)$:

i) Para $|w| = m_g$,

$$\begin{aligned}
\rho(w,s) &= 2^{\Delta g(|w|,s)} \frac{d_g^{(s)}(w)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(wb)} \\
&= \frac{2^{-g(|w|,s)} d_g^{(s)}(w)}{\sum_{b\in\{0,1,\square\}} 2^{-g(|w|+1,s)} d_g^{(s)}(wb)} \\
&= \frac{2^{-g(|w|,t)} d_g^{(t)}(w)}{\sum_{b\in\{0,1,\square\}} 2^{-g(|w|+1,t)} d_g^{(t)(wb)}} \\
&= 2^{\Delta g(|w|,t)} \frac{d_g^{(t)}(w)}{\sum_{b\in\{0,1,\square\}} d_g^{(t)}(wb)} \\
&= \rho(w,t).
\end{aligned}$$

ii) Cierto para $|w|$, cierto para $|w|+1$,

$$\begin{aligned}
\rho(w\tilde{b},s) &= 2^{\Delta g(|w|+1,s)} \frac{d_g^{(s)}(w\tilde{b})}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(w\tilde{b}b)} \rho(w,s) \\
&= \frac{2^{-g(|w|+1,s)} d_g^{(s)}(w\tilde{b})}{\sum_{b\in\{0,1,\square\}} 2^{-g(|w|+2,s)} d_g^{(s)}(w\tilde{b}b)} \rho(w,s) \\
&= \frac{2^{-g(|w|+1,t)} d_g^{(t)}(w\tilde{b})}{\sum_{b\in\{0,1,\square\}} 2^{-g(|w|+2,t)} d_g^{(t)}(w\tilde{b}b)} \rho(w,s) \\
&= 2^{\Delta g(|w|,t)} \frac{d_g^{(t)}(w\tilde{b})}{\sum_{b\in\{0,1,\square\}} d_g^{(t)}(w\tilde{b}b)} \rho(w,s) \\
&= 2^{\Delta g(|w|,t)} \frac{d_g^{(t)}(w\tilde{b})}{\sum_{b\in\{0,1,\square\}} d_g^{(t)}(w\tilde{b}b)} \rho(w,t) \\
&= \rho(w\tilde{b},t).
\end{aligned}$$

Así pues, si definimos

$$\tilde{d}_g^{(s)}(wb) := d_g^{(s)}(wb)\rho(w,s)$$

tenemos que $\tilde{d}_g$ es una g-termgala. En efecto,

i) para cada $s \in [0,\infty)$, $\tilde{d}_g^{(s)}$ es s^g-termgala:

$$\begin{aligned}
& [\tilde{d}_g^{(s)}(w0) + \tilde{d}_g^{(s)}(w1) + \tilde{d}_g^{(s)}(w\square)] 2^{-\Delta g(s,|w|)} \\
= \; & [d_g^{(s)}(w0) + d_g^{(s)}(w1) + d_g^{(s)}(w\square)] 2^{-\Delta g(s,|w|)} \rho(w,s) \\
= \; & [d_g^{(s)}(w0) + d_g^{(s)}(w1) + d_g^{(s)}(w\square)] 2^{-\Delta g(s,|w|)} 2^{\Delta g(|w|,s)} \frac{d_g^{(s)}(w)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(wb)} \rho(a(w),s) \\
= \; & d_g^{(s)}(w)\rho(a(w),s) = \tilde{d}_g^{(s)}(w),
\end{aligned}$$

ii) para todas s y t se tiene que $d_g^{(s)}(w)2^{-g(|w|,s)} = d_g^{(t)}(w)2^{-g(|w|,t)}$ puesto que d_g es g-supertermgala y $\rho(w,s) = \rho(w,t)$.

Veamos que $\tilde{d}_g$ es la g-termgala que buscamos. Sea $s \in [0,\infty)$ tal que $d_g^{(s)}(w\square) > 1$, entonces $\forall t > s$ tenemos que

$$\begin{aligned} \tilde{d}_g^{(t)}(w\square) &= 2^{g(|w|+1,t)-g(|w|+1,s)}\tilde{d}_g^{(s)}(w\square) \\ &= 2^{g(|w|+1,t)-g(|w|+1,s)}d_g^{(s)}(w\square)\rho(w,s) \\ &> 2^{g(|w|+1,t)-g(|w|+1,s)}\rho(w,s). \end{aligned}$$

Haciendo la recursión en la definición de ρ se tiene que

$$\begin{aligned} \rho(w,s) &= \prod_{u\sqsubseteq w,|u|\geq m_g} 2^{\Delta g(|u|,s)}\frac{d_g^{(s)}(u)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(ub)} \\ &= 2^{-g(m_g,s)+g(|w|+1,s)}\prod_{u\sqsubseteq w,|u|\geq m_g}\frac{d_g^{(s)}(u)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(ub)}. \end{aligned}$$

Además, como $d_g^{(s)}$ es s^g-supertermgala, se tiene que

$$\frac{d_g^{(s)}(u)}{\sum_{b\in\{0,1,\square\}} d_g^{(s)}(ub)} \geq 2^{-\Delta g(|u|,s)}$$

y por lo tanto,

$$\rho(w,s) \geq 2^{-g(m_g,s)+g(|w|+1,s)}\prod_{u\sqsubseteq w,|u|\geq m_g} 2^{-\Delta g(|u|,s)} = 1.$$

Así pues tenemos que,

$$\tilde{d}_g^{(t)}(w\square) > 2^{g(|w|+1,t)-g(|w|+1,s)}$$

y al ser $t > s$ cualquiera, $\tilde{d}_g^{(t)}(w\square) > 1$, para $|w|$ suficientemente grande. □

Pese a que los teoremas anteriores no son suficientemente fuertes como para demostrar que es posible intercambiar g-supertermgalas por g-termgalas en la definición de dimensión constructiva en $\{0,1\}^*$, sí que sirven para proporcionar una nueva caracterización de dimensión constructiva en $\mathbf{C}$, basándonos sólo en g-termgalas constructivas. Para ello se utilizará el siguiente corolario.

Corolario 2.5.6. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala estrictamente regular. Sea $w \in \{0,1\}^*$ (con $|w|$ suficientemente grande), entonces

$$\dim_g(w) \geq \text{ínf}\{\dim_d(w) \mid d \text{ es una } g\text{-termgala constructiva}\}$$

Demostración. Sea $s_0 = \dim_g(w)$. Es suficiente demostrar que para todo s' número real calculable mayor que s_0 existe una g-termgala constructiva $\tilde{d}$ tal que $\dim_{\tilde{d}}(w) \leq s'$. Por la definición de $\dim_d$, esto se cumple si existe una g-termgala contructiva $\tilde{d}$ tal que $\tilde{d}^{(s')}(w\Box) > 1$.

Sea s un número real calculable tal que $s_0 < s < s'$. Entonces, como $s_0 = \dim_g(w)$, tenemos que $d_{\Box}^{(s)}(w\Box) > 1$. Por el Teorema 2.5.3, existe una $(s')^g$-termgala constructiva tal que $\tilde{d}^{(s')}(w\Box) > 1$.

Sea $\tilde{d}$ la g-termgala constructiva construida usando esta $(s')^g$-termgala, entonces

$$\dim_{\tilde{d}}(w) \leq s',$$

tal como queríamos demostrar. □

Utilizando el corolario anterior, la Proposición 2.2.2 y el Teorema 2.3.1 se tiene el siguiente resultado.

Corolario 2.5.7. Sea $A \in \mathbf{C}$ y g una función escala estrictamente regular,

$$\mathrm{cdim}_g(A) = \liminf_{n\to\infty} \mathcal{D}_g(A[0\ldots n-1])$$

donde $\mathcal{D}_g(w) = \text{ínf}\{\dim_d(w) \mid d \text{ es una } g\text{-termgala constructiva}\}$.

Demostración. Por el Teorema 2.3.1 sabemos que

$$\mathrm{cdim}_g(A) = \liminf_{n\to\infty} \dim_g(A[0\ldots n-1]). \tag{2.5.5}$$

Por el corolario anterior tenemos que

$$\dim_g(A[0\ldots n-1]) \geq \mathcal{D}_g(A[0\ldots n-1]). \tag{2.5.6}$$

Por la Proposición 2.2.2, para toda g-termgala d tenemos que

$$\dim_g(A[0\ldots n-1]) \leq \dim_d(A[0,\ldots n-1]) + \frac{C}{\alpha(n+1)}, \tag{2.5.7}$$

donde α tiende a infinito cuando n tiende a infinito.

Así que, por (2.5.6),

$$\liminf_{n\to\infty} \dim_g(A[0\ldots n-1]) \geq \liminf_{n\to\infty} \mathcal{D}_g(A[0\ldots n-1]).$$

Por (2.5.7),

$$\liminf_{n\to\infty} \dim_g(A[0\ldots n-1]) \leq \liminf_{n\to\infty} \mathcal{D}_g(A[0\ldots n-1]).$$

Luego, por la regla del sandwich y por (2.5.5) tenemos el resultado. □

Este resultado establece que la dimensión constructiva de una secuencia infinita $A \in \mathbf{C}$ no es más que el límite del ínfimo del conjunto de dimensiones de los prefijos de A, construidas con g-termgalas constructivas. Esto quiere decir que realmente, cuando se trabaja en secuencias finitas suficientemente grandes, su dimensión se puede aproximar por el ínfimo de las dimensiones obtenidas a partir de g-termgalas constructivas.

2.6. Dimensión con escala en $\{0,1\}^*$ y predicción

Una de las líneas de investigación relacionadas con dimensión en $\mathbf{C}$ es aquella que caracteriza la dimensión con la predicción. Esta caracterización resulta ser muy natural debido a la interpretación que se hace de las s-galas como estrategias de juego: si se pueden predecir los bits de la secuencia, se podrán construir estrategias ganadoras. En el Capítulo 6 de esta tesis se tratará en mayor profundidad la relación entre dimensión y aprendizaje.

En esta sección se relaciona dimensión en $\{0,1\}^*$ con predicción "on-line". También resulta natural esta aproximación, puesto que como se ha visto, se pueden interpretar las s^g-termgalas como estrategias de juego. Sin embargo, una caracterización completa de dimesión con predicción no parece ser posible.

Considérese que se quieren predecir los bits de una secuencia finita. Entonces, dado un prefijo de esta secuencia, el siguiente bit puede ser un 0, un 1 o quizá, la secuencia no tenga más bits.

Un predictor Π dará una estimación de la probabilidad de cada uno de esos casos.

Definición 2.6.1. Una función $\Pi : \{0,1\}^* \times \{0,1,\square\} \to [0,1]$ es un *predictor* si verifica

$$\Pi(w,0) + \Pi(w,1) + \Pi(w,\square) = 1.$$

Se interpreta $\Pi(w,a)$ como la estimación que hace el predictor Π de que el bit a sea el siguiente después de aparecer w en el juego (en el caso de que $a = 0$ ó 1) o que no hay mas bits siguiendo w (en el caso de que $a = \square$).

Nota 2.6.2. En esta sección serán necesaria las siguientes restricciones sobre la función escala:

1. $g(m_g, s) \geq 0$, $\forall s$.

2. $g(m_g, s) \leq m_g$, $\forall s \leq 1$.

El siguiente lema establece la correspondencia entre predictores y g-termgalas.

Lema 2.6.3. Sea g una función escala regular y sea $m_g = \text{mín}(H \cap \mathbb{N})$.

1. Sea Π un predictor, se define para todo $s \in [0,\infty)$ la función

$$d^{(s)}_{\Pi,g} : \mathcal{T} \to [0,\infty),$$

como

$$d^{(s)}_{\Pi,g}(x) = \begin{cases} 1 & \text{si } |x| \notin H. \\ 2^{g(|x|,s)} \prod_{i=m_g+1}^{|x|} \Pi(x[0\ldots i-2], x[i-1]) & \text{si } |x| \in H. \end{cases}$$

Entonces, $d_{\Pi,g}$ es una g-termgala.

2. Sea d una g-termgala, entonces para todo $s \in [0,\infty)$ se define la función

$$\Pi_{d,g} : \{0,1\}^* \times \{0,1,\square\} \to [0,1]$$

como

$$\Pi_{d,g}(w,a) = \begin{cases} 2^{-\Delta g(|w|,s)} \frac{d^{(s)}(wa)}{d^{(s)}(w)} & \text{si } |w| \in H \text{ y } d^{(s)}(w) \neq 0. \\ 1/2 & \text{si } a \neq \square \text{ y } |w| \notin H \text{ ó } d^{(s)}(w) = 0\ . \\ 0 & \text{si } a = \square \text{ y } |w| \notin H \text{ ó } d^{(s)}(w) = 0\ . \end{cases}$$

Entonces, $\Pi_{d,g}$ es un predictor y esta definición no depende de s.

3. Los conceptos anteriores están relacionados del siguiente modo.

   i) Sea d una g-termgala tal que para todo $s \in [0,\infty)$, $x \in \mathcal{T}$ con $|x| \notin H$, $d^{(s)}(x) = 1$. Entonces, para todo $s \in [1,\infty)$, $x \in \mathcal{T}$ con $x \in H$ y $d^{(s)}(x) \neq 0$,

$$d^{(s)}_{\Pi_{d,g},g}(x) \geq d^{(s)}(x) 2^{m_g},$$

   y para $s \leq 1$,

$$d^{(s)}_{\Pi_{d,g},g}(x) \leq d^{(s)}(x) 2^{m_g}.$$

   ii) Sea Π un predictor tal que $\Pi(w,a) \neq 0$, $\forall w \in \{0,1\}^*$ con $|w| \geq m_g$, $a \in \{0,1,\square\}$. Para todo $w \in \{0,1\}^*$ con $|w| \in H$, $a \in \{0,1,\square\}$,

$$\Pi_{d_{\Pi,g},g}(w,a) = \Pi(w,a).$$

Demostración. 1. Sea Π un predictor y $s \in [0,\infty)$.

i) Veamos que $d^{(s)}_{\Pi,g}$ cumple la condición de s^g-termgala. Para todo $w \in \{0,1\}^*$ con $|w| \in H$,

$$\begin{aligned} & [d^{(s)}_{\Pi,g}(w0) + d^{(s)}_{\Pi,g}(w1) + d^{(s)}_{\Pi,g}(w\Box)]2^{-\Delta g(|w|,s)} \\ = \; & 2^{g(|w|+1,s)} \prod_{i=m_g}^{|w|} \Pi(w[0\ldots i-2], w[i-1]) \\ & [\Pi(w,0) + \Pi(w,1) + \Pi(w,\Box)]2^{-\Delta g(|w|,s)} \\ = \; & 2^{g(|w|,s)} \prod_{i=m_g}^{|w|} \Pi(w[0\ldots i-2], w[i-1]) \\ = \; & d^{(s)}_{\Pi,g}(w). \end{aligned}$$

ii) Veamos que para todo s', s, se cumple la condición de g-termgala. Para todo $x \in \mathcal{T}$ con $|x| \in H$,

$$\begin{aligned} d^{(s)}_{\Pi,g}(x)2^{-g(|x|,s)} &= 2^{g(|x|,s)} \prod_{i=m_g+1}^{|x|} \Pi(x[0\ldots i-2], x[i-1])2^{-g(|x|,s)} \\ &= \prod_{i=m_g+1}^{|x|} \Pi(x[0\ldots i-2], x[i-1]) \\ &= 2^{g(|x|,s')} \prod_{i=m_g+1}^{|x|} \Pi(x[0\ldots i-2], x[i-1])2^{-g(|x|,s')} \\ &= d^{(s')}_{\Pi,g}(x)2^{-g(|x|,s')}. \end{aligned}$$

Luego $d_{\Pi,g}$ es una g-termgala.

2. Sea d una g-termgala y $s \in [0,\infty)$. Veamos que $\Pi_{d,g}$ cumple la condición de predictor.

Si $|w| \notin H$ ó $d(w) = 0$ es claro que se cumple.

Si $|w| \in H$ y $d(w) \neq 0$, entonces

$$\begin{aligned} \Pi_d(w,0) + \Pi_d(w,1) + \Pi_d(w,\Box) &= 2^{-\Delta g(|w|,s)} \frac{[d^{(s)}_g(w0) + d^{(s)}_g(w1) + d^{(s)}_g(w\Box)]}{d^{(s)}_g(w)} \\ &= 1. \end{aligned}$$

3. *i)* Para todo $x \in \mathcal{T}$ con $|x| \in H$ y $d^{(s)}(x) \neq 0$,

$$\begin{aligned} d^{(s)}_{\Pi_{d,g}}(x) &= 2^{g(|x|,s)} \prod_{i=m_g+1}^{|x|} \Pi_{d,g}(x[0\ldots i-2], x[i-1]) \\ &= 2^{g(|x|,s)} \prod_{i=m_g+1}^{|x|} 2^{-\Delta g(i-1,s)} \frac{d^{(s)}(x[0\ldots i-1])}{d^{(s)}(x[0\ldots i-2])} \\ &= d^{(s)}(x) 2^{g(m_g,s)}. \end{aligned}$$

ii) Para todo $w \in \{0,1\}^*$ con $|w| \in H$, $a \in \{0,1,\square\}$, $\Pi(w[0\ldots i-1], w[i]) \neq 0$ para $i \geq m_g$,

$$\begin{aligned} \Pi_{d_{\Pi,g},g}(w,a) &= 2^{-\Delta g(|w|,s)} \frac{d^{(s)}_{\Pi,g}(wa)}{d^{(s)}_{\Pi,g}(w)} \\ &= 2^{-\Delta g(|w|,s)} \frac{2^{g(|w|+1,s)} \prod_{i=m_g+1}^{|w|} \Pi(w[0\ldots i-2], w[i-1]) \Pi(w,a)}{2^{g(|w|,s)} \prod_{i=m_g+1}^{|w|} \Pi(w[0\ldots i-2], w[i-1])} \\ &= \Pi(w,a). \end{aligned}$$

□

Para tener una medida de la efectividad que tiene un predictor, se considerará la siguiente función de pérdida.

Definición 2.6.4. Sea $x \in \mathcal{T}$ y Π un predictor. Definimos la *función de pérdida logaritmica (log-loss) de* Π como

$$\mathcal{L}^{\log}_{\Pi}(x) = \sum_{i=0}^{|x|-1} \log \frac{1}{\Pi(x[0\ldots i-1], x[i])}.$$

Notar que cuanto más efectivo es el predictor para x, menor será el valor de la función log-loss evaluada en x (puesto que los valores que tomarán $\Pi(x[0\ldots i-1], x[i])$ serán próximos a 1).

El siguiente resultado proporciona una caracterización de la dimensión relativa a una termgala constructiva en términos de predictores.

Teorema 2.6.5. Sea g una función escala regular. Sea d una g-termgala constructiva y $w \in \{0,1\}^*$ con $|w| \in H$ y $d^{(1)}(w\square) > 1$ suficientemente grande. Entonces, la dimensión relativa a d de w verifica

$$\dim_d(w) \leq f_g^{|w|+1}(\mathcal{L}^{\log}_{\Pi_{d,g}}(w\square)).$$

Demostración. Sea $s \leq 1$, por el punto 3 del Lema 2.6.3 (tomando logaritmos) tenemos que

$$\log d^{(s)}(w\square) = -g(m_g, s) + \log d^{(s)}_{\Pi_{d,g},g}(w\square). \tag{2.6.1}$$

Por la definición de $d_{\Pi_{d,g}}$ y tomando logaritmos, se tiene

$$\begin{aligned}
\log d^{(s)}_{\Pi_{d,g},g}(w\Box) &= g(|w|+1,s) \\
&+ \sum_{i=m_g}^{|w|-1} \log \Pi_{d,g}(w[0\ldots i-1], w[i]) + \log \Pi_{d,g}(w,\Box) \\
&= g(|w|+1,s) + \sum_{i=0}^{|w|-1} \log \Pi_{d,g}(w[0\ldots i-1], w[i]) \\
&- \sum_{i=0}^{m_g-1} \log \Pi_{d,g}(w[0\ldots i-1], w[i]) + \log \Pi_{d,g}(w,\Box) \\
&= g(|w|+1,s) - \mathcal{L}^{\log}_{\Pi_{d,g}}(w\Box) - \sum_{i=0}^{m_g-1} \log 1/2 \\
&= g(|w|+1,s) - \mathcal{L}^{\log}_{\Pi_{d,g}}(w\Box) + m_g
\end{aligned}$$

Luego, por (2.6.1)

$$\log d^{(s)}(w\Box) = -g(m_g,s) + g(|w|+1,s) - \mathcal{L}^{\log}_{\Pi_{d,g}}(w\Box) + m_g.$$

Ahora, si $d^{(1)}(w\Box) > 1$ tenemos,

$$\begin{aligned}
\dim_d(w) &= \text{ínf}\{s | d^{(s)}(w\Box) > 1\} \\
&= \text{ínf}\{s \mid -g(m_g,s) + g(|w|+1,s) - \mathcal{L}^{\log}_{\Pi_{d,g}}(w\Box) + m_g > 0\} \\
&\leq f^{|w|+1}(\mathcal{L}^{\log}_{\Pi_{d,g}}(w\Box)).
\end{aligned}$$

donde la última desigualdad se obtiene a partir de la restricción $g(m_g,s) \leq m_g$ para todo $s \leq 1$ de la Nota 2.6.2. □

En particular, para una termgala simple d sin escalar y $w \in \{0,1\}^*$, se tiene que la dimensión relativa a d es exactamente

$$\dim_d(w) = \frac{\mathcal{L}^{\log}_{\Pi_d}(w\Box)}{|w|+1}.$$

Desafortunadamente, y relacionado con la sección anterior, el que no se conozca la existencia de termgalas constructivas óptimas hace que no sea posible conseguir una igualdad de este tipo para la dimensión en $\{0,1\}^*$. Aún así es posible obtener el siguiente resultado para dimensión contructiva en $\mathbf{C}$ como consecuencia de la Proposición 2.2.2, el Teorema 2.6.5 y el Corolario 2.5.7. Se utilizarán los siguientes conceptos.

Definición 2.6.6. Sea g una función escala regular.

i) Sea Π un predictor. La *función de pérdida logarítmica de* Π *con respecto a* g se define para cada $A \in \mathbf{C}$ como

$$\mathcal{L}^{\log}_{\Pi,g}(A) = \liminf_n f_g^{n+1}(\mathcal{L}^{\log}_{\Pi}(A[0\ldots n-1])\square).$$

Notemos que, para todo $A \in \mathbf{C}$, $\mathcal{L}^{\log}_{\Pi,g}(A) \leq 1$ utilizando el predictor constante $1/2$.

ii) Sea $w \in \{0,1\}^*$, entonces

$$\mathcal{L}_g(w) = \text{ínf}\{f_g^{|w|+1}(\mathcal{L}^{\log}_{\Pi}(w\square)) \,|\, \Pi \text{ predictor constructivo}\}.$$

Notemos que para toda $w \in \{0,1\}^*$, $\mathcal{L}_g(w) \leq 1$ utilizando el predictor constante $1/2$.

Teorema 2.6.7. Sea g una función escala estríctamente regular y $A \in \mathbf{C}$ una secuencia infinita. Entonces,

$$i) \quad \text{cdim}^g(A) \leq \text{ínf}\{\mathcal{L}^{\log}_{\Pi,g}(A) \mid \Pi \text{ predictor constructivo }\},$$

$$ii) \quad \text{cdim}^g(A) \leq \liminf_n \mathcal{L}_g(A[0\ldots n-1]).$$

Demostración. *i)* Sea Π un predictor constructivo siempre positivo y sea $d_{\Pi,g}$ la g-termgala constructiva definida como en el Lema 2.6.3. Por la Proposición 2.2.2 y el Teorema 2.3.1 se tiene que

$$\begin{aligned} \text{cdim}_g(A) &= \liminf_n \dim_g(A[0\ldots n-1]) \leq \\ &\quad \liminf_n \dim_{d_{\Pi,g}}(A[0\ldots n-1]). \end{aligned}$$

Aplicando el Teorema 2.6.5 se tiene el resultado, ya que si $\mathcal{L}^{\log}_{\Pi,g}(A) < 1$, $d^{(1)}_{\Pi,g}(A[0\ldots n-1]\square) > 1$ para casi todo n.

ii) Aplicando el Corolario 2.5.7,

$$\text{cdim}_g(A) = \liminf_n \mathcal{D}_g(A[0,\ldots n-1])$$

donde

$$\mathcal{D}_g(A[0,\ldots n-1]) = \text{ínf}\{\dim_d(A[0\ldots n-1]) \,|\, d \;\; g\text{-termgala constructiva}\}.$$

Notemos que $\forall n$, $\mathcal{D}_g(A[0\ldots n-1]) \leq 1$. Por el Lema 2.6.3 se tiene que, si Π es un predictor constructivo siempre positivo, entonces existe una g-termgala constructiva $d_{\Pi,g}$ tal que $\Pi =$

$\Pi_{d_{\Pi,g},g}$. Por lo tanto, por el Teorema 2.6.5 y notando que si $f_g^{n+1}(\mathcal{L}_\Pi^{\log}(A[0\dots n-1]\square)) < 1$ entonces $d_{\Pi,g}^{(1)}(A[0\dots n-1]\square) > 1$,

$$\mathcal{D}_g(A[0\dots n-1]) \leq \mathcal{L}_g(A[0\dots n-1]),$$

de donde se obtiene el resultado.

□

Notar que en los dos apartados del teorema anterior no parece posible, en general, alcanzar la igualdad. En el primer caso la dificultad está en a la cuestión tratada en la sección anterior sobre si es posible alcanzar con termgalas la dimensión en $\{0,1\}^*$. En el segundo caso, la igualdad no se alcanza directamente puesto que, a partir de una termgala constructiva, el predictor que se obtiene (Lema 2.6.3) no tiene por que ser constructivo.

Capítulo 3

Caracterizaciones de dimensión efectiva con escala en C

Tal como se comentaba en el capítulo anterior, uno de los campos de interés de la dimensión es el estudiar como se relaciona con otros conceptos bien conocidos de la Teoría de la Información. En particular, es natural intentar establecer una relación con la complejidad de Kolmogorov, puesto que ambas son medidas de cantidad de información. Por ejemplo, Ryabko [80, 81], Staiger [87, 88], y Cai y Hartmanis [16] fueron los primeros en estudiar la relación entre dimensión clásica de Hausdorff y complejidad de Kolmogorov. Por un lado, Ryabko [81] proporcionó una cota superior para la dimensión de Hausdorff en términos de la complejidad de Kolmogorov. Más concretamente, demostró que, para todo conjunto de secuencias infinitas $X \subseteq \mathbf{C}$,

$$\dim_{\mathrm{H}}(X) \leq \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

Por otro lado, Staiger [87] demostró que la igualdad no era posible. Es decir, existen conjuntos $X \subseteq \mathbf{C}$ para los cuales

$$\dim_{\mathrm{H}}(X) < \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n}.$$

Con el desarrollo de la versión constructiva de la dimensión de Hausdorff, Mayordomo [77] y Lutz [68] demostraron que se conseguía una caracterización completa para la dimensión constructiva:

$$\mathrm{cdim}(X) = \sup_{A \in X} \liminf_{n \to \infty} \frac{K(A[0..n-1])}{n},$$

para todo conjunto $X \subseteq \mathbf{C}$. Además, este resultado se puede generalizar para el caso de la dimensión con escala constructiva, tal como se demuestra en el capítulo anterior de esta tesis. En cuanto a los casos de dimensión en espacio polinómico y dimensión calculable, Hitchcock [33] demostró que también es posible una caracterización completa para el caso no escalado. En este caso es necesario utilizar la versión de complejidad de Kolmogorov con recursos de espacio acotados. Más

concretamente se tiene que, para todo conjunto $X \subseteq \mathbf{C}$,

$$\dim_{\text{pspace}}(X) = \inf_{s \in \text{pspace}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^s(A[0..n-1])}{n}. \tag{3.0.1}$$

$$\dim_{\text{comp}}(X) = \inf_{s \in \text{comp}} \sup_{A \in X} \liminf_{n \to \infty} \frac{KS^s(A[0..n-1])}{n}. \tag{3.0.2}$$

Para demostrar estos resultados, las técnicas utilizadas por Hitchock pasan por caracterizar dimensión con otro concepto bien conocido de Teoría de la Información: la entropía. Esta, a su vez, se relaciona con la complejidad de Kolmogorov, obteniéndose así las igualdades (3.0.1) y (3.0.2).

La idea de Hitchcock se basa en el trabajo de Staiger [87, 88], donde se define un tipo de entropía que caracteriza la dimensión clásica de Hausdorff. A partir de la versión calculable y la versión en espacio polinómico de esta entropía, Hitchcok demostró [33] que era posible caracterizar las dimensiones calculable y en espacio polinómico respectivamente.

En este capítulo se extienden los resultados obtenidos por Hitchcock para el caso de dimensión con escala, caracterizándola mediante ambos conceptos: la complejidad de Kolmogorov y la entropía. En el caso de la dimensión en tiempo, parece que una caracterización de este estilo no es posible, estudiándose con más detalle dicho caso en el Capítulo 4.

Por último, se utiliza esta caracterización para estudiar el comportamiento de las reducciones P/poly-Turing en la clase ESPACE. Juedes y Lutz [47] probaron un *small span theorem* para reducciones P/poly-Turing en ESPACE. Este teorema dice que para cualquier $A \in$ ESPACE, tanto las clases de lenguajes que se reducen a A (el *lower span*) o las clases de problemas a los que A puede ser reducido (el *upper span*) tienen medida 0 en ESPACE. En esta tesis, se mejora este teorema reemplazando medida por dimensión con escala de orden -3. La demostración usa la caracterización de la dimensión con escala con complejidad de Kolmogorov demostrada en este capítulo. Este resultado también conlleva el *small span theorem* para dimension con escala para reducciones *many-one* polinómicas en ESPACE [36].

Este *small span theorem* implica que la clase de conjuntos P/poly-Turing duros para ESPACE tiene dimensión pspace con escala de orden -3 igual a 0. Este resultado implica que cada conjunto P/poly-Turing duro tiene complejidad de Kolmogorov acotada en espacio inusualmente baja. La cota superior que se obtiene coincide con la cota proporcionada por Juedes y Lutz [47] para conjuntos P/poly-*many-one* duros.

Los resultados de este capítulo han sido publicados junto con John Hitchcock y Elvira Mayordomo en [41] y posteriormente en una versión extendida de revista en [38].

3.1. Caracterización con Complejidad de Kolmogorov

En esta sección se extienden los resultados (3.0.1) y (3.0.2) obtenidos por Hitchcok en [33] al caso escalado. Las técnicas que se utilizan para obtener estos resultados son distintas, puesto que se demuestra directamente la relación entre dimensión con escala y complejidad de Kolmogorov, sin necesidad de utilizar la entropía (aunque esta caracterización también es posible y se verá en la siguiente sección).

En la caracterización se utilizará la siguiente notación.

Definición 3.1.1. Sea $A \in \mathbf{C}$ y sean $\mathrm{s}, \mathrm{t} : \mathbb{N} \to \mathbb{N}$ cotas de espacio y de tiempo. Sea $g : H \times [0, \infty) \to \mathbb{R}$ una función escala. Se definen:

$$
\begin{aligned}
&i) \quad \mathcal{K}_g(A) = \liminf_{n\to\infty} f_g^n(K(A[0..n-1])).\\
&ii) \quad \mathcal{K}_g^{\mathrm{t}}(A) = \liminf_{n\to\infty} f_g^n(K^{\mathrm{t}(n)}(A[0..n-1])).\\
&iii) \quad \mathcal{KS}_g^{\mathrm{s}}(A) = \liminf_{n\to\infty} f_g^n(KS^{\mathrm{s}(n)}(A[0..n-1])).
\end{aligned}
$$

donde f_g^n esta definida en el Capítulo 1 (Sección 1.5, Definición 1.5.5) como la inversa parcial de g.

Notación 3.1.2. En el caso que $g = g_k$, con $k \in \mathbb{Z}$, denotaremos los conceptos anteriores por $\mathcal{K}_{(k)}(A), \mathcal{K}_{(k)}^{\mathrm{t}}(A)$ y $\mathcal{KS}_{(k)}^{\mathrm{s}}(A)$ respectivamente.

Ejemplo 3.1.3. Sea una secuencia infinita $A \in \mathbf{C}$ y $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ cota de espacio. Entonces, utilizando la Proposición 1.5.20, se tiene que:

$$\mathcal{KS}_{(2)}^{\mathrm{s}}(A) = \liminf_n \frac{\log\log(KS^{\mathrm{s}}(A[0\ldots n-1]))}{\log\log n}.$$

$$\mathcal{KS}_{(1)}^{\mathrm{s}}(A) = \liminf_n \frac{\log(KS^{\mathrm{s}}(A[0\ldots n-1]))}{\log n}.$$

$$\mathcal{KS}_{(0)}^{\mathrm{s}}(A) = \liminf_n \frac{KS^{\mathrm{s}}(A[0\ldots n-1])}{n}.$$

$$KS_{(-1)}^{\mathrm{s}}(A) = \liminf_n \left(1 - \frac{\log(n+1-KS^{\mathrm{s}}(A[0\ldots n-1]))}{\log n}\right).$$

$$\mathcal{KS}^{\mathrm{s}}_{(-2)}(A) = \liminf_n \left(1 - \frac{\log\log(n+2-KS^{\mathrm{s}}(A[0\ldots n-1]))}{\log\log n}\right).$$

La siguiente observación establece el significado de los conceptos anteriores en términos de cotas i.o. (infinitamente a menudo).

Observación 3.1.4. Sea $k \in \mathbb{Z}$ y $A \in \mathbf{C}$. Sean $\mathrm{s}, \mathrm{t} : \mathbb{N} \to \mathbb{N}$ cotas de espacio y tiempo. Entonces,

$$\begin{aligned}
&i) \quad \mathcal{K}_g(A) = \text{ínf}\{s \in [0,\infty) | \exists^\infty n \; K(A[0..n-1]) < g(n,s)\}.\\
&ii) \quad \mathcal{K}^{\mathrm{t}}_g(A) = \text{ínf}\{s \in [0,\infty) | \exists^\infty n \; K^{\mathrm{t}(n)}(A[0..n-1]) < g(n,s)\}.\\
&iii) \quad \mathcal{KS}^{\mathrm{s}}_g(A) = \text{ínf}\{s \in [0,\infty) | \exists^\infty n \; KS^{\mathrm{s}(n)}(A[0..n-1]) < g(n,s)\}.
\end{aligned}$$

Cuando se consideran clases de lenguajes, se utilizará el peor caso, dando lugar a la siguiente definición.

Definición 3.1.5. Sea $X \subseteq \mathbf{C}$ y Δ una de las clases de funciones definidas en el Capítulo 1 (Subsección 1.3.1). Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala. Se definen:

$$\begin{aligned}
&i) \quad \mathcal{K}_g(X) = \sup_{A\in X} \mathcal{K}_g(A).\\
&ii) \quad \mathcal{K}^{\Delta}_g(X) = \inf_{\mathrm{t}\in\Delta} \sup_{A\in X} \mathcal{K}^{\mathrm{t}}_g(A).\\
&iii) \quad \mathcal{KS}^{\Delta}_g(X) = \inf_{\mathrm{s}\in\Delta} \sup_{A\in X} \mathcal{KS}^{\mathrm{s}}_g(A).
\end{aligned}$$

Notar que cuando $\Delta = \mathrm{comp}$, entonces $\mathcal{K}^{\mathrm{comp}}_g(X) = \mathcal{KS}^{\mathrm{comp}}_g(X)$.

Notación 3.1.6. En el caso que $g = g_k$, con $k \in \mathbb{Z}$, denotaremos los conceptos anteriores por $\mathcal{K}_{(k)}(X), \mathcal{K}^{\Delta}_{(k)}(X)$ y $\mathcal{KS}^{\Delta}_{(k)}(X)$ respectivamente.

El principal teorema de este capítulo se trata de una caracterización de la dimensión para los casos $\Delta \subseteq \mathrm{p}_j\mathrm{space}$ ó comp en términos de las funciones definidas anteriormente.

Teorema 3.1.7. Sea $X \subseteq \mathbf{C}$

1. Para todo $i, j \in \mathbb{N}$ con $i \leq j$,

$$\begin{aligned}
\dim^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) &= \mathcal{KS}^{\mathrm{p}_j\mathrm{space}}_{(i)}(X),\\
\dim^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X) &= \mathcal{KS}^{\mathrm{p}_j\mathrm{space}}_{(-i)}(X).
\end{aligned}$$

2. Para todo $k \in \mathbb{Z}$,

$$\dim^{(k)}_{\text{comp}}(X) = \mathcal{K}^{\text{comp}}_{(k)}(X).$$

Nota 3.1.8. Con esta notación, el Teorema 2.4.1 obtenido en el capítulo anterior es un resultado dual al enunciado en el Teorema 3.1.7 para el caso de dimensión constructiva:

Para todo $k \in \mathbb{Z}$,

$$\text{cdim}^{(k)}(X) = \mathcal{K}_{(k)}(X).$$

Ejemplo 3.1.9. Sean $i, j \in \mathbb{N}$ con $i \leq j$ y $X \in \mathbf{C}$,

$$\dim^{(i)}_{\text{p}_j\text{space}}(X) = \inf_{\text{s}\in\text{p}_j\text{space}} \sup_{A\in X} \liminf_{n} \frac{\log(\ldots^{i}\ldots(\log KS^{\text{s}}(A[0\ldots n-1])\ldots))}{\log(\ldots^{i}\ldots(\log n)\ldots)}.$$

Proposición 3.1.10. Una caracterización del tipo del Teorema 3.1.7 para los casos $\dim^{(-i)}_{\text{p}_j\text{space}}$, cuando $i > j$, no es posible.

Demostración. Por el Teorema 3,3 en [47] se tiene que, para cada $A \in$ ESPACE existe un $\epsilon > 0$ tal que

$$KS^{2^{2n}}(A^{\leq n}) < 2^{n+1} - 2^{\epsilon n} \qquad \text{a.e. } n.$$

Por lo tanto, dada la definición de $\mathcal{KS}$,

$$\mathcal{KS}^{\text{pspace}}_{(-2)}(\text{ESPACE}) = 0.$$

Por otro lado, es conocido (ver [67]) que $\dim^{(-2)}_{\text{pspace}}(\text{ESPACE}) = 1$. Así pues,

$$\dim^{(-2)}_{\text{pspace}}(\text{ESPACE}) \neq \mathcal{KS}^{\text{pspace}}_{(-2)}(\text{ESPACE})$$

y se tiene el resultado. □

Nota 3.1.11. Se puede probar una versión dual del Teorema 3.1.7 para la dimensión "*Packing*" (definida en la Subsección 1.4.4 del Capítulo 1.

3.1.1. Demostración del Teorema 3.1.7.

En esta sección se demuestra el Teorema 3.1.7 a partir de los Lemas 3.1.15 y 3.1.19. Ambos lemas proporcionan un resultado más general que el del Teorema 3.1.7 puesto que se enuncian no sólo para la familia de escalas $\{g_k\}_{k\in\mathbb{Z}}$, sino para funciones escala más generales.

El primer lema establece que la dimensión es menor que $\mathcal{K}$ o $\mathcal{KS}$ (dependiendo del caso). La demostración de este lema, se basa en la siguiente versión escalada del lema de Borel-Cantelli.

Lema 3.1.12. [39] Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala Δ-calculable con $m_g = \min(H \cap \mathbb{N})$ y sea $s \in [0,\infty)$. Sea $d : \mathbb{N}^2 \times \{0,1\}^* \to [0,\infty)$ una función Δ-calculable que verifica:

i) Para todo $j, n \in \mathbb{N}$, $d_{j,n}$ es una s^g-supergala.

ii) Para todo w con $|w| = m_g$ las series

$$\sum_{n=0}^{\infty} d_{j,n}(w) \ (j = 0, 1, 2 \ldots),$$

son uniformemente Δ-convergentes.

Entonces,

$$\dim_{\Delta}^{g}(\bigcup_{j=0}^{\infty}\bigcap_{t=0}^{\infty}\bigcup_{n=t}^{\infty} S^1[d_{j,n}]) \leq s.$$

Lema 3.1.13. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala Δ-calculable tal que para todo $s' > s$, la serie

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$$

es Δ-convergente (donde $m_g = \min(H \cap \mathbb{N})$). Entonces,

1. Para todo $j \in \mathbb{N}$, si $\Delta \subseteq \mathrm{p}_j\mathrm{space}$,

$$\dim_{\mathrm{p}_j\mathrm{space}}^{g}(X) \leq \mathcal{KS}_{g}^{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $\Delta \subseteq \mathrm{comp}$,

$$\dim_{\mathrm{comp}}^{g}(X) \leq \mathcal{K}_{g}^{\mathrm{comp}}(X).$$

Demostración. Para probar el caso 1, sean $t > s' > s > \mathcal{KS}_{g}^{\mathrm{p}_j\mathrm{space}}(X)$ números racionales. Sea $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ en $\mathrm{p}_j\mathrm{space}$ con $\mathrm{s}(n) \geq n$ tal que

$$\exists^{\infty} n \ KS^{\mathrm{s}(n)}(A[0..n-1]) < g(n,s),$$

para todo $A \in X$. Notar que dicha función $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ existe por la definición de $\mathcal{KS}_{g}^{\mathrm{p}_j\mathrm{space}}(X)$ y la Observación 3.1.4.

Definimos la siguiente función $d : \mathbb{N} \times \{0,1\}^* \to [0,\infty)$ mediante,

$$d_n(w) = \begin{cases} 2^{-g(n,s')+g(|w|,t)}\rho(w) & \text{si } m_g \leq |w| \leq n. \\ 2^{-g(n,s')+g(|w|,t)-|w|+n} d_n(w[0 \ldots n-1]) & \text{si } |w| > n. \end{cases}$$

donde

$$\rho(w) = \#\{\pi \in \{0,1\}^{<g(n,s)} \mid U(\pi) \text{ se calcula en espacio } \leq \mathrm{s}(n) \text{ y } \ w \sqsubseteq U(\pi)\}.$$

Notar que $\rho(w0) + \rho(w1) \leq \rho(w)$.

Veamos que para todo $n \geq m_g$, d_n es una t^g-supergala. Para ello debemos distinguir dos casos:

i) Si $|w| \geq n$, entonces

$$\begin{aligned} & [d_n(w0) + d_n(w1)]2^{-\Delta g(|w|,t)} \\ = \ & 2 \cdot 2^{-g(n,s')+g(|w|+1,t)-(|w|+1)+n} d_n(w[0 \ldots n-1]) 2^{-\Delta g(|w|,t)} \\ = \ & 2^{-g(n,s')+g(|w|,t)-|w|+n} d_n(w[0 \ldots n-1]) \\ = \ & d_n(w). \end{aligned}$$

ii) Si $m_g \leq |w| < n$, entonces

$$\begin{aligned} & [d_n(w0) + d_n(w1)]2^{-\Delta g(|w|,t)} \\ = \ & 2^{-g(n,s')+g(|w|+1,t)} [\rho(w0) + \rho(w1)] 2^{-\Delta g(|w|,t)} \\ \leq \ & 2^{-g(n,s')+g(|w|,t)} \rho(w) \\ = \ & d_n(w). \end{aligned}$$

Como $\Delta \subseteq \mathrm{p}_j\mathrm{space}$, entonces d es $\mathrm{p}_j$space-calculable.

Veamos ahora que las series $\sum_{n=m_g}^{\infty} d_n(w)$ son Δ-convergentes cuando $|w| = m_g$. De este modo, podremos aplicar Lema de Borel- Cantelli.

$$\begin{aligned} \sum_{n=m_g}^{\infty} d_n(w) &= \sum_{n=m_g}^{\infty} 2^{-g(n,s')+g(|w|,t)} \rho(w) \\ &\leq 2^{g(m_g,t)} \sum_{n=m_g}^{\infty} 2^{-g(n,s')+g(n,s)}. \end{aligned}$$

Por hipótesis, esta última serie es Δ-convergente. Así que, como $\Delta \subseteq \mathrm{p}_j\mathrm{space}$, también es $\mathrm{p}_j$space-convergente y por lo tanto, la serie original lo es.

Definimos ahora para cada $n \in \mathbb{N}$ con $n \geq m_g$ los conjuntos

$$Y_n = \{A \in \mathbf{C} \mid \mathrm{KS}^{\mathrm{s}(n)}(A[0 \ldots n-1]) < g(n,s)\}.$$

Entonces, para todo $n \in \mathbb{N}$ con $n \geq m_g$ tenemos que $Y_n \subseteq S^1[d_n]$. En efecto, si $A \in Y_n$ significa que $\rho(A[0 \ldots n-1]) \geq 1$, luego

$$d_n(A[0 \ldots n-1]) \geq 2^{-g(n,s')+g(|w|,t)} > 1.$$

Por otro lado,

$$X \subseteq \bigcap_{k=m_g}^{\infty} \bigcup_{n=k}^{\infty} Y_n.$$

Luego por el Lema 3.1.12, $\dim_{\Delta}^{g}(X) \leq s$. Como esto se cumple para cada $s > \mathcal{KS}_{g}^{\Delta}(X)$ se sigue que

$$\dim_{\Delta}^{g}(X) \leq \mathcal{KS}_{g}^{\Delta}(X).$$

La misma demostración sirve en el caso 2, puesto que no tenemos que preocuparnos por cotas en los recursos de cálculo. □

Nota 3.1.14. Esta misma demostración utilizando cotas de tiempo no es posible si se quiere demostrar que

$$\dim_{\mathrm{p}_j}^{g}(X) \leq \mathcal{K}_{g}^{\mathrm{p}_j}(X).$$

Esto es debido a que sería necesario asegurar que la función

$$\rho(w) = \#\{\pi \in \{0,1\}^{<g(n,s)} \mid U(\pi) \text{ se calcula en tiempo } \leq \mathrm{t}(n) \text{ y } \ w \sqsubseteq U(\pi)\}$$

fuera calculable en p_j, y para ello las funciones escala g deberían verificar que $2^{g(n,s)} \in \mathrm{p}_j$.

Así pues, la cuestión de cuando

$$\dim_{\mathrm{p}_j}^{g}(X) \leq \mathcal{K}_{g}^{\mathrm{p}_j}(X)$$

queda abierta, aunque en el Capítulo 4 se verá que parece necesario considerar compresión reversible para capturar la dimensión polinómica.

Como caso particular del Lema 3.1.13, se obtiene el siguiente resultado cuando se consideran las funciones escala $\{g_k\}_{k \in \mathbb{Z}}$.

Lema 3.1.15. Sea $X \subseteq \mathbf{C}$

1. Para todo $i, j \in \mathbb{N}$ con $i \leq j$

$$\dim_{\mathrm{p}_j\mathrm{space}}^{(i)}(X) \leq \mathcal{KS}_{(i)}^{\mathrm{p}_j\mathrm{space}}(X),$$

$$\dim_{\mathrm{p}_j\mathrm{space}}^{(-i)}(X) \leq \mathcal{KS}_{(-i)}^{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $k \in \mathbb{Z}$

$$\dim^{(k)}_{\text{comp}}(X) \leq \mathcal{K}^{\text{comp}}_{(k)}(X).$$

Lo único necesario para demostrar este lema es que la familia de escalas g_k verifica la condición requerida en el Lema 3.1.13, es decir:

Lema 3.1.16. Sea $k \in \mathbb{Z}$ y $m_k = \min(H \cap \mathbb{N})$. Sea $s' > s \geq 0$. Entonces la serie

$$\sum_{n=m_k}^{\infty} 2^{g_k(n,s)-g_k(n,s')}$$

es $\mathrm{p}_{|k|}$space-convergente.

Demostración. Tal como se ve en la demostración de la Proposición 1.5.14, existe un $\tilde{s} \in [s, s']$ y una constante $c > 0$ para la cual,

$$\sum_{n=h(r)}^{\infty} 2^{g_k(n,s)-g_k(n,s')} \leq \sum_{n=h(r)}^{\infty} 2^{cg_k(n,\tilde{s})},$$

luego basta ver que $\sum_{n=m_k}^{\infty} 2^{cg_k(n,\tilde{s})}$ es $\mathrm{p}_{|k|}$space-convergente.

Por el Lema 1.5.12 tenemos que existe M racional tal que

$$\sum_{n=m_k}^{\infty} 2^{c/2g_k(n,\tilde{s})} \leq M.$$

Definimos $h(r) = g_k(\frac{2}{c}\log(2^{-r}M), \frac{1}{s})$. Veamos que $h(r)$ es módulo de convergencia. Es claro que $h \in \mathrm{p}_{|k|}$space. Además,

$$\begin{aligned}
\sum_{n=h(r)}^{\infty} 2^{cg_k(n,\tilde{s})} &= \sum_{n=h(r)}^{\infty} [2^{c/2g_k(n,\tilde{s})} \cdot 2^{c/2g_k(n,\tilde{s})}] \\
&\leq 2^{c/2g_k(h(r),\tilde{s})} \cdot \sum_{n=h(r)}^{\infty} 2^{c/2g_k(n,\tilde{s})} \\
&\leq 2^{c/2g_k(h(r),\tilde{s})} \cdot M \\
&\leq 2^{-r},
\end{aligned}$$

donde la última desigualdad se obtiene al sustituir $h(r)$ por su valor y aplicar el punto 3 de la Proposición 1.5.11.

Así pues, la serie $\sum_{n=h(r)}^{\infty} 2^{cg_k(n,\tilde{s})}$ es $\mathrm{p}_{|k|}$space-convergente, con lo que se obtiene el resultado. □

Hasta el momento se ha demostrado que la dimensión es menor que $\mathcal{K}$ o $\mathcal{KS}$ (dependiendo del caso). El siguiente lema establece la desigualdad contraria y por tanto la caracterización del Teorema 3.1.7.

Lema 3.1.17. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala. Entonces,

$$\begin{aligned} \mathcal{KS}_g^{\mathrm{p}_j\mathrm{space}}(X) &\leq \dim^g_{\mathrm{p}_j\mathrm{space}}(X) \quad \forall j \in \mathbb{N}. \\ \mathcal{K}_g^{\mathrm{comp}}(X) &\leq \dim^g_{\mathrm{comp}}(X). \end{aligned}$$

Demostración. Veamos la demostración del primer caso.

Sea $s > \dim^g_{\mathrm{p}_j\mathrm{space}}(X)$ y sea d una s^g-gala $\mathrm{p}_j$space-calculable tal que $X \subseteq S^\infty[d]$. Fijemos $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ en $\mathrm{p}_j$space de forma que $\mathrm{s}(n) \geq n$ y tal que d se pueda calcular en espacio s. Podemos asumir, sin perdida de generalidad, que $d(w) < 1$ para todo $|w| \leq m_g$ (donde $m_g = \mathrm{mín}(H \cap \mathbb{N})$).

Por el Lema 1.4.21, para todo $n \geq m_g$,

$$\sum_{w \in \{0,1\}^n} d(w) \leq C 2^{g(n,s)}, \tag{3.1.1}$$

donde $C = 2^{-g(m_g,s)}$.

Sea el lenguaje $L = \{w \in \{0,1\}^* \mid d(w) > 1\}$. Entonces para todo $n \geq m_{|k|}$, por la desigualdad (3.1.1), se tiene que $|L^{=n}| < C 2^{g(n,s)}$.

Consideremos la lista de elementos de $L^{=n}$ ordenada en orden lexicográfico. Entonces cada $w \in L^{=n}$ puede describirse dando n y el índice dentro de esa lista. Reutilizando espacio, w puede calcularse a partir de esta descripción utilizando un espacio $3\mathrm{s}(n)$. Así pues, para todo $w \in L^{=n}$ con $n \geq m_g$,

$$KS^{3\mathrm{s}(n)}(w) \leq \log(|L^{=n}|) + O(\log n) < g(n,s) + O(\log n).$$

Consideremos ahora un lenguaje $A \in X$. Entonces, como $X \subseteq S^\infty[d]$ tenemos que existen infinitos n's tales que

$$A[0 \ldots n-1] \in L^{=n}$$

así que, existen infinitos n's tales que

$$KS^{3\mathrm{s}(n)}(A[0 \ldots n-1]) < g(n,s) + O(\log n).$$

Por lo tanto, $\mathcal{KS}_g^{3s}(A) \leq s$ y como esto es cierto para cualquier $A \in X$ y teníamos que s $\in$ p$_j$space, entonces $\mathcal{KS}_g^{\mathrm{p}_j\mathrm{space}}(X) \leq s$.

Como ésto se cumple para cualquier $s > \dim^g_{\mathrm{p}_j\mathrm{space}}(X)$, se tiene que $\mathcal{KS}_g^{\mathrm{p}_j\mathrm{space}}(X) \leq \dim^g_{\mathrm{p}_j\mathrm{space}}(X)$.

Esta misma demostración sirve también para el segundo caso, donde los recursos de cálculo no están acotados. □

Nota 3.1.18. Aunque esta demostración no puede adaptarse al caso de dimensión con escala en tiempo polinómico y complejidad de Kolmogorov acotada en tiempo (puesto que no podemos reutilizar tiempo), el resultado también es cierto en ese caso. Esto se demostrará en el Capítulo 4, donde se caracteriza la dimensión en tiempo polinómico utilizando compresión.

Como caso particular del Lema 3.1.17 tenemos el siguiente resultado cuando se considera la familia de escalas $\{g_k\}_{k\in\mathbb{Z}}$.

Lema 3.1.19. Sea $X \subseteq \mathbf{C}$. Para todo $j \in \mathbb{N}$, $k \in \mathbb{Z}$,

$$\mathcal{KS}_{(k)}^{\mathrm{p}_j\mathrm{space}}(X) \leq \dim^{(k)}_{\mathrm{p}_j\mathrm{space}}(X).$$

$$\mathcal{K}_{(k)}^{\mathrm{comp}}(X) \leq \dim^{(k)}_{\mathrm{comp}}(X).$$

Con esto queda demostrado el Teorema 3.1.7. Es más, a partir de los Lemas 3.1.13 y 3.1.17, se obtiene el siguiente resultado, algo más general que el Teorema 3.1.7.

Teorema 3.1.20. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala Δ-calculable tal que para todo $s' > s$, la serie

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$$

es Δ-convergente (donde $m_g = \mín(H \cap \mathbb{N})$). Entonces,

1. Para todo $j \in \mathbb{N}$, si $\Delta \subseteq$ p$_j$space,

$$\dim^g_{\mathrm{p}_j\mathrm{space}}(X) = \mathcal{KS}_g^{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $\Delta \subseteq$ comp,

$$\dim^g_{\mathrm{comp}}(X) = \mathcal{K}_g^{\mathrm{comp}}(X).$$

El siguiente resultado muestra que la caracterización que establece el Teorema 3.1.7 se cumple también cuando se restringe la complejidad de Kolmogorov a los prefijos de la forma $A^{\leq n}$, excepto en el caso de la escala cero.

Teorema 3.1.21. Sea $X \subseteq \mathbf{C}$,

1. Para todo $i, j \in \mathbb{N}$ con $0 < i \leq j$,
$$\dim^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) < s$$
sí y sólo sí existe una función $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ en $\mathrm{p}_j$space tal que para cada $A \in X$,
$$KS^{\mathrm{s}(2^{n+1})}(A^{\leq n}) < g_i(2^{n+1}, s) \qquad \text{i.o. } n.$$

2. Para todo $i, j \in \mathbb{N}$ con $0 < i \leq j$,
$$\dim^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X) < s$$
sí y sólo sí existe una función $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ en $\mathrm{p}_j$space tal que para cada $A \in X$,
$$KS^{\mathrm{s}(2^{n+1})}(A^{\leq n}) < g_{-i}(2^{n+1}, s) \qquad \text{i.o. } n.$$

3. Para todo $k \in \mathbb{Z}$, $k \neq 0$,
$$\dim^{(k)}_{\mathrm{comp}}(X) < s$$
sí y sólo sí existe una función $\mathrm{t} : \mathbb{N} \to \mathbb{N}$ calculable tal que para cada $A \in X$,
$$KT^{\mathrm{t}(2^{n+1})}(A^{\leq n}) < g_k(2^{n+1}, s) \qquad \text{i.o. } n.$$

Demostración. Veamos primero la implicación $\Rightarrow$, tendremos que distinguir los siguientes tres casos:
Caso 1. Sean s, s' números racionales tales que $\dim^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) < s < s'$. Por el Lema 3.1.19 y la Observación 3.1.4 podemos asegurar que existe una función $\mathrm{s} : \mathbb{N} \to \mathbb{N}$ en $\mathrm{p}_j$space verificando que para cada $A \in X$ e infinitos $m's$,
$$KS^{\mathrm{s}(m)}(A[0 \ldots m-1]) < g_i(m, s). \tag{3.1.2}$$

Fijemos $A \in X$ y $m \in \mathbb{N}$ de forma que la desigualdad (3.1.2) se verifique y además se cumpla $g_i(m, s) \leq g_i(\lfloor m/2 \rfloor, s')$.

Sea n el mayor número natural tal que $2^{n+1} - 1 \leq m$. Entonces $A[0.,2^{n+1} - 2]$ puede describirse a partir de $A[0 \ldots m-1]$ del siguiente modo:

Se utiliza una cinta auxiliar donde se irá escribiendo en unario $2^{n+1}-1$ para diferentes n's. Cuando alcancemos el mayor n que verifica $2^{n+1}-1 \leq m$, nos quedaremos con esos $2^{n+1}-1$ bits de $A[0\ldots m-1]$. Esta cinta auxiliar utilizará a lo más m celdas (luego, en términos de n, menos de 2^{n+2} celdas). Así pues,

$$\begin{aligned} KS^{\mathrm{s}(2^{n+2})+2^{n+2}}(A[0\ldots 2^{n+1}-2]) &< g_i(m,s) \\ &\leq g_i(2^{n+1},s') \end{aligned}$$

Si definimos la función $\mathrm{s}' : \mathbb{N} \to \mathbb{N}$ como $\mathrm{s}'(m) = \mathrm{s}(2m)+2m$, entonces $\mathrm{s}' \in \mathrm{p}_j\mathrm{space}$ y existen infinitos $n's$ para los cuales

$$KS^{\mathrm{s}'(2^{n+1})}(A^{\leq n}) < g_i(2^{n+1},s').$$

Caso 2. Para este caso, se puede repetir el argumento anterior, con la diferencia de que se debe tomar $m \in \mathbb{N}$ de modo que

$$KS^{\mathrm{s}(m)}(A[0\ldots m-1]) < g_{-i}(m,s) \leq m - g_i(m,1-s)$$

y verificando $g_i(m,1-s) \geq g_i(2m,1-s')$.

Se considera ahora el número natural n como el menor que verifica $2^{n+1}-1 \geq m$. De este modo, $A[0\ldots 2^{n+1}-2]$ puede describirse a partir de $A[0\ldots m-1]$ añadiendo los $2^{n+1}-m$ bits que faltan. En este caso,

$$\begin{aligned} KS^{\mathrm{s}(2^{n+1})+2^n}(A[0\ldots 2^{n+1}-2]) &< m - g_i(m,1-s) + (2^{n+1}-m) \\ &= 2^{n+1} - g_i(m,1-s) \\ &\leq 2^{n+1} - g_i(2^n,1-s) \\ &\leq 2^{n+1} - g_i(2^{n+1},1-s') = g_{-i}(2^{n+1},s') \end{aligned}$$

En este caso basta tomar $\mathrm{s}'(m) = \mathrm{s}(m) + m/2$.

Caso 3. La demostración es una combinación de los casos anteriores.

Para la implicación en el otro sentido $\Leftarrow$, para todos los casos, la demostración es una consecuencia de la Observación 3.1.4 y el Lema 3.1.15. □

Ejemplo 3.1.22. Dado $X \subseteq \mathbf{C}$, $\dim^{(-1)}_{\mathrm{pspace}}(X) < s$ si y solo si existe una constante $c > 0$ tal que para cada $A \in X$

$$KS^{2^{c(n+1)}}(A^{\leq n}) < 2^{n+1} - 2^{(n+1)(1-s)} \text{ i.o. } n.$$

En [47], Juedes y Lutz realizaron un estudio exahustivo de $KS(A^{=n})$. Utilizando el teorema anterior (Teorema 3.1.21) se hará, en la Sección 3.3, un estudio similar con $KS(A^{\leq n})$.

Notar que en cualquier caso que no es equivalente considerar $KS(A^{=n})$ y $KS(A^{\leq n})$.

En el primer caso, se requiere describir 2^n bits (describir la secuencia $A[2^n - 1 \dots 2^{n+1} - 2]$) mientras que, en el segundo caso, se describen $2^{n+1} - 1$ bits (la secuencia $A[0 \dots 2^{n+1} - 1]$).

Si $KS^{2^{cn}}(A^{=n}) < 2^n - 2^{\epsilon n}$ entonces $KS^{2^{c'n}}(A^{\leq n}) < 2^{n+1} - 2^{\epsilon(n+1)}$ lo cual implica que, en relación con el número de bits que hay que describir en uno y otro caso, $KS^{2^{c'n}}(A^{\leq n})$ no puede ser mayor que $KS^{2^{cn}}(A^{=n})$. Sin embargo, $KS^{2^{c'n}}(A^{\leq n})$ si puede ser mucho más pequeño que $KS(A^{=n})$.

3.2. Caracterización mediante Entropía

En esta sección se verán las relaciones entre dimensión y entropía. Esto se utilizará, junto con los resultados de la anterior sección, para demostrar la estrecha relación entre complejidad de Kolmogorov y entropía.

Definición 3.2.1. Sea $g : H \times [0, +\infty) \to \mathbb{R}$ una función escala.

1. La *entropía de un conjunto* $A \subseteq \{0,1\}^*$ *para la función escala* g se define como

$$H_A^g = \limsup_{n \to \infty} f_g^n(\log |A^{=n}|),$$

donde f_g^n es la inversa parcial de g definida en el Capítulo 1 (Sección 1.5, Definición 1.5.5).

2. La Δ-*entropía de* $X \subseteq \{0,1\}^\infty$ *para la función escala* g se define como

$$H_\Delta^g(X) = \text{ínf}\{\ H_A^g \mid X \subseteq A^{io} \text{ y } A \in R(\Delta)\ \},$$

donde A^{io} se define en los preliminares como

$$A^{io} = \{\ S \in \{0,1\}^\infty \mid \exists^\infty n \text{ con } S[0 \dots n] \in A\ \}.$$

Notación 3.2.2. Cuando $g = g_k$ $(k \in \mathbb{Z})$ se escribirá $H_A^{(k)}$ para referirse a $H_A^{g_k}$ y $H_\Delta^{(k)}(X)$ para referirse a $H_\Delta^{g_k}(X)$.

El siguiente resultado establece una definición equivalente de la entropía de un conjunto $A \subseteq \{0,1\}^*$ para la escala g, en el caso que la escala sea estrictamente regular.

Proposición 3.2.3. Sean un lenguaje $A \subseteq \{0,1\}^*$, una función escala $g : H \times [0,+\infty) \to \mathbb{R}$ estrictamente regular y $m_g = \text{mín}(H \cap \mathbb{N})$. Entonces,

$$H_A^g = \text{ínf } \Big\{ s \in [0,+\infty) \ \Big| \sum_{w \in A^{\geq m_g}} 2^{-g(|w|,s)} < +\infty \Big\}.$$

Demostración. Sea

$$r = \text{ínf } \Big\{ s \in [0,+\infty) \ \Big| \sum_{w \in A^{\geq m_g}} 2^{-g(|w|,s)} < +\infty \Big\}.$$

Observar que $r \leq 1$ puesto que $g(n,1) = n$.

Demostraremos primero la desigualdad $H_A^g \leq r$. Para ello sea $s > r$, entonces

$$\sum_{n=m_g}^{\infty} |A^{=n}| 2^{-g(n,s)} = \sum_{w \in A^{\geq m_g}} 2^{-g(|w|,s)} < +\infty.$$

Ahora bien, que esta serie sea convergente implica que debe existir algún $n_0 \in \mathbb{N}$ tal que $\forall n \geq n_0$,

$$2^{-g(n,s)} |A^{=n}| < 1.$$

Por lo tanto, utilizando la función inversa parcial de g, tenemos que $\forall n \geq n_0$,

$$f_g^n(\log(|A^{=n}|)) \leq s$$

y por lo tanto $H_A^g \leq s$. Al ser $s > r$ arbitrario, tenemos que $H_A^g \leq r$.

Veamos ahora la desigualdad $r \leq H_A^g$. Para ello sea $H_A^g < s < s'$. Por la definición de H_A^g, para algún $n_0 \in \mathbb{N}$ tenemos que $\forall n \geq n_0$,

$$f_g^n(\log(|A^{=n}|)) < s,$$

es decir, para todo $n \geq n_0$ se tiene que $|A_{=n}| < 2^{g(n,s)}$. Utilizando esta desigualdad y que g es función escala estrictamente regular, tenemos que la siguiente serie converge.

$$\sum_{w \in A^{\geq m_g}} 2^{-g(|w|,s')} = \sum_{n=m_g}^{\infty} |A^{=n}| 2^{-g(n,s')}$$
$$< \sum_{n=m_g}^{n_0} |A^{=n}| 2^{-g(n,s')} + \sum_{n=n_0}^{\infty} 2^{g(n,s)-g(n,s')} \leq \infty.$$

Por lo tanto tenemos que $r \leq s'$. Como hemos elegido $s' > H_A^g$ arbitrario, se tiene que $r \leq H_A^g$. □

Por la Proposición 1.5.14, las funciones g_k con $k \in \mathbb{Z}$ son estrictamente regulares, por lo tanto se tiene el siguiente corolario.

Corolario 3.2.4. Para todo $k \in \mathbb{Z}$,

$$H_A^{(k)} = \text{ínf}\left\{ s \in [0,+\infty) \;\middle|\; \sum_{w \in A} 2^{-g_k(|w|,s)} < +\infty \right\}.$$

Los siguientes resultados prueban que la dimensión es menor o igual que la entropía.

Lema 3.2.5. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala Δ-calculable tal que para todo $s' > s$, la serie

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$$

es Δ-convergente (donde $m_g = \text{mín}(H \cap \mathbb{N})$). Entonces,

1. Para todo $j \in \mathbb{N}$, si $\Delta \subseteq \mathrm{p}_j\mathrm{space}$,

$$\dim^g_{\mathrm{p}_j\mathrm{space}}(X) \leq H^g_{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $\Delta \subseteq \mathrm{comp}$,

$$\dim^g_{\mathrm{comp}}(X) \leq H^g_{\mathrm{comp}}(X).$$

Demostración. Para los dos apartados utilizaremos, basándonos en las ideas de [33], la siguiente técnica:

Sea $\Delta' \in \{\mathrm{p}_j\mathrm{space}, \mathrm{comp}\}$. Sea $r \leq 1$ tal que $H^g_{\Delta'} < r$. Sea $A \in R(\Delta')$ tal que $X \subseteq A^{io}$ y $2^{s'}, 2^s, 2^t$ racionales tales que $H^g_A < s < s' < t < r$. Notar que entonces, por la Proposición 3.2.3, tenemos que $|A_{=n}| \leq 2^{g(n,s)}$.

Para cada $n \geq m_g$, definimos $d_n : \{0,1\}^{\geq m_g} \to [0,+\infty)$ como

$$d_n(w) = \begin{cases} 2^{-g(n,t)+g(|w|,t)}\rho(w) & \text{si } |w| \leq n, \\ \\ 2^{-g(n,t)+g(|w|,t)-|w|+n} d_n(w[0 \ldots n-1]) & \text{si } |w| > n, \end{cases}$$

donde $\rho(w) = \#\{v \in \{0,1\}^n \mid v \in A \text{ y } w \sqsubseteq v\}$. Notar que $\rho(w0) + \rho(w1) \leq \rho(w)$.

Veamos que d_n es una t^g-gala para todo $n \geq m_g$. Para ello distinguiremos dos casos:

i) Si $|w| < n$ entonces

$$\begin{aligned} [d_n(w0) + d_n(w1)]2^{-\Delta g(|w|,t)} &= [\rho(w0) + \rho(w1)]2^{-\Delta g(|w|,t)} \\ &= 2^{-g(n,t)+g(|w|,t)}\rho(w) \\ &= d_n(w). \end{aligned}$$

ii) Si $w \geq n$ entonces

$$\begin{aligned}[d_n(w0)+d_n(w1)]2^{-\Delta g(|w|,t)} &= 2\cdot 2^{-g(n,t)+g(|w|+1,t)-|w|-1+n}d_n(w[0\dots n-1])2^{-\Delta g(|w|,t)}\\ &= 2^{-g(n,t)+g(|w|,t)}d_n(w[0\dots n-1])\\ &= d_n(w).\end{aligned}$$

Se define ahora $d:\{0,1\}^{\geq m_g}\to[0,+\infty)$ mediante

$$d=\sum_{n=m_g}^{\infty}2^{g(n,t)-g(n,s')}d_n.$$

Veamos que d es una t^g gala, es decir, por el Lema 1.4.20 tenemos que ver que, para todo w con $|w|=m_g$, se tiene que $d(w)<\infty$.

$$\begin{aligned}d(w) &= \sum_{n=m_g}^{\infty}2^{g(n,t)-g(n,s')}d_n(w)\\ &= \sum_{n=m_g}^{\infty}2^{g(n,t)-g(n,s')}\cdot 2^{-g(n,t)+g(m_g,t)}|A^{=n}|\\ &= 2^{g(m_g,t)}\sum_{n=m_g}^{\infty}2^{-g(n,s')}|A^{=n}|\\ &= 2^{g(m_g,t)}\sum_{w\in A^{>m_g}}2^{-g(|w|,s')}<\infty\end{aligned}$$

puesto que $s'>H_A^g$.

Luego d es t^g-gala y además, para cualquier $w\in A^{\geq m_g}$, se tiene que

$$d(w)\geq 2^{-g(|w|,s')+g(|w|,t)}d_{|w|}(w)=2^{-g(|w|,s')+g(|w|,t)}$$

y por tanto, $A^{io}\subseteq S^{\infty}[d]$.

Luego, lo único que tenemos que ver es que d cumple las restricciones de calculabilidad requeridas en cada apartado.

1. Veamos el caso en el que $\forall j\in\mathbb{N}$, $\Delta\subseteq \mathrm{p}_j\mathrm{space}$. Tenemos que demostrar que $d\in \mathrm{p}_j\mathrm{space}$. Notar que, al estar A en $R(\mathrm{p}_j\mathrm{space})$, entonces $d_n\in \mathrm{p}_j\mathrm{space}$ y por lo tanto existe una función $\hat{d_n}:\mathbb{N}\times\{0,1\}^*\to\mathbb{Q}$ en $\mathrm{p}_j\mathrm{space}$ tal que

$$|\hat{d_n}(r,w)-d_n(w)|\leq 2^{-r}.$$

Por otro lado, al ser $\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$ Δ-convergente ($\Delta \subseteq$ p$_j$space), tenemos que existe una función $h : \mathbb{N} \to \mathbb{N}$ en Δ tal que

$$\sum_{n=h(r)}^{\infty} 2^{g(n,s)-g(n,s')} \leq 2^{-r}.$$

Sea ahora

$$M(r) := \sum_{n=m_g}^{h(r)} 2^{g(n,t)-g(n,s')}$$

y sea $q : \mathbb{R} \to \mathbb{R}$ una función tal que $2^{-q(r)}M(r) \leq 2^{-r-1}$. Sea $r'(r,w) = g(|w|,t) + r + 1$. Definimos entonces $\hat{d} : \mathbb{N} \times \{0,1\}^* \to \mathbb{Q}$ como

$$\hat{d}(r,w) = \sum_{n=m_g}^{h(r')} 2^{g(n,t)-g(n,s')} \hat{d}_n(q(r'),w).$$

Claramente $\hat{d}$ está en p$_j$space, puesto que g y $\hat{d}_n$ son p$_j$space-calculables y podemos calcular las funciones $M(r)$, $q(r)$ y $r'(r,w)$ en espacio adecuado.

$$\begin{aligned}
|\hat{d}(r,w) - d(w)| &\leq \sum_{n=m_g}^{h(r')} 2^{g(n,t)-g(n,s')} |\hat{d}_n(q(r'),w) - d_n(w)| \\
&+ \sum_{n=h(r')}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w) \\
&\leq M(r')2^{-q(r')} + \sum_{n=h(r')}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w) \\
&\leq 2^{-r'-1} + \sum_{n=h(r')}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w).
\end{aligned}$$

Ahora bien, para acotar el último sumando, distinguiremos dos casos:

i) Si $|w| \leq h(r')$, entonces tenemos que

$$\begin{aligned}
\sum_{n=h(r')}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w) &= 2^{g(|w|,t)} \sum_{n=h(r')}^{\infty} 2^{-g(n,s')} \rho(w) \\
&\leq 2^{g(|w|,t)} \sum_{n=h(r')}^{\infty} 2^{g(n,s)-g(n,s')} \leq 2^{-r-1}
\end{aligned}$$

ii) Si $|w| > h(r')$, entonces tenemos que

$$\begin{aligned}\sum_{n=h(r')}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w) &= \sum_{n=h(r')}^{|w|} 2^{g(n,t)-g(n,s')} d_n(w) \\ &+ \sum_{n=|w|+1}^{\infty} 2^{g(n,t)-g(n,s')} d_n(w) \\ &\leq \sum_{n=h(r')}^{|w|} 2^{-g(n,s')} 2^{g(|w|,t)-|w|+n} d_n(w[0\ldots n-1]) \\ &+ \sum_{n=|w|+1}^{\infty} 2^{-g(n,s')} 2^{g(|w|,t)} \rho(w) \\ &\leq 2^{g(|w|,t)} \sum_{n=h(r')}^{\infty} 2^{g(n,s)-g(n,s')} \leq 2^{-r-1}\end{aligned}$$

Luego en ambos casos,

$$|\hat{d}(r,w) - d(w)| \leq 2^{-r'-1} + 2^{-r-1} \leq 2^{-r},$$

y $d \in \mathrm{p}_j\mathrm{space}$.

2. En este caso $A \in DEC$ y la demostración es análoga, sin necesidad de preocuparnos por los recursos de cálculo.

□

Como caso particular, si consideramos la familia de escalas g_k, se tiene el siguiente resultado.

Corolario 3.2.6. Sea $X \subseteq \mathbf{C}$. Entonces,

1. Para todo $i, j \in \mathbb{N}$ con $i \leq j$

$$\dim^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) \leq H^{(i)}_{\mathrm{p}_j\mathrm{space}}(X),$$

$$\dim^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X) \leq H^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $k \in \mathbb{Z}$

$$\dim^{(k)}_{\mathrm{comp}}(X) \leq H^{(k)}_{\mathrm{comp}}(X).$$

Demostración. Por el Lema 3.1.16 tenemos que la familia de funciones escala $\{g_k\}_{k\in\mathbb{Z}}$ verifica las hipotesis del Lema 3.2.5. □

Así pues, hasta el momento se ha demostrado que dimensión es menor o igual que entropía. El siguiente lema demuestra que la otra desigualdad es cierta también con menos restricciones e incluso admite las cotas de tiempo.

Lema 3.2.7. Sea $\Delta \in \{\mathrm{p}_j\mathrm{space}, \mathrm{p}_j, \mathrm{comp}\}$. Sean $X \subseteq \mathbf{C}$ y $g : H \times [0,\infty) \to \mathbb{R}$ una función escala estrictamente regular. Entonces,

$$H^g_\Delta(X) \leq \dim^g_\Delta(X).$$

Demostración. Sea $\dim^g_\Delta(X) < s$ y sea d una s^g-gala exactamente Δ-calculable verificando $X \subseteq S^\infty[d]$. Podemos asumir sin perdida de generalidad que $d(w) < 1$ para todo w con $|w| = m_g$, donde $m_g = \mín\{H \cap \mathbb{N}\}$. Sea $n \geq m_g$, entonces

$$\sum_{w \in \{0,1\}^n} d(w) \leq C 2^{g(n,s)},$$

donde $C = 2^{g(m_g,s)}$. Sea $A = \{w \mid d(w) > 1\}$, entonces $|A^{=n}| \leq C 2^{g(n,s)}$ y $X \subseteq S^\infty[d] \subseteq A^{io}$. Para cualquier $s' > s$,

$$\begin{aligned}\sum_{w \in A^{\geq m_g}} 2^{-g(|w|,s')} &= \sum_{n=m_g}^{\infty} 2^{-g(n,s')} |A^{=n}| \\ &\leq \sum_{n=m_g}^{\infty} 2^{-g(n,s')+g(n,s)} C < +\infty,\end{aligned}$$

luego $H^g_A \leq s'$. Así pues, $H^g_A \leq s$ y como $A \in R(\Delta)$, se tiene que $H^g_\Delta(X) \leq s$. Al ser s arbitrariamente cercano a $\dim^g_\Delta(X)$ se sigue que $H^g_\Delta(X) \leq \dim^g_\Delta$. □

Como caso particular, si consideramos la familia de escalas g_k, se tiene el siguiente corolario.

Corolario 3.2.8. Sea $\Delta \in \{\mathrm{p}_j\mathrm{space}, \mathrm{p}_j, \mathrm{comp}\}$ y sea $X \subseteq \mathbf{C}$. Entonces, para todo $k \in \mathbb{Z}$,

$$H^{(k)}_\Delta(X) \leq \dim^{(k)}_\Delta(X).$$

Como combinación de los Lemas 3.2.5 y 3.2.7 se tiene la siguiente caracterización de dimensión en términos de entropía.

Teorema 3.2.9. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0,\infty) \to \mathbb{R}$ una función escala Δ-calculable tal que para todo $s' > s$, la serie

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$$

es Δ-convergente (donde $m_g = \mín(H \cap \mathbb{N})$). Entonces,

1. Para todo $j \in \mathbb{N}$, si $\Delta \subseteq \mathrm{p}_j\mathrm{space}$,

$$\dim^g_{\mathrm{p}_j\mathrm{space}}(X) = H^g_{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $\Delta \subseteq \mathrm{comp}$,

$$\dim^g_{\mathrm{comp}}(X) = H^g_{\mathrm{comp}}(X).$$

Para el caso particular de la familia de escalas $\{g_k\}_{k\in\mathbb{Z}}$ se tiene el siguiente corolario.

Corolario 3.2.10. Sea $X \subseteq \mathbf{C}$

1. Para todo $i, j \in \mathbb{N}$ con $i \leq j$

$$\dim^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) = H^{(i)}_{\mathrm{p}_j\mathrm{space}}(X),$$

$$\dim^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X) = H^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $k \in \mathbb{Z}$

$$\dim^{(k)}_{\mathrm{comp}}(X) = H^{(k)}_{\mathrm{comp}}(X).$$

3.2.1. Entropía y Complejidad de Kolmogorov

Tal como se ha visto, se puede caracterizar dimensión mediante complejidad de Kolmogorov y mediante entropía. A partir de estas caracterizaciones se puede establecer entonces la relación entre complejidad de Kolmogorov y entropía.

Teorema 3.2.11. Sea $X \subseteq \mathbf{C}$. Sea $g : H \times [0, \infty) \to \mathbb{R}$ una función escala Δ-calculable tal que para todo $s' > s$, la serie

$$\sum_{n=m_g}^{\infty} 2^{g(n,s)-g(n,s')}$$

es Δ-convergente (donde $m_g = \mín(H \cap \mathbb{N})$). Entonces,

i) Para todo $j \in \mathbb{N}$, si $\Delta \subseteq \mathrm{p}_j\mathrm{space}$,

$$H^g_{\mathrm{p}_j\mathrm{space}}(X) = \mathcal{KS}^g_{\mathrm{p}_j\mathrm{space}}(X).$$

ii) Para todo $\Delta \subseteq \mathrm{comp}$,

$$H^g_{comp}(X) = \mathcal{K}^g_{comp}(X).$$

Demostración. Basta aplicar las caracterizaciones de dimensión con entropía (Teorema 3.2.9) y con complejidad de Kolmogorov (Teorema 3.1.20). □

Como caso particular, cuando se considera la familia de funciones escala $\{g_k\}_{k\in\mathbb{Z}}$, se tiene el siguiente resultado.

Corolario 3.2.12. Sea $X \subseteq \mathbf{C}$. Entonces,

1. Para todo $i, j \in \mathbb{N}$ con $i \leq j$

$$H^{(i)}_{\mathrm{p}_j\mathrm{space}}(X) = \mathcal{KS}^{(i)}_{\mathrm{p}_j\mathrm{space}}(X),$$

$$H^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X) = \mathcal{KS}^{(-i)}_{\mathrm{p}_j\mathrm{space}}(X).$$

2. Para todo $k \in \mathbb{Z}$

$$H^{(k)}_{\mathrm{comp}}(X) = \mathcal{K}^{(k)}_{comp}(X).$$

3.3. Aplicaciones de la Caracterización

El estudio de las reducciones utilizando la teoría de la medida con recursos acotados ha dado lugar a numerosos resultados relacionados con la noción de completitud (ver por ejemplo [45, 46, 69, 64, 74]). Una de las herramientas que ha facilitado esta línea de investigación ha sido el desarrollo de los *Small Span Theorems* [56, 5, 15, 47, 63].

Brevemente, dado un tipo de reducción $\leq_{\mathcal{R}}$ y un lenguaje $A \subseteq \{0,1\}^*$, el *lower-span* de A es el conjunto $\mathcal{R}(A)$ consistente en todos los lenguajes que son $\leq_{\mathcal{R}}$-reducibles a A y el *upper-span* de A es el conjunto $\mathcal{R}^{-1}(A)$ de todos los lenguajes a los cuales A se $\leq_{\mathcal{R}}$-reduce. Si $\mathcal{C}$ es una clase de complejidad, como las vistas en los preliminares, que tiene una estructura de medida (en el sentido de [62]), entonces un resultado de *Small Span Theorem para $\leq_{\mathcal{R}}$-reducciones en $\mathcal{C}$* nos asegura que, para todo $A \in \mathcal{C}$, al menos uno de los *spans* ($\mathcal{R}(A)$ ó $\mathcal{R}^{-1}(A)$) es pequeño en $\mathcal{C}$ (en el sentido de que tiene medida 0).

El primer *Small Span Theorem* fue demostrado por Juedes y Lutz en [46] para reducciones *many-one* ($\leq^p_m$) y para las clases E y E_2. Posteriormente fueron demostrándose *Small Span Theorems* con otro tipo de reducciones y clases , obteniéndose implicaciones por ejemplo en la cuestión BPP $\neq$ EXP.

Con el desarrollo de la dimensión de recursos acotados, ha sido natural plantearse si se podían demostrar *Small Span Theorems* utilizando dimensión en vez de medida. El primer resultado en esta línea fue negativo, y se debe al siguiente Teorema.

Teorema 3.3.1. [4] Para todo $A \in \mathrm{E}$,

$$\dim(\deg_m^p(A)|\mathrm{E}) = \dim(P_m(A)|\mathrm{E}),$$

donde $\deg_m^p(A) = P_m(A) \cap P_m^{-1}(A)$.

Así pues, como $\dim(\mathrm{E}|\mathrm{E}) = 1$, para los lenguajes $\leq_m^p$-completos, se tiene que

$$\dim(P_m^{-1}(A) \cap P_m(A)|\mathrm{E}) = \dim(P_m(A)|\mathrm{E}) = 1,$$

y por lo tanto, no se cumple un *Small Span Theorem* con dimensión en E y reducciones $\leq_m^p$.

Por otro lado, en [36] se demuestra que en el caso escalado:

1. Es posible demostrar un *Small Span Theorem* para dimensión con escala de grado -3 y $\leq_m^p$-reducciones:

   Teorema 3.3.2. Sea $\mathcal{C} \in \{\mathrm{E}, \mathrm{EXP}, \mathrm{ESPACE}, \mathrm{EXPSPACE}\}$. Para cada $A \in \mathcal{C}$

   $$\dim^{(-3)}(P_m^{-1}(A)|\mathcal{C}) = 0$$

   ó

   $$\dim^{(-3)}(P_m(A)|\mathcal{C}) = 0.$$

2. No es posible mejorar este resultado para escalas de orden -2:

   Teorema 3.3.3. Sea $\mathcal{C} \in \{\mathrm{E}, \mathrm{EXP}, \mathrm{ESPACE}, \mathrm{EXPSPACE}\}$. Para cada $A \in \mathcal{C}$ y $-2 \leq i \leq 2$,

   $$\dim^{(i)}(\deg_m^p(A)|\mathcal{C}) = \dim^{(i)}(P_m(A)|\mathcal{C})$$

3. Estos resultados tienen implicaciones en las cuestiones BPP $\neq$ P y P $\neq$ PSPACE.

Siguiendo esta línea de investigación sobre *Small Span Theorems* para dimensión con escala, en esta sección se estudiará el comportamiento de las reducciones P/poly-Turing en la clase ESPACE. Estas reducciones, denotadas como $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$, son reducciones de Turing que se computan por una familia no uniforme de circuitos de tamaño polinomial (ver Sección 1.3.3 de los preliminares). En particular, se estudiaran los *lower* y *upper spans* con respecto a estas reducciones, más formalmente:

Definición 3.3.4. Sea $A \subseteq \{0,1\}^*$.

1. Se define el $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$- lower span de A como

   $$(\mathrm{P/poly})_{\mathrm{T}}(A) = \{B \subseteq \{0,1\}^* | B \leq_{\mathrm{T}}^{\mathrm{P/poly}} A\}.$$

2. Se define el $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$- upper span de A como

$$(\mathrm{P/poly})_{\mathrm{T}}^{-1}(A) = \{B \subseteq \{0,1\}^* | A \leq_{\mathrm{T}}^{\mathrm{P/poly}} B\}.$$

El siguiente teorema mejora el resultado obtenido por Lutz en [63], donde se establece un *Small Span Theorem* para medida con reducciones $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$ en ESPACE.

Teorema 3.3.5. Para cada $A \in \mathrm{ESPACE}$,

$$\dim^{(1)}(\,(\mathrm{P/poly})_{\mathrm{T}}(A) \mid \mathrm{ESPACE}) = 0$$

o

$$\dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(A)\,) = 0.$$

Demostración. Sea

$$Z = \left\{ A \subseteq \{0,1\}^* \;\middle|\; \dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(A)\,) > 0 \right\}.$$

Es suficiente demostrar que

$$\dim^{(1)}(\,Z \mid \mathrm{ESPACE}) = 0. \tag{3.3.1}$$

En efecto, si la ecuación (3.3.1) se cumple, entonces podemos considerar dos casos:

1. Si $(\mathrm{P/poly})_{\mathrm{T}}(A) \cap \mathrm{ESPACE} \subseteq Z$.

   Entonces se sigue que

   $$\dim^{(1)}((\mathrm{P/poly})_{\mathrm{T}}(A) \mid \mathrm{ESPACE}) \leq \dim^{(1)}(\,Z \mid \mathrm{ESPACE}) = 0.$$

2. Si $(\mathrm{P/poly})_{\mathrm{T}}(A) \cap \mathrm{ESPACE} \nsubseteq Z$.

   Entonces existe un lenguaje $B \in (\mathrm{P/poly})_{\mathrm{T}}(A) \cap \mathrm{ESPACE}$ tal que $B \notin Z$. Como $B \leq_{\mathrm{T}}^{\mathrm{P/poly}} A$, tenemos que $(\mathrm{P/poly})_{\mathrm{T}}^{-1}(A) \subseteq (\mathrm{P/poly})_{\mathrm{T}}^{-1}(B)$ y por lo tanto, por ser la dimensión con escala monótona,

   $$\dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(A)\,) \leq \dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(B)\,).$$

   Como $B \notin Z$, entonces $\dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(B)\,) = 0$, de lo que se sigue que $\dim_{\mathrm{pspace}}^{(-3)}(\,(\mathrm{P/poly})_{\mathrm{T}}^{-1}(B)\,) = 0$.

Luego, basta demostrar que se cumple la ecuación (3.3.1) para tener el resultado. Esta demostración se basará en la demostración del Teorema 4.5 en [63] y se usarán los siguiente conceptos y notación.

Para cada $r \in \mathbb{N}$, definimos las funciones $a_r, b_r : \mathbb{N} \to \mathbb{N}$ como

$$a_r(n) = n^r + r \quad \text{y} \quad b_r(n) = \sum_{i=0}^{n} a_r(i).$$

Sea ADV_r la clase de todas las funciones *advice* $h : \mathbb{N} \to \{0,1\}^*$ que satisfacen $|h(n)| = a_r(n)$ para todo $n \in \mathbb{N}$. Para todos los $A, B \subseteq \{0,1\}^*$ que satisfacen $A \leq_{\mathrm{T}}^{\mathrm{P/poly}} B$, existen $r, k \in \mathbb{N}$ y $h \in \mathrm{ADV}_r$ tales que

$$A = L(M_k^B/h),$$

donde

$$L(M_k^B/h) = \{w \in \{0,1\}^* \,|M_k^B \text{ acepta } \langle w, h(|w|)\rangle\}\},$$

y M_k es la k-esima máquina de Turing con oráculo en tiempo polinómico.

Una *función parcial* $a_r(n)$-*advice* es una función finita

$$h' : \{0, 1, \ldots, k-1\} \to \{0,1\}^*$$

para algún $k \in \mathbb{N}$, tal que para todo $0 \leq n < k$, $|h'(n)| = a_r(n)$. Para cada función parcial $a_r(n)$-*advice* h', el *cilindro generado por* h' es

$$\mathrm{CYL}(h') = \{h \in \mathrm{ADV}_r \mid h \upharpoonright \{0, 1, \ldots, k-1\} = h'\},$$

donde $h \upharpoonright \{0, 1, \ldots, k-1\}$ denota la restricción de h al dominio $\{0, 1, \ldots, k-1\}$. La *probabilidad* de este cilindro se define como

$$\Pr(\mathrm{CYL}(h')) = \prod_{n=0}^{k-1} 2^{-a_r(n)}.$$

Para cada $r \in \mathbb{N}$, usaremos el espacio de probabilidad

$$\Omega_r = \mathrm{ADV}_r \times \mathbf{C}.$$

En este espacio usaremos la medida producto de la anterior medida de probabilidad en ADV_r y de la distribución uniforme en $\mathbf{C}$. Para cada $r, k, j \in \mathbb{N}$, definimos el evento $\mathcal{E}_{r,k,j}^A \subseteq \Omega_r$ como

$$\mathcal{E}_{r,k,j}^A = \{(h, B) \mid (\forall\, 0 \leq i < j)\ [\![s_i \in A]\!] = [\![s_i \in L(M_k^B/h)]\!]\}.$$

Para cada $r, k, j \in \mathbb{N}$ y $A \subseteq \{0,1\}^*$, sea

$$N_A(r,k,j) = \left| \left\{ i < j \;\middle|\; \Pr(\mathcal{E}^A_{r,k,i+1}) \leq \frac{1}{2}\Pr(\mathcal{E}^A_{r,k,i}) \right\} \right|.$$

Entonces para todo $r, k, j \in \mathbb{N}$ y $A \subseteq \{0,1\}^*$, tenemos

$$\Pr(\mathcal{E}^A_{r,k,j}) \leq 2^{-N_A(r,k,j)}.$$

Sea $A \subseteq \{0,1\}^*$ y sean s, $\delta > 0$ números racionales, definimos la $s^{(-3)}$-gala $d^A_{s,\delta} : \{0,1\}^* \to [0,\infty)$ como

$$d^A_{s,\delta}(w) = 2^{-g_3(|w|,1-s)} \sum_{r=0}^{\infty}\sum_{k=0}^{\infty}\sum_{j=0}^{\infty} 2^{-(r+k)/4-j^\delta} \cdot d^A_{r,k,j}(w),$$

donde para todo $r, k, j \in \mathbb{N}$, $d^A_{r,k,j}$ es la martingala

$$d^A_{r,k,j}(w) = \begin{cases} 2^{|w|}\Pr(\mathrm{ADV}_r \times \mathbf{C}_w \mid \mathcal{E}^A_{r,k,j}) & \text{si } \Pr(\mathcal{E}^A_{r,k,j}) > 0 \\ 1 & \text{si } \Pr(\mathcal{E}^A_{r,k,j}) = 0. \end{cases}$$

Es fácil ver que $d^A_{s,\delta}$ es pspace-calculable si $A \in \mathrm{ESPACE}$.

Sea $A, B \subseteq \{0,1\}^*$, $k, r \in \mathbb{N}$, y $h \in \mathrm{ADV}_r$ tales que $A = L(M^B_k/h)$. Al existir una cota de tiempo polinómico en M_k y una cota polinómica sobre la longitud en h, entonces existe una constante $c \in \mathbb{N}$ de modo que todas las preguntas de $(M^B_k/h)(s_i)$ tienen longitud estrictamente acotada por $|s_i|^c$ para i's suficientemente grandes. Sea $n(i) = \lceil \log(i+2) - 1 \rceil$, entonces $|s_i| = n(i)$ para todo i. Fijemos ahora $j \in \mathbb{N}$. Si elegimos

$$l = 2^{(\log(j+1))^c},$$

entonces todas las preguntas de $L(M^B_k/h)(s_i)$ para $0 \leq i < j$ serán sobre $s_0, s_1, \ldots, s_{l-1}$. Es decir, $A[0..j-1]$ se determina por $B[0..l-1]$. Notar que

$$j + 1 = 2^{(\log l)^{\frac{1}{c}}}.$$

Sea $h_j = h \upharpoonright \{0, 1, \ldots, n(j-1)\}$. Entonces h_j es una restricción de h que proporciona "consejo"para todos las entradas $s_0, \ldots, s_{j-1}$. Se sigue que $\mathrm{CYL}(h_j) \times \mathbf{C}_{B[0..l-1]} \subseteq \mathcal{E}^A_{r,k,j}$, luego podemos deducir como en [63] que

$$\Pr(\mathcal{E}^A_{r,k,j} \mid \mathrm{ADV}_r \times \mathbf{C}_{B[0..l-1]}) \geq 2^{-b_r(n(j))},$$

y por lo tanto tenemos que

$$d^A_{r,k,j}(B[0..l-1]) \geq 2^{N_A(r,k,j)-b_r(n(j))}.$$

Sea $\epsilon > \delta > 0$ y

$$X_\epsilon = \{A \subseteq \{0,1\}^* \mid (\forall k)(\forall r)(\forall^\infty j) N_A(r,k,j) > j^\epsilon\}.$$

Veamos que $X_\epsilon \cap \text{ESPACE} \subseteq Z^c$, es decir, que

$$\dim^{(-3)}_{\text{pspace}}(\ (\text{P/poly})^{-1}_{\text{T}}(A)\ ) = 0 \tag{3.3.2}$$

para cada $A \in X_\epsilon \cap \text{ESPACE}$. Para ello, sea $A \in X_\epsilon \cap \text{ESPACE}$ y $B \in (\text{P/poly})^{-1}_{\text{T}}(A)$. Entonces existen $k, r \in \mathbb{N}$ y $h \in \text{ADV}_r$ tales que $A = L(M^B_k/h)$. Sea j suficientemente grande como para asegurar que $N_A(r,k,j) > j^\epsilon$. Entonces, definiendo c y l como antes, tenemos que

$$\begin{aligned}
\log d^A_{s,\delta}(B[0..l-1]) &\geq \log d^A_{r,k,j}(B[0..l-1]) - g_3(l,1-s) - (r+k)/4 - j^\delta \\
&\geq j^\epsilon - b_r(n(j)) - g_3(l,1-s) - (r+k)/4 - j^\delta \\
&= \left(2^{(\log l)^{1/c}} - 1\right)^\epsilon - b_r(n(j)) - 2^{2^{(\log\log l)^{(1-s)}}} - (r+k)/4 - \left(2^{(\log l)^{1/c}} - 1\right)^\delta.
\end{aligned}$$

Como r y k son constantes, se sigue que $B \in S^\infty[d^A_{s,\delta}]$. Por lo tanto $(\text{P/poly})^{-1}_{\text{T}}(A) \subseteq S^\infty[d^A_{s,\delta}]$. Como $A \in \text{ESPACE}$, se tiene que $d^A_{s,\delta}$ es pspace-calculable, luego $\dim^{(-3)}_{\text{pspace}}((\text{P/poly})^{-1}_{\text{T}}(A)) \leq s$. Como esto se cumple para un $s > 0$ arbitrario, obtenemos la igualdad (3.3.2).

Veamos ahora que para cada $\epsilon > 0$,

$$\dim^{(1)}_{\text{pspace}}(X^c_\epsilon) \leq \epsilon. \tag{3.3.3}$$

Sea $A \in X^c_\epsilon$. Entonces existen $r, k \in \mathbb{N}$ tales que $N_A(r,k,j) \leq j^\epsilon$ para infinitos $j \in \mathbb{N}$. Notar que $N_A(r,k,j)$ se determina por $A[0..j-1]$. Para cada $j \in \mathbb{N}$, sea

$$Z_{r,k,j} = \{B[0..j-1] \mid N_B(r,k,j) \leq j^\epsilon\} \subseteq \{0,1\}^j.$$

Podemos acotar el tamaño de $Z_{r,k,j}$ como

$$|Z_{r,k,j}| \leq j^\epsilon \binom{j}{j^\epsilon} 2^{j^\epsilon} \leq j^\epsilon \cdot 2^{\mathcal{H}(j^{\epsilon-1})j + j^\epsilon}$$

puesto que podemos especificar un elemento del conjunto identificando primero al menos j^ϵ posiciones i en las cuales $\mathcal{E}^A_{r,k,i+1} \leq \frac{1}{2}\mathcal{E}^A_{r,k,i}$ y luego usando j^ϵ bits para especificar cual de las dos posibilidades

usar para el i-esimo bit en caso de $\mathcal{E}^A_{r,k,i+1} = \frac{1}{2}\mathcal{E}^A_{r,k,i}$. Entonces

$$\mathcal{H}(j^{\epsilon-1})j + j^\epsilon + \log j$$

bits son suficientes para identificar cada secuencia en $Z_{r,k,j}$, donde $\mathcal{H}(x)$ es la entropía binaria $\mathcal{H}(x) = x\log x + (1-x)\log(1-x)$. A partir de esta descripción con codificaciones de r, k, y j podemos computar la secuencia usando espacio polinómico: para algún polinomio p tenemos que

$$KS^p(w) \leq \mathcal{H}(j^{\epsilon-1})j + j^\epsilon + 2\log j + \log r + \log k$$

para todo $w \in Z_{r,k,j}$. Tenemos un polinomio p que funciona para cada r, k y para cada $j \geq j_0(r,k)$ para algún $j_0(r,k)$.

Notar que

$$\begin{aligned}
\mathcal{H}(j^{\epsilon-1})j &= \left(j^{\epsilon-1}\log j^{1-\epsilon} + (1-j^{\epsilon-1})\log\frac{1}{1-j^{\epsilon-1}}\right)j \\
&= j^\epsilon(1-\epsilon)\log j + j(1-j^{\epsilon-1})\log\left(1+\frac{j^{\epsilon-1}}{1-j^{\epsilon-1}}\right) \\
&\leq j^\epsilon(1-\epsilon)\log j + j(1-j^{\epsilon-1})\frac{j^{\epsilon-1}}{1-j^{\epsilon-1}}\log e \\
&= j^\epsilon[(1-\epsilon)\log j + \log e].
\end{aligned}$$

Se sigue entonces que $\mathcal{KS}^p_{(1)}(A) \leq \epsilon$ puesto que A satisface $A[0..j-1] \in Z_{r,k,j}$ infinitamente a menudo. Como $A \in X_\epsilon$ es arbitraria y el polinomio p no depende de A, tenemos que $\mathcal{KS}^{\text{pspace}}_{(1)}(X_\epsilon) \leq \epsilon$. Por el Teorema 3.1.7, se tiene que la desigualdad (3.3.3) se cumple.

Hemos probado que $X_\epsilon \cap \text{ESPACE} \subseteq Z^c$ para todo $\epsilon \in (0,1)$. Esto implica que $Y^c \cap \text{ESPACE} \subseteq X^c_\epsilon$, luego

$$\dim^{(1)}(Z \mid \text{ESPACE}) = \dim^{(1)}_{\text{pspace}}(Z \cap \text{ESPACE}) \leq \dim^{(1)}_{\text{pspace}}(X^c_\epsilon) \leq \epsilon$$

para todo $\epsilon \in (0,1)$. Así pues, $\dim^{(1)}(Z \mid \text{ESPACE}) = 0$.

□

El Teorema 3.3.5 mejora el teorema de Juedes y Lutz puesto que $\dim^{(-3)}(X) < 1$ implica que la medida en pspace de X es cero. Además, en [36] (Teorema 1,4) se prueba que no son posibles teoremas de tipo *Small Span* para escala (-2) (Teorema 3.3.3 en esta tesis).

Así pues, no es posible sustituir la escala -3 por una mayor en el enunciado del Teorema 3.3.5.

Como consecuencia de las conexiones entre dimensión con escala y complejidad de Kolmogorov vistas en las secciones anteriores, tenemos el siguiente resultado.

Teorema 3.3.6. Sea $A \in \mathrm{ESPACE}$, si

$$\dim^{(1)}((\mathrm{P/poly})_{\mathrm{T}}(A)|\mathrm{ESPACE}) > 0$$

entonces

$$\mathcal{KS}^{(-3)}_{\mathrm{pspace}}((\mathrm{P/poly})^{-1}_{\mathrm{T}}(A)) = 0$$

Demostración. El teorema se sigue del Teorema 3.3.5 y el Lema 3.1.19. □

En particular para lenguajes $\leq^{\mathrm{P/poly}}_{\mathrm{T}}$-*hard*, se tiene el siguiente corolario.

Corolario 3.3.7. Sea $\mathcal{H}$ la clase de los lenguajes que son $\leq^{\mathrm{P/poly}}_{\mathrm{T}}$-*hard* para ESPACE. Entonces

$$\mathcal{KS}^{(-3)}_{\mathrm{pspace}}(\mathcal{H}) = 0.$$

Es decir, para cada $\epsilon > 0$ existe una constante c tal que para cada lenguaje $\leq^{\mathrm{P/poly}}_{\mathrm{T}}$-*hard* H,

$$KS^{2^{cn}}(H_{\leq n}) < 2^{n+1} - 2^{2^{(\log n)^{1-\epsilon}}} \text{ i.o. } n.$$

Es más, examinando la demostración del Teorema 3.3.5, se obtiene una cota mejor. Esta cota coincide con la cota superior dada por Juedes y Lutz en [47] para los lenguajes $\leq^{\mathrm{P/poly}}_{\mathrm{m}}$-*hard*.

Teorema 3.3.8. Existe una constante c tal que para cada lenguaje $H \leq^{\mathrm{P/poly}}_{\mathrm{T}}$-*hard* para ESPACE, existe algún $\epsilon > 0$ tal que

$$KS^{2^{cn}}(H_{\leq n}) < 2^{n+1} - 2^{n^{\epsilon}} \text{ i.o. } n.$$

Demostración. Sea $\epsilon = \frac{1}{2}$ y sea X_ϵ como en la demostración del Teorema 3.3.5. Como $\dim^{(1)}_{\mathrm{pspace}}(\mathrm{ESPACE}) = 1$, tenemos que $\mathrm{ESPACE} \not\subseteq X^c_\epsilon$. Sea $A \in \mathrm{ESPACE} \cap X^c_\epsilon$.

Sea B un lenguaje $\leq^{\mathrm{P/poly}}_{\mathrm{T}}$-*hard* para ESPACE. Entonces A se $\leq^{\mathrm{P/poly}}_{\mathrm{T}}$-reduce a B. Elegimos c de modo que para todos i suficientemente grande, todas las preguntas de esta reducción en la entrada s_i tengan longitud acotada por $|s_i|^c$. La demostración de la igualdad (3.3.2) (en la demostración del Teorema 3.3.5 muestra que la $s^{(-3)}$-gala $d^A_{s,\delta}$ tiene éxito en B.

Sea $\gamma \in (1 - \frac{\epsilon}{c}, 1)$. Definimos la siguiente $\gamma^{(-2)}$-gala d como

$$d(w) = 2^{g_3(|w|,1-s)-g_2(|w|,1-\gamma)} d^A_{s,\delta}(w).$$

Entonces, el cálculo para ver que $d^A_{s,\delta}$ tiene éxito en B cambia a

$$\log d(B[0 \ldots l-1]) \geq \left(2^{(\log l)^{1/c}} - 1\right)^{\epsilon} - b_r(n(j)) - 2^{(\log l)^{1-\gamma}} - (r+k)/4 - \left(2^{(\log l)^{1/c}} - 1\right)^{\delta}.$$

Como $1-\gamma < \frac{\epsilon}{c}$, d también tiene éxito en B. Así pues, $\dim^{(-2)}_{\text{pspace}}(H) \leq \gamma < 1$.

Sea $\alpha \in (\gamma, 1)$. Por el Lema 3.1.19 (tal como se aplica en el Teorema 3.1.21) obtenemos que

$$KS^{2^{dn}}(B_{\leq n}) < 2^{n+1} - 2^{n^{1-\alpha}}$$

para infinitos n's, donde d es una constante que no depende de B. □

Este resultado dice que los lenguajes $\leq^{\text{P/poly}}_{\text{T}}$-*hard* son inusualmente simples, puesto que para la mayoría de los lenguajes, lo contrario se cumple, incluso cuando permitimos cualquier cota en la complejidad de Kolmgorov.

Teorema 3.3.9. Para cualquier cota $\text{t} : \mathbb{N} \to \mathbb{N}$, la clase de todos los lenguajes A que satisfacen

$$KS^{t(2^n)}(A^{\leq n}) < 2^{n+1} - 2^{n^\epsilon} \text{ i.o. } n$$

para algún $\epsilon > 0$ tiene dimensión calculable de orden -3 cero.

Demostración. El resultado se sigue de la caracterización proporcionada por el Teorema 3.1.7. □

El Teorema 3.3.9 implica que casi todos los lenguajes decidibles satisfacen la cota

$$KS^{2^{cn}}(A^{\leq n}) \geq 2^{n+1} - 2^{n^\epsilon} \text{ a.e. } n$$

para cada $\epsilon > 0$, sin embargo, los lenguajes $\leq^{\text{P/poly}}_{\text{T}}$-*hard* tienen la propiedad opuesta por el Teorema 3.3.8. Es más, la mejor cota inferior conocida para los lenguajes $\leq^{\text{P/poly}}_{\text{T}}$-*hard* es mucho menor, en [47] se prueba que para todo lenguaje H $\leq^{\text{P/poly}}_{\text{T}}$-*hard* existe un $\epsilon > 0$ tal que

$$KS^{2^{n^\epsilon}}(H_{\leq n}) > 2^{n^\epsilon} \text{ a.e. } n.$$

Capítulo 4

Dimensión es compresión

Tal como se ha visto a lo largo de la tesis y tal como se demuestra en diversos artículos [77, 68, 33, 21, 38], las conexiones entre dimensión y Teoría de la Información (*algorithmic information*) son muy estrechas. En los casos de dimensión constructiva, dimensión calculable y dimensión en espacio polinómico se consiguen caracterizaciones a partir de la complejidad de Kolmogorov clásica y acotada en espacio y, en el caso de dimensión con estados finitos se consigue una caracterización en función de la compresión con estados finitos.

Sin embargo, en el caso de la dimensión en tiempo polinómico no parecen existir posibles caracterizaciones de ese tipo [40]. Esto no es extraño puesto que computar, incluso de manera aproximada, la complejidad de Kolmogorov acotada en tiempo parece que requiere una búsqueda exponencial (bajo las hipótesis habituales en Complejidad Computacional). La principal diferencia entre complejidad de Kolmogorov acotada en tiempo y en espacio es la reversivilidad. En el caso de cotas en espacio, la fase de codificación puede hacerse con cotas de espacio similares a la de decodificación.

En este capítulo se utiliza la noción usual de algoritmo de compresión para secuencias finitas para caracterizar la dimensión en tiempo polinómico. Un esquema de compresión en tiempo polinómico no es más que un par de algoritmos: un codificador y un decodificador, ambos trabajando en tiempo polinómico. Sin embargo, para hacer posible la caracterización será necesario trabajar con codificadores que no empiezan a trabajar exactamente desde cero cuando trabajan sobre la extensión de una entrada previa. Esta condición se formalizará en la sección 4.1.

El principal resultado de este capítulo es una caracterización exacta de la dimensión en tiempo polinómico como, asintóticamente, el mejor caso (es decir, infinitamente a menudo ó i.o.) de ratio de compresión que se consigue con este tipo de compresores en tiempo polinómico. Dualmente, se ve que la dimensión empaquetadora o fuerte (*Packing*) en tiempo polinómico corresponde al asintóticamente peor caso de ratio de compresión asintótico.

Gracias a esta nueva caracterización, varios resultados conocidos para dimensión en tiempo

polinómico pueden interpretarse como resultados de compresión. Por ejemplo, los lenguajes de una clase de p-dimensión 1 no pueden comprimirse (i.o.) en más que una cantidad sublineal. Así se obtienen resultados sobre la compresibilidad de lenguajes completos y autoreducibles.

Buhrman y Longprè dieron una caracterización de p-medida en términos de compresión en [13], pero en este caso los compresores se restringían a extensores y el codificador tenía varias alternativas, dentro de las cuales se encontraba la salida correcta. A partir de los resultados obtenidos en este capítulo se puede ver la p-dimensión como medida de información para secuencias infinitas, mientras que en el caso de p-medida únicamente es capaz de distinguir el caso extremo de las clases de medida 0 que son las más incompresibles.

Los resultados de este capítulo han sido publicados junto con Elvira Mayordomo en [61].

4.1. Codificadores que no empiezan desde cero

En esta sección se formalizará la idea de que un codificador "no empiece desde cero". Este tipo de codificadores cuando trabajan sobre extensiones cada vez más largas de una entrada proporcionan salidas que están restringidas de un modo que se verá más adelante. El caso más extremo de codificadores que cumplen esta propiedad son los simples extensores, es decir, codificadores donde $C(w)$ es siempre un prefijo de $C(wu)$. La restricción que se considerará en este capítulo es mucho más permisiva que la extensión.

Definición 4.1.1. Una función $C : \{0,1\}^* \to \{0,1\}^*$ es un *codificador en tiempo polinómico* si $\forall n$, $C : \{0,1\}^n \to \{0,1\}^*$ es inyectiva (ó $(C(w), |w|)$ es inyectiva) y puede calcularse en tiempo polinómico en la longitud de la entrada.

Se utilizarán codificadores libres de prefijos, es decir, codificadores C tales que el conjunto $C(\{0,1\}^n)$ es un conjunto prefijo para cada n. Esto significa que la codificación de una secuencia de longitud n nunca será prefijo de la codificación de otra secuencia de longitud n distinta. Nótese que, para las aplicaciones que interesarán en esta tesis (radios de compresión asintóticos), restringir los codificadores en este sentido no es significativo, ya que para cada codificador existe otro libre de prefijos con el mismo ratio de compresión.

Sin embargo, la Definición 4.1.1 es demasiado general para relacionar dimensión y compresión. Dicha definición no impone ninguna restricción en el comportamiento de C cuando trabaja en dos entradas y una de ellas es prefijo de la otra. Esto significa que, en general, podría ocurrir que para un codificador en tiempo polinómico, $|C(wu)|$ sea mucho más pequeño que $|C(w)|$ y también que $C(wu)$ y $C(w)$ no tuvieran ningún prefijo en común.

Permitir que esto ocurra es permitir demasiado grado de libertad a los codificadores. Para poder relacionar la dimensión en tiempo polinómico y la compresión se necesitará restringir los codificadores del siguiente modo: los valores que puede tomar $C(wu)$ y la longitud que puede llegar a tener para diferentes u's, estará controlada por la longitud de $|C(w)|$.

Más formalmente,

Definición 4.1.2. Decimos que un codificador en tiempo polinómico C *no empieza desde cero* si para todo $\epsilon > 0$ y para todas las secuencias finitas $w \in \{0,1\}^*$ (salvo quizá un número finito), existe $k = O(\log(|w|))$, $k > 0$, tal que

$$\sum_{|u|\leq k} 2^{-|C(wu)|} \leq 2^{\epsilon k} 2^{-|C(w)|}. \tag{4.1.1}$$

En esta tesis, se considerarán sólo codificadores que no empiezan desde cero.

Notar que cuando existe una constante k tal que $\sum_{|u|\leq k} 2^{-|C(wu)|} \leq 2^{-|C(w)|}$, la condición (4.1.1) es trivial. Sin embargo, en general, la cantidad $\sum_{|u|\leq k} 2^{-|C(wu)|}$ puede ser tan grande como 1, así que la condición (4.1.1) es una restricción propia de los codificadores.

Los primeros ejemplos que se considerarán serán aquellos para los cuales $C(w)$ y $C(wu)$ tienen un prefijo en común largo.

Nota 4.1.3. Sea un codificador C en tiempo polinómico verificando que, para todo $w, u \in \{0,1\}^*$, $C(w)$ y $C(wu)$ tienen un prefijo común de longitud al menos $|C(w)| - M\log(|w|)$, con $M \in \mathbb{N}$ fijo. Entonces, C es un codificador que no empiezan desde cero.

En efecto, usando que $(C(w), |w|)$ es inyectivo,

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

Ejemplo 4.1.4. Los siguientes codificadores en tiempo polinómico verifican la condición de la nota 4.1.3 y, por lo tanto, son codificadores que no empiezan desde cero.

1. Un extensor, es decir, un codificador que verifica $\forall\, w, w' \in \{0,1\}^*$

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

   En efecto, si C es un extensor, entonces C verifica la propiedad de tener prefijos comunes de la nota 4.1.3 y por lo tanto es un codificador que no empieza desde cero.

2. El algoritmo de compresión de datos de Lempel-Ziv para sus dos variantes más comunes vistas en el Capítulo 1 (Subsección 1.3.8). Notar que no son extensores.

   Veamos que verifican la condición de poseer prefijos comunes suficientemente largos de la Nota 4.1.3:

   1. Versión LZ_{78}: Sea $w \in \{0,1\}^*$ y $w_1 w_2 \dots w_n$ su análisis único válido. Entonces, dado $u \in \{0,1\}^*$, el análisis único válido de wu compartirá las primeras $n-1$ frases del análisis único válido de w y quizá difiera en la frase w_n. Esta frase se codifica con $\lceil \log n+1 \rceil +1$ bits, así pues, $\mathrm{LZ}_{78}(w)$ y $\mathrm{LZ}_{78}(wu)$ tienen un prefijo común de longitud al menos $|\mathrm{LZ}_{78}(w)| - (\lceil \log n + 1 \rceil + 1) \geq |\mathrm{LZ}_{78}(w)| - 3 \log |w|$.
   2. Versión LZ_{77}: Sea $w \in \{0,1\}^*$ y $w_1 w_2 \dots w_n$ la historia exhaustiva de w. Entonces, dado $u \in \{0,1\}^*$, la historia exhaustiva de wu compartirá las primeras $n-1$ frases de la historia exhaustiva de w y quizá difiera en la frase w_n. Esta frase, se codifica con $2\lceil \log |w| \rceil +1$ bits, así pues, $\mathrm{LZ}_{77}(w)$ y $\mathrm{LZ}_{77}(wu)$ tienen un prefijo común de longitud al menos $|\mathrm{LZ}_{77}(w)| - (2\lceil \log |w| \rceil + 1) \geq |\mathrm{LZ}_{77}(w)| - 3 \log |w|$.

Ejemplos de codificadores en tiempo polinómico que no empiezan desde cero son aquellos codificadores C que son crecientes en longitud y para los cuales podemos controlar, para todo w y todo $i \geq 0$, el número de secuencias finitas u que verifican $|C(wu)| = |C(w)| + i$. Más formalmente,

Nota 4.1.5. Los codificadores en tiempo polinómico C que satisfacen las siguientes dos condiciones, no empiezan desde cero.

i) Para todo $w, u \in \{0,1\}^*$,

$$|C(wu)| \geq |C(w)|.$$

ii) Para todo $\epsilon > 0$ y para todos los $w \in \{0,1\}^*$ (salvo quizá un número finito) existe un $k = O(\log(|w|))$ tal que $\forall i \geq 0$,

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

Para demostrar esto, lo que hacemos es suponer el peor de los casos, es decir, que en el sumatorio de la izquierda en la condición (4.1.1), $\sum_{|u| \leq k} 2^{-|C(wu)|}$, tengamos el máximo número de sumandos con $|C(wu)|$ el menor posible. Esto significa que sumaremos un total de $(2^{k\epsilon - \log k})$ veces $2^{-|C(w)|}$,

un total de $(2^{1+k\epsilon-\log k})$ veces $2^{-|C(w)|-1}$ y así sucesivamente hasta finalmente haber sumado los $2^{k+1}-1$ sumandos que se suman en el sumatorio de la izquierda. Notar que esto es posible puesto que

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

Así pues,

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

El siguiente resultado relaciona los codificadores en tiempo polinómico que no empiezan desde cero y la dimensión en tiempo polinómico.

Teorema 4.1.6. Sea $X \subseteq \mathbf{C}$,

1. Si existe un codificador en tiempo polinómico C que no empieza desde cero y para todo $A \in X$

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

entonces

$$\dim_p(X) \leq \alpha.$$

2. Si existe un codificador en tiempo polinómico C que no empieza desde cero y para todo $A \in X$

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

entonces

$$\mathrm{Dim}_p(X) \leq \alpha.$$

Demostración. Veremos el caso 1 puesto que el caso 2 es análogo.

Sea $s > \alpha$ y $\epsilon > 0$ de modo que $s - \alpha > 2\epsilon$. Sea N de modo que la condición (4.1.1) se cumple para cada secuencia finita $w \in \{0,1\}^{\geq N}$. Para cada una de estas secuencias w, sea $k = k(w,\epsilon) = O(\log(|w|))$ el menor k que verifica,

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

Sea $w = w_1 \dots w_n$ con $|w_1| = N$ y $|w_i| = k(w_1 \dots w_i - 1, \epsilon)$ para $i > 0$.

Definimos la función $d : \{0,1\}^* \to [0, \infty)$ del siguiente modo,

$$d(wu) := d(w)\frac{2^{-|C(wu)|}}{\sum_{|v|\leq k} 2^{-|C(wv)|}}2^{s|u|} \qquad \text{si } |u| = k(w, \epsilon),$$
$$d(w\tilde{u}) := \sum_{\tilde{u}\sqsubseteq u, |u|=k} d(wu)2^{s(|\tilde{u}|-|u|)} \qquad \text{si } |\tilde{u}| < k(w, \epsilon).$$

Veamos que d es una s-gala. Para ello tendremos que distinguir varios casos:

i) Sea $w = w_1 \dots w_n\tilde{u}$, donde w_i como antes. Denotemos $\tilde{w} = w_1 \dots w_n$. Consideremos en este caso que $0 < |\tilde{u}| < k(\tilde{w}, \epsilon)$ y $|\tilde{u}| + 1 < k(\tilde{w}, \epsilon)$. Entonces,

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

ii) Supongamos ahora que w es exactamente de la forma $w_1 \dots w_n$. En este caso, se tiene que,

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

Ahora bien, como $|u| = k$ tenemos que

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

y por lo tanto,

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

iii) Por último, suponer que $w = w_1 \dots w_n \tilde{u}$, donde w_i como antes. Denotemos $\tilde{w} = w_1 \dots w_n$. Consideremos en este caso que $0 < |\tilde{u}| < k(\tilde{w}, \epsilon)$ y $|\tilde{u}| + 1 = k(\tilde{w}, \epsilon)$. Entonces,

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

Así pues, d es una s-gala. Además d es calculable en tiempo polinómico. En efecto, el número de sumandos que aparecen en la definición de d es a lo más $2^{k(w,\epsilon)+1}$ y al ser $k(w, \epsilon) = O(\log(|w|))$, se tiene que el número de sumandos es polinómico en la longitud de la entrada. Además, cada sumando y el resto de los términos que aparecen en la definición de d son calculables en tiempo polinómico.

Por otro lado, desarrollando la definición de d, tenemos que si $w = w_1 w_2 \dots w_n$ con $|w_1| = N$ y $|w_i| = k(w_1 \dots w_{i-1}, \epsilon)$, entonces,

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

Por la condición (4.1.1),

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

donde a es el mínimo de

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

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

Veamos que d tiene éxito en X. Para ello sea $A \in X$. Entonces, por hipótesis,

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

luego existe una secuencia $(b_n)_{n \in \mathbb{N}}$ de números naturales que verifica

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

Es decir, existen infinitos $n's$ para los cuales

$$|C(A[0 \dots b_n - 1])| \leq b_n(\alpha + \epsilon). \tag{4.1.2}$$

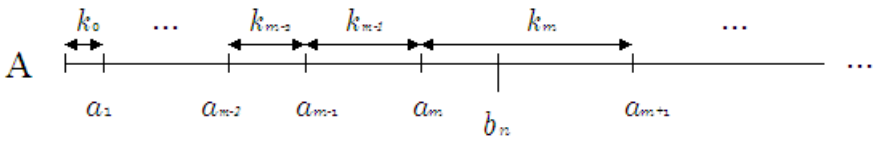

Figura 4.1: Distribución de los a_i's en relación con b_n y k_i dentro de la secuencia A.

Sea $(a_n)_{n\in\mathbb{N}}$ una secuencia definida recursivamente como sigue,

$$a_1 = k_0 = N,$$
$$a_{i+1} = a_i + k_i \qquad \text{para } i > 1,$$

donde $k_i = k(A[0\ldots a_i - 1], \epsilon)$, es decir, $k_i = O(\log a_i)$.

Entonces

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

Para cada n, sea $m \in \mathbb{N}$ tal que $a_m < b_n \leq a_{m+1}$ (ver figura 4.1).

Como $b_n - a_m \leq k_m$ se tiene que por la condición (4.1.1)

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

y por lo tanto,

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

Entonces, para todo n salvo un número finito,

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

Y d tiene éxito en X, puesto que $i)$ $k_m = O(\log a_m)$ y $ii)$ $s - \alpha > 2\epsilon$. Así pues, hemos demostrado que $\dim_{\mathrm{p}}(X) \leq s$. Al haber tomado $\epsilon > 0$ cualquiera, también se puede tomar $s > \alpha$ arbitraria y podemos concluir el resultado. □

Este teorema establece que la dimensión en tiempo polinómico es menor o igual que el radio de compresión de cualquier codificador en tiempo polinómico que no empieza desde cero.

4.2. Teorema principal

El teorema principal de este capítulo demuestra una caracterización exacta de la dimensión en tiempo polinómico en términos de los codificadores en tiempo polinómico que no empiezan desde cero. Más aún, se demuestra que la caracterización se cumple para una familia más restrictiva de codificadores: los codificadores que son reversibles en tiempo polinómico (compresores).

Como consecuencia se obtiene que el ratio de compresión de cualquier codificador en tiempo polinómico que no empieza desde cero, puede alcanzarse por un codificador en tiempo polinómico que puede decodificarse también en tiempo polinómico. Es decir, un codificador en tiempo polinómico que no empieza desde cero no comprime mejor que uno que pueda decodificarse en tiempo polinómico.

Esta caracterización se cumple para el ratio de compresión en caso mejor y caso peor, correspondiendo a p-dimensión y p-dimensión fuerte.

La siguiente es una definición formal de lo que significa que un codificador pueda decodificarse en tiempo polinómico.

Definición 4.2.1. 1. Un par de funciones (C, D) (C el codificador, D el decodificador) $C, D : \{0,1\}^* \to \{0,1\}^*$ es un *compresor en tiempo polinómico* si:

   (*i*) C y D pueden calcularse en tiempo polinómico en la longitud de su correspondiente entrada.

   (*ii*) Para todo $w \in \{0,1\}^*$,

$$D(C(w), |w|) = w.$$

2. Un *compresor en tiempo polinómico* (C, D) *no empieza desde cero* si el codificador C no empieza desde cero.

A continuación se definen los conceptos de compresión i.o. (*infinitely often*) y a.e. (*almost everywhere*) para conjuntos de secuencias infinitas como el peor ratio de compresión asintótico en el caso de i.o. y el mejor ratio en el caso de a.e.

Definición 4.2.2. Sea $\alpha \in [0,1]$ y $X \subseteq \mathbf{C}$,

1. Se dice que X es α-*i.o. compresible en tiempo polinómico* si existe un compresor en tiempo polinómico (C, D) que no empieza desde cero y que verifica, para todo $A \in X$,

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

2. Se dice que X es α-*a.e. compresible en tiempo polinómico* si existe un compresor en tiempo polinómico (C, D) que no empieza desde cero y que verifica, para todo $A \in X$,

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

Consecuentemente con la definición anterior, se definen los conceptos de incompresibilidad i.o. y a.e.

Definición 4.2.3. Sea $X \subseteq \mathbf{C}$,

1. Se dice que X es *i.o. incompresible en tiempo polinómico* si para cada compresor en tiempo polinómico (C, D) que no empieza desde cero, existe una secuencia infinita $A \in X$ que verifica

$$\liminf_n \frac{|C(A[0 \ldots n-1])|}{n} = 1.$$

2. Se dice que X es *a.e. incompresible en tiempo polinómico* si para cada compresor en tiempo polinómico (C, D) que no empieza desde cero, existe una secuencia infinita $A \in X$ que verifica

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

El siguiente teorema es el resultado principal de este capítulo.

Teorema 4.2.4 (Teorema principal)**.** Sea $X \subseteq \mathbf{C}$,

$$\begin{aligned} \dim_{\mathrm{p}}(X) &= \text{ínf}\{\, \alpha \,|\, X \text{ es } \alpha\text{-i.o. compresible en tiempo polinómico}\}. \\ \mathrm{Dim}_{\mathrm{p}}(X) &= \text{ínf}\{\, \alpha \,|\, X \text{ es } \alpha\text{-a.e. compresible en tiempo polinómico}\}. \end{aligned}$$

La demostración de este teorema se estructura en la demostración de dos teoremas: el primero es el Teorema 4.1.6 que establece que la compresión proporciona una cota superior a la dimensión. El segundo se enunciará más adelante y demuestra que la dimensión proporciona una cota superior a la compresión.

Para demostrar esto último (Teorema 4.2.9) se necesitará utilizar que en la definición de la dimensión en tiempo polinómico es suficiente considerar una familia simple de galas que requiere

poca exactitud. A continuación aplicamos una generalización de la codificación aritmética a esta gala simple.

Para probar esto, se utiliza el siguiente resultado.

Lema 4.2.5. [75] Sea d_1 una martingala. Sea $c : \{0,1\}^* \to [0,+\infty)$ una función calculable en tiempo polinómico tal que para cada $w \in \{0,1\}^*$,

$$|c(w) - d_1(w)| \leq 2^{-|w|}.$$

Sea $d_2 : \{0,1\}^* \to [0,\infty)$ definida recursivamente como sigue,

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

Entonces d_2 es una martingala en p que verifica

$$|d_1(w) - d_2(w)| \leq 4,$$

$$d_2(w) > 3/2 \quad \forall w.$$

Utilizando este resultado, se obtiene el siguiente lema.

Lema 4.2.6. Sea $X \subseteq \mathbf{C}$. Si $\dim_{\mathrm{p}}(X) = \alpha$ entonces $\forall s > \alpha$ existe una s-gala d con $X \subseteq S^{\infty}[d]$ tal que, para todo $w \in \{0,1\}^*$, existen $m_w, n_w \in \mathbb{N}^+$ con $n_w \leq |w| + 1$ y

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

Demostración. Si $\dim_{\mathrm{p}}(X) = \alpha$ entonces $\forall s > \alpha$ existe una s-gala d' calculable en tiempo polinómico que tiene éxito en X. Podemos suponer, sin perdida de generalidad, que $d'(\lambda) = 1$.

Sea d_1 la martingala calculable en tiempo polinómico definida como

$$d_1(w) = 2^{(1-s)|w|} d'(w).$$

Sea $\mathbb{D}$ el conjunto de números diádicos y sea la función $c : \{0,1\}^* \to \mathbb{D}$ definida como $c(w) = m'_w 2^{-n'_w}$ donde

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

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

Notar que $n'_w \leq |w|+1$ porque dentro de un intervalo de longitud $2^{-|w|}$ existe al menos un número diádico $m2^{-n}$ con $n = |w|+1$. Notar que, por esto mismo, c se puede calcular en tiempo polinómico.

Sea d_2 obtenida a partir de la martingala d_1, tal como se hace en el Lema 4.2.5. Entonces existen $m_w, n_w \in \mathbb{N}$ tales que $d_2(w) = m_w 2^{-n_w}$ con $n_w \leq |w|+1$. En efecto, lo probaremos por inducción:

i) Si $|w| = 0$ entonces $d_2(\lambda) = 3 = 3 \cdot 2^{-0}$.

ii) Si es cierto para $|w|$ entonces

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

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

Utilizando d_2, definimos la s-gala d del siguiente modo

$$d(w) = 2^{(s-1)|w|} d_2(w), \qquad \forall w \in \{0,1\}^*.$$

Entonces, es claro que d es calculable en tiempo polinómico y además, para todo $w \in \{0,1\}^*$ se tiene que

i) Por definición, $d(w)2^{-|w|s} = 2^{-|w|}d_2(w) = m_w 2^{-(n_w+|w|)}$ es un número diádico y $n_w \leq |w|+1$.

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

□

Es decir, si $\dim_{\text{p}}(X) < s$, entonces existe una s-gala computable en tiempo polinómico d como en el lema anterior. A partir de esa s-gala se define un compresor en tiempo polinómico que no empieza desde cero. Básicamente, la idea para el codificador C es asociar a cada $w \in \{0,1\}^*$ un intervalo de tamaño proporcionalmente relacionado con $d(w)$. Por las propiedades de d, los extremos de ese intervalo son racionales diádicos y usando el siguiente lema se codificará cada intervalo con una secuencia z y se definirá $C(w) = z$.

Lema 4.2.7. Sean a, b números diádicos con $0 \leq a < b \leq 1$ y sea $I = [a, b)$ un intervalo de longitud $l \in (0,1)$. Entonces existe una secuencia finita $z \in \{0,1\}^*$ de longitud $-\lfloor \log l \rfloor + 1$ tal que $a < 0.z < b$ y z puede calcularse en tiempo polinómico en $|z|$.

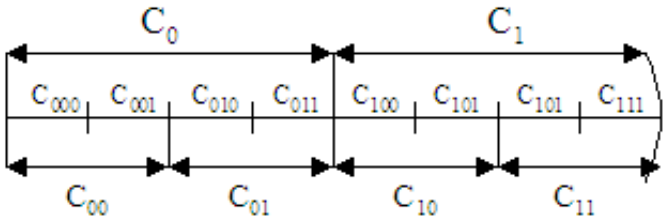

Figura 4.2: Ejemplo de la distribución de cilindros en $[0, 1)$.

Demostración. Sea $n \in \mathbb{N}$ tal que $\lfloor \log l \rfloor = -n$. Dividimos el intervalo $[0, 1)$ en intervalos de longitud 2^{-n-1}. Notar que cada uno de esos intervalos, son exactamente los cilindros C_w con $w \in \{0, 1\}^{n+1}$ (ver figura 4.2).

Al ser

$$\frac{1}{2^n} < 2^{\lceil \log l \rceil} = \frac{1}{2^{n-1}},$$

se tiene que al menos uno de los extremos de estos intervalos está en el intervalo (a, b). Además, a lo más puede haber dos extremos de estos intervalos dentro de (a, b). El extremo más pequeño será el $0.z$ que buscamos.

Lo que haremos será representar z de modo que C_z sea el intervalo cuyo extremo izquierdo sea $0.z$ (ver Ejemplo 4.2.8). Para ello, si $z = z_1 z_2 \dots z_m$, tenemos que

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

Veamos que podemos calcular z en tiempo polinómico. Para ello, calculamos $z = z_1 z_2 \dots z_{n+1}$ bit a bit del siguiente modo:

$$z_1 = \begin{cases} 0 & \text{si } a < 1/2. \\ 1 & \text{si } a \geq 1/2. \end{cases}$$

Conocidos los bits $z_i \dots z_i$ definimos z_{i+1} de modo que

$$z_{i+1} = \begin{cases} 0 & \text{si } a < 0.z_1 \dots z_i. \\ 1 & \text{si } a \geq 0.z_1 \dots z_i. \end{cases}$$

Obtendremos z cuando $a < 0.z \leq b$ y $|z| = -\lfloor \log l \rfloor + 1$ □

Ejemplo 4.2.8. Sea la siguiente situación: $a = 3 \cdot 2^{-2}$ y $b = 7 \cdot 2^{-3}$. En ese caso $l = b - a = 2^{-3}$ y por lo tanto $\lfloor \log l \rfloor = -3$.

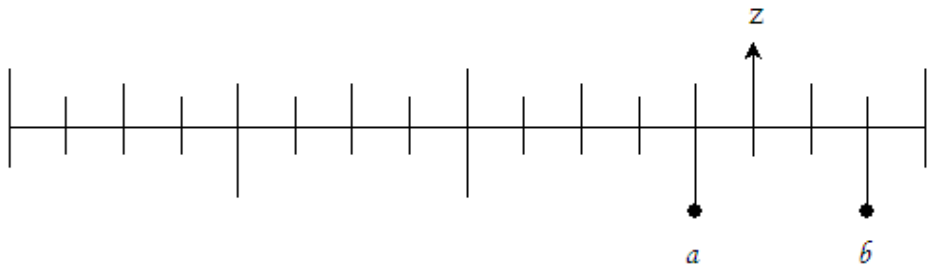

Figura 4.3: En este ejemplo particular, $z = 1110$.

Dividimos el intervalo $[0, 1)$ en intervalos de longitud 2^{-4}, tal como se ve en la figura 4.3.

Para calcular z bit a bit,

$$\begin{aligned} z_1 &= 1, && \text{puesto que } a \geq 1/2. \\ z_2 &= 1, && \text{puesto que } a \geq 0.z_1 = 1/2. \\ z_3 &= 1, && \text{puesto que } a \geq 0.z_1 z_2 = 3/4. \\ z_4 &= 0, && \text{puesto que } a < 0.z_1 z_2 z_3 = 7/8. \end{aligned}$$

Y, por lo tanto, $z = 1110$ y $|z| = -\lfloor \log l \rfloor + 1 = 4$.

Con estos resultados es posible demostrar que la dimensión proporciona una cota superior a la compresión y por lo tanto, demostrar el teorema principal de este capítulo.

Teorema 4.2.9. Sea $X \subseteq \mathbf{C}$,

$$\dim_{\mathrm{p}}(X) < s \Rightarrow X \text{ es } s - i.o. \text{ compresible en tiempo polinómico.}$$

$$\mathrm{Dim}_{\mathrm{p}}(X) < s \Rightarrow X \text{ is } s - a.e. \text{ compresible en tiempo polinómico.}$$

Demostración. Veamos la demostración de la primera desigualdad; la demostración para dimensión fuerte es análoga.

Sea s tal que $\dim_{\mathrm{p}}(X) < s$, entonces por el Lema 4.2.6 existe una s-gala d' calculable en tiempo polinómico tal que:

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

ii) Para toda secuencia finita $w \in \{0,1\}^*$, existen $m_w, n_w \in \mathbb{N}^+$ con $n_w \leq |w|+1$ tales que

$$d'(w)2^{-|w|s} = m_w 2^{-(n_w+|w|)}. \tag{4.2.1}$$

Además, podemos suponer, sin perdida de generalidad, que $d'(\lambda) = 1$.

Definimos para cada $w \in \{0,1\}^*$,

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

d es una martingala calculable en tiempo polinómico tal que,

i) Para toda secuencia infinita $A \in X$, $\quad d(A[0\ldots n-1]) > 2^{(1-s)n}$ *i.o.* n

ii) Por la igualdad (4.2.1), para toda secuencia finita $w \in \{0,1\}^*$,

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

Denotemos por $y < w$ que $y \in \{0,1\}^*$ preceda a w en el orden lexicográfico. Definimos la función $h : \{0,1\}^* \to \mathbb{R}$ como sigue:

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

Sea $suc(w)$ el sucesor de w en el orden lexicográfico. Notar que $h(w)$ es un número diádico $m2^{-n}$ con $n \leq 2|w|+1$, luego, por el Lema 4.2.7, existe una secuencia finita $z \in \{0,1\}^*$ tal que $|z| \leq 2|w|+2$ y

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

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

En efecto,

i) Si $w \neq 11\ldots1$, entonces

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

ii) Cuando $w = 11\ldots1$, entonces

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

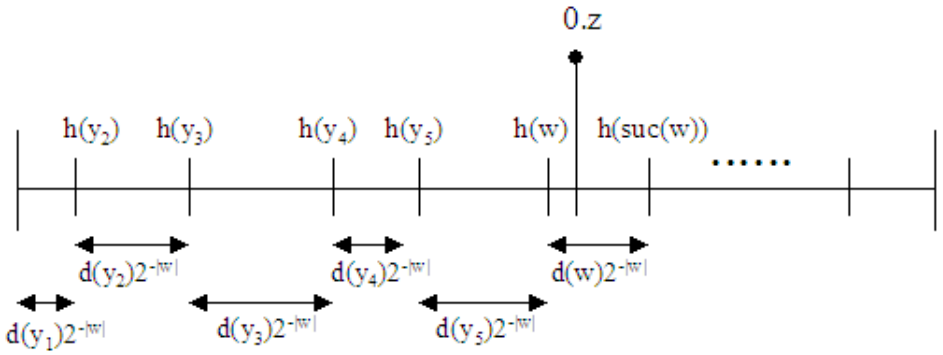

Figura 4.4: En este ejemplo, w es el sexto elemento de longitud $|w|$ en el orden lexicográfico.

Donde la última igualdad se obtiene de aplicar el Lema 1.4.21 que, al ser d martingala exacta, establece que

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

Luego, en cualquier caso,

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

Y por lo tanto,

$$-\lfloor \log l \rfloor + 1 = -\lfloor \log m_w \rfloor + n_w + |w| + 1 \leq 2|w| + 1.$$

Sea z_w la primera secuencia finita en orden lexicográfico tal que $h(w) < 0.z < h(suc(w))$ (ver figura 4.4). Definimos el codificador C como $C(w) = z_w$. Es claro que se puede calcular en tiempo polinómico ya que z_w se calcula en tiempo polinómico en la longitud de z_w (Lema 4.2.7) y $|z_w| \leq 2|w| + 2$.

Para definir el decodificador D, sea $z \in \{0,1\}^*$ y $n \in \mathbb{N}$, entonces para generar una secuencia finita de longitud n a partir de (z, n), simularemos la martingala empezando en λ y en prefijos sucesivamente más largos. Supongamos que ya hemos generado la secuencia w, entonces si $h(w0) \leq 0.z < h(w1)$, añadimos un 0 a w y si $h(w1) \leq 0.z$, añadimos un 1 a w. Continuamos así hasta que $|w| = n$. Al final de este proceso, tendremos una secuencia w de longitud n tal que $h(w) \leq 0.z < h(succ(w))$.

A continuación demostraremos que el compresor en tiempo polinómico (C, D) no empieza desde cero.

Notar que para cada w, el intervalo $[h(w), h(suc(w)))$ tiene exactamente la longitud $d(w)2^{-|w|}$. Entonces, por el Lema 4.2.7, existe una secuencia finita z de longitud $-\lfloor \log(2^{-|w|}d(w)) \rfloor + 1$ tal que $h(w) < 0.z < h(succ(w))$. Luego,

$$|z_w| \leq |w| - \lfloor \log(d(w)) \rfloor + 1.$$

Para ver que C verifica la condición (4.1.1), probaremos que C verifica las dos condiciones de la Nota 4.1.5.

i) Es claro que para todo $w, u \in \{0,1\}^*$, $|C(wu)| \geq |C(w)|$ dado que el intervalo $[h(wu), h(succ(wu)))$ esta incluido en el intervalo $[h(w), h(succ(w)))$.

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

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

Tenemos que

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

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

pero como $|z_w| \leq |w| - \lfloor \log d(w) \rfloor + 1$,

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

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

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

para todo k salvo quizá un número finito.

Finalmente, veamos que (C, D) comprime X. Para toda secuencia infinita $A \in X$,

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

□

El teorema principal (Teorema 4.2.4) es consecuencia inmediata de los Teorema 4.1.6 y 4.2.9. Notar que en la demostración del Teorema 4.1.6 no es necesario un decodificador, por lo que como corolario de estos resultados se tiene que si existe un codificador en tiempo polinómico que no empieza desde cero, entonces existe un codificador en tiempo polinómico que no empieza desde cero, que puede decodificarse en tiempo polinómico y con ratio de compresión igual o incluso más pequeño.

Corolario 4.2.10. Sea C un codificador en tiempo polinómico que no empieza desde cero. Entonces existe un compresor en tiempo polinómico (C', D') que no empieza desde cero tal que para toda secuencia infinita $A \in \mathbf{C}$

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

Hitchcock demostró en [34] que la dimensión en tiempo polinómico podía caracterizarse en términos de algoritmos de predicción on-line, usando el log-loss prediction ratio. Así pues, el resultado principal de este capítulo puede interpretarse como un puente entre el comportamiento de la predicción en tiempo polinómico y los algoritmos de compresión, tanto en el mejor como en el peor caso.

4.3. Aplicaciones de la caracterización

En esta sección se obtienen, como consecuencia de la caracterización vista en la sección anterior, resultados sobre la compresibilidad en tiempo polinómico de conjuntos completos y autoreducibles. Estos resultados son una consecuencia directa de aplicar la caracterización en resultados conocidos de dimensión en tiempo polinómico.

Notar que en esta sección se identifica, como en la mayor parte de la tesis, cada lenguaje $A \subseteq \{0,1\}^*$ con su secuencia característica χ_A, luego la compresibilidad de una clase siempre se referirá a la compresibilidad de sus secuencias características correspondientes.

El primer resultado muestra que no existe un compresor en tiempo polinómico que trabaje en todos los conjuntos completos bajo una reducción *many-one*.

Teorema 4.3.1. La clase de todos los conjuntos completos bajo una reducción *many-one* en tiempo polinómico para E es i.o. incompresible en tiempo polinómico.

Demostración. Ambos-Spies et al. demostraron [4] que esta clase tiene dimensión en tiempo polinómico 1.

□

Sea $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ la clase de conjuntos que son equivalentes a A mediante reducciones del tipo $\leq_{\mathrm{m}}^{\mathrm{P}}$. El ratio de compresión de $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ y $\deg_{\mathrm{m}}^{\mathrm{P}}(B)$, cuando $A \leq_{\mathrm{m}}^{\mathrm{P}} B$, se relaciona mediante el siguiente teorema.

Teorema 4.3.2. Sean A y B conjuntos en E de forma que $A \leq_{\mathrm{m}}^{\mathrm{P}} B$. Entonces,

1. El i.o. p-*compression ratio* de $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ es a lo mas el i.o. p-*compression ratio* de $\deg_{\mathrm{m}}^{\mathrm{P}}(B)$.

2. El a.e. p-*compression ratio* de $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ es a lo mas el a.e. p-*compression ratio* de $\deg_{\mathrm{m}}^{\mathrm{P}}(B)$.

Demostración. Ambos-Spies et al. demostraron 1. en [4] para dimensión polinómica . Athreya et al. demostraron en [8] el resultado para dimensión fuerte a partir del cual obtenemos 2. □

Para la siguiente aplicación se considerará la noción de autoreducibilidad.

Definición 4.3.3. Un conjunto A es autoreducible si A puede decidirse usando A como un oráculo pero sin permitir la pregunta x cuando la entrada sea x.

Los siguientes resultados de incompresibilidad se obtienen para el caso de autoreducibilidad *many-one* en tiempo polinómico y para el caso del complementario de los conjuntos i.o. p-Turing autoreducibles. Luego, para cada cota de tiempo polinómico existen conjuntos i.o. incompresibles que son $\leq_{\mathrm{m}}^{\mathrm{P}}$-autoreducibles y otros que ni siquiera son i.o. $\leq_{\mathrm{T}}^{\mathrm{P}}$-autoreducibles.

Teorema 4.3.4. La clase de los conjuntos autoreducibles para reducciones *many-one* en tiempo polinómico son i.o. incompresibles en tiempo polinómico.

Demostración. Ambos-Spies et al. demostraron en [4] que esta clase tiene dimensión en tiempo polinómico 1.

□

Teorema 4.3.5. La clase de conjuntos que no son i.o. polynomial-time Turing autoreducibles son i.o. incompresibles en tiempo polinómico.

Demostración. Beigel et al. demostraron en [10] que esta clase tiene dimensión en tiempo polinómico 1. □

El siguiente teorema muestra que existen *polynomial-time many-one degrees* con cada posible valor para compresibilidad, tanto a.e. como i.o.

Teorema 4.3.6. Sean x, y números reales calculables tales que $0 \leq x \leq y \leq 1$. Entonces existe un conjunto A en E tal que el ratio de i.o. compresión en tiempo polinómico de $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ es x y el ratio de a.e. compresión en tiempo polinómico de $\deg_{\mathrm{m}}^{\mathrm{P}}(A)$ es y.

Demostración. Athreya et al. demostraron en [8] el resultado para dimensión y dimensión fuerte en tiempo polinómico. □

Este último resultado incluye el caso extremo para el cual el ratio de i.o. compresión es 0 mientras que el ratio a.e. es 1.

Finalmente, la hipotesis "NP tiene dimensión positiva en tiempo polinómico" puede interpretarse en términos de incompresibilidad. Esta hipotesis tiene consecuencias interesantes para algoritmos de aproximación para MAX3SAT.

Teorema 4.3.7. Si para algún $\alpha > 0$, NP no es α-i.o-compresible en tiempo polinómico, entonces cualquier algoritmo de aproximación $\mathcal{A}$ para MAX3SAT debe satisfacer al menos una de las siguientes propiedades,

1. Para algún $\delta > 0$, $\mathcal{A}$ usa tiempo al menos $2^{n^{\delta}}$.
2. Para todo $\epsilon > 0$, $\mathcal{A}$ tiene un ratio de rendimiento menor que $7/8 + \epsilon$ (es decir, $\mathcal{A}(x) < (7/8 + \epsilon) \cdot \mathrm{MAX3SAT}(x)$ para todo x) en un conjunto exponencialmente denso de instancias satisfacibles.

Demostración. Hitchcock demostró en [32] que la consecuencias se siguen de considerar que NP tiene dimensión en tiempo polinómico positiva. □

Capítulo 5

Dimensión de Lempel-Ziv

En el Capítulo 4 se caracteriza la dimensión en tiempo polinómico mediante compresores que trabajan en tiempo polinómico y que verifican la condición (4.1.1). Dentro de este tipo de compresores se encuentra el algoritmo de compresión de Lempel-Ziv (ver Ejemplo 4.1.4).

El algoritmo de Lempel-Ziv (ver Subsección 1.3.8) fue definido por Lempel y Ziv en [96]. En dicho artículo se demostraba la universalidad del algoritmo frente a los algoritmos de compresión de estados finitos. Es decir, se probaba de un modo teórico que LZ_{78} tenía un ratio de compresión mejor que el de cualquier compresor de estados finitos. Este tipo de demostración teórica aseguraba, no sólo la conveniencia de usar el algoritmo de Lempel-Ziv frente a los algoritmos conocidos hasta el momento, sino frente a los algoritmos de compresión de estados finitos que pudieran surgir en un futuro.

Este ha sido el motivo del éxito que ha obtenido el compresor de Lempel-Ziv y por esto LZ es actualmente el método de compresión universal más utilizado en el mundo (por ejemplo, suele comprimir textos largos en inglés a la mitad de su tamaño original). No sólo eso, el algoritmo de Lempel-Ziv también se utiliza en los formatos de imágenes TIFF y GIF, y dentro del software de Adobe Acrobat, entre otros.

En este momento, Lempel-Ziv sigue siendo uno de los algoritmos más estudiados en el mundo y pese a ello, todavía quedan cuestiones abiertas sobre el comportamiento del algoritmo. En este capítulo se estudiará el ratio de compresión de Lempel-Ziv desde el punto de vista de la dimensión y se tratarán algunas de estas cuestiones, como por ejemplo, el problema abierto de la catástrofe del bit (Sección 5.2).

Los resultados de este capítulo se corresponden a los publicados en [59] con modificaciones.

5.1. La dimensión de Lempel-Ziv

En esta sección se desarrollará la dimensión de Lempel-Ziv y sus propiedades fundamentales.

Definición 5.1.1. Para cada $s \in [0, \infty)$, se define la *s-supergala de Lempel-Ziv* como la función $d_{\mathrm{LZ}}^{(s)}$ siguiente:

$$\begin{aligned} d_{\mathrm{LZ}}^{(s)}(\lambda) &= 1 \\ d_{\mathrm{LZ}}^{(s)}(w) &= \begin{cases} d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s|u|} \cdot \frac{\#\{i \in \{1 \dots n\} \mid u \sqsubseteq w_i\}}{n} & \text{si} \quad w = w_1 w_2 \dots w_n u \\ \\ d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s|w_{n+1}|} \cdot \frac{1}{2n} & \text{si} \quad w = w_1 w_2 \dots w_n w_{n+1} \ . \end{cases} \end{aligned}$$

donde, en el primer caso, $w_1, \dots w_n$ son las distintas frases en el análisis único válido de w y $u = w_i$ para algún $i \in \{1 \dots n\}$ y, en el segundo caso, w está formado por frases todas diferentes.

Lema 5.1.2. Sea $s \in [0, \infty)$, entonces $d_{\mathrm{LZ}}^{(s)}$ es una s-supergala.

Demostración. Para ver que es s-supergala tendremos que distinguir varios casos:

Caso 1. Sea $w = w_1 \dots w_n u$, donde $u0 \sqsubseteq w_i$ y $u1 \sqsubseteq w_j$ para $i, j \leq n$. En este caso,

$$\begin{aligned} [d_{\mathrm{LZ}}^{(s)}(w0) + d_{\mathrm{LZ}}^{(s)}(w1)] 2^{-s} &= d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s(|u|+1)} \left[\frac{\#\{i \in \{1 \dots n\} \mid u0 \sqsubseteq w_i\}}{n} \right. \\ &+ \left. \frac{\#\{i \in \{1 \dots n\} \mid u1 \sqsubseteq w_i\}}{n} \right] 2^{-s} \\ &\leq d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s|u|} \frac{\#\{i \in \{1 \dots n\} \mid u \sqsubseteq w_i\}}{n} \\ &= d_{\mathrm{LZ}}^{(s)}(w). \end{aligned}$$

Caso 2. Sea $w = w_1 \dots w_n u$, donde $u0 \sqsubseteq w_i$ para algún $i \leq n$ y $u1$ no es una secuencia que aparezca previamente en el distinct análisis ($u1 = w_{n+1}$). En este caso,

$$\begin{aligned} [d_{\mathrm{LZ}}^{(s)}(w0) + d_{\mathrm{LZ}}^{(s)}(w1)] 2^{-s} &= d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s(|u|+1)} [\frac{\#\{i \in \{1 \dots n\} \mid u0 \sqsubseteq w_i\}}{n} + \frac{1}{2n}] 2^{-s} \\ &\leq d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n) 2^{s|u|} \frac{\#\{i \in \{1 \dots n\} \mid u \sqsubseteq w_i\}}{n} \\ &= d_{\mathrm{LZ}}^{(s)}(w). \end{aligned}$$

Notar que es análogo el caso de cuando se intercambian 0 y 1.

Caso 3. Sea $w = w_1 \dots w_n u$, donde $u = w_i$ y $u1$, $u0$ no son secuencias que aparezcan previamente en el análisis único.

$$\begin{aligned}
[d_{\mathrm{LZ}}^{(s)}(w0) + d_{\mathrm{LZ}}^{(s)}(w1)]2^{-s} &= d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n)2^{s(|u|+1)}2\frac{1}{2n}2^{-s} \\
&\leq d_{\mathrm{LZ}}^{(s)}(w_1 \dots w_n)2^{s|u|}\frac{\#\{i \in \{1 \dots n\} \mid u \sqsubseteq w_i\}}{n} \\
&= d_{\mathrm{LZ}}^{(s)}(w).
\end{aligned}$$

□

Observación 5.1.3. Para cualquier real s calculable en tiempo polinómico, la s-supergala de Lempel-Ziv es calculable en tiempo polinómico.

Observación 5.1.4. Para todo $s, t \in [0, \infty)$ y $w \in \{0,1\}^*$,

$$d_{\mathrm{LZ}}^{(s)}(w)2^{-s|w|} = d_{\mathrm{LZ}}^{(t)}(w)2^{-t|w|}.$$

Definición 5.1.5. Sea una clase $X \subseteq \mathbf{C}$ y una secuencia infinita $A \in \mathbf{C}$.

1. La *dimensión de Lempel-Ziv de* X de define como,

$$\dim_{\mathrm{LZ}}(X) = \text{ínf}\{s \in [0, \infty) \mid X \subseteq S^\infty[d_{\mathrm{LZ}}^{(s)}]\}.$$

2. La *dimensión de Lempel-Ziv de* A se define como

$$\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(\{A\}).$$

Observación 5.1.6. Para todas las clases $X \subseteq Y \subseteq \mathbf{C}$, se tiene que

$$\dim_{\mathrm{LZ}}(X) \leq \dim_{\mathrm{LZ}}(Y).$$

El siguiente teorema establece que la dimensión de Lempel-Ziv de cualquier clase $X \subseteq \mathbf{C}$ se determina completamente a partir de la dimensión de las secuencias del conjunto (tal como ocurre con la dimensión constructiva y con cualquier dimensión definida a partir de una única martingala).

Teorema 5.1.7. Para todo $X \subseteq \mathbf{C}$,

$$\dim_{\mathrm{LZ}}(X) = \sup_{A \in X} \dim_{\mathrm{LZ}}(A).$$

Demostración. Sea $X \subseteq \mathbf{C}$ y sea $s = \sup_{A \in X} \dim_{\mathrm{LZ}}(A)$. Es claro por la Observación 5.1.6 que $\dim_{\mathrm{LZ}}(X) \geq s$.

Para ver que $\dim_{\mathrm{LZ}}(X) \leq s$, sea s' un número real tal que $s' > s$, será entonces suficiente demostrar que $\dim_{\mathrm{LZ}}(X) \leq s'$. Pero esto es trivial, puesto que al ser $s' > s > \sup_{A \in X} \dim_{\mathrm{LZ}}(A)$, significa que $d_{\mathrm{LZ}}^{(s')}$ tiene éxito en todas las $A \in X$ y por lo tanto tiene éxito en X. □

Este teorema implica una propiedad importante que debe cumplir cualquier dimensión, la *estabilidad contable.*

Corolario 5.1.8. 1. Para todos los conjuntos $X, Y \subseteq \mathbf{C}$,

$$\dim_{\mathrm{LZ}}(X \cup Y) = \max\{\dim_{\mathrm{LZ}}(X), \dim_{\mathrm{LZ}}(Y)\}.$$

2. Sean $X_1, X_2 \ldots \subseteq \mathbf{C}$,

$$\dim_{\mathrm{LZ}}(\bigcup_{i=1}^{\infty} X_i) = \sup_{i \in \mathbb{N}} \dim_{\mathrm{LZ}}(X_i).$$

El teorema principal de esta sección proporciona una caracterización exacta de la dimensión de Lempel-Ziv de una secuencia infinita. Esto se hace en términos de el ratio de compresión asintótico obtenido con el algoritmo de Lempel-Ziv en la secuencia.

Teorema 5.1.9. Sea $A \in \mathbf{C}$, entonces

$$\dim_{\mathrm{LZ}}(A) = \liminf_{n \to \infty} \frac{|\mathrm{LZ}(A[0 \ldots n-1])|}{n}.$$

Demostración. Para demostrar que

$$\dim_{\mathrm{LZ}}(A) \geq \liminf_{n \to \infty} \frac{|\mathrm{LZ}(A[0 \ldots n-1])|}{n},$$

sea $s \geq \dim_{\mathrm{LZ}}(A)$. Entonces, por la definición de dimensión de Lempel-Ziv,

$$\limsup_{n \to \infty} d_{\mathrm{LZ}}^{(s)}(A[0 \ldots n-1]) > 1$$

y por lo tanto existen infinitos n's $\in \mathbb{N}$ tales que $d_{\mathrm{LZ}}^{(s)}(A[0 \ldots n-1]) > 1$. Es decir, al ser

$$d_{\mathrm{LZ}}^{(s)}(w_1 \ldots w_n u) \leq \frac{2^{s|w_1 \ldots w_n u|}}{2^n n!},$$

tenemos que existen ininitos n's $\in \mathbb{N}$ tales que

$$\frac{2^{sn}}{2^{z(A,n)-1}(z(A,n)-1)!} > 1, \tag{5.1.1}$$

donde $z(A,n)$ denota el número de frases en el análisis único válido de $A[0\ldots n-1]$.

Por la desigualdad (5.1.1) existen infinitos n's $\in \mathbb{N}$ tales que

$$\begin{aligned} s &> \frac{z(A,n)-1+\log((z(A,n)-1)!)}{n} \\ &= \frac{z(A,n)-1+\sum_{k=1}^{z(A,n)-1}\log k}{n}. \end{aligned}$$

Luego, tomando límites tenemos que,

$$\begin{aligned} s > \quad & \liminf_n \frac{\sum_{k=1}^{z(A,n)-1}\log k + z(A,n) - 1}{n} = \\ & \liminf_n \frac{|\mathrm{LZ}(A[0\ldots n-1])| - \log z(A,n) - 1}{n}, \end{aligned}$$

como $z(A,n) \leq n$,

$$\dim_{\mathrm{LZ}}(A) \geq \liminf_n \frac{|\mathrm{LZ}(A[0\ldots n-1])|}{n}.$$

Para ver la otra desigualdad, sea $s > \liminf_n \frac{|\mathrm{LZ}(A[0\ldots n-1])|}{n}$. Esto significa que existen infinitas n's $\in \mathbb{N}$ tales que

$$s > \frac{\sum_{k=1}^{z(A,n)}\log k + z(A,n)}{n},$$

y por lo tanto,

$$\begin{aligned} 2^{sn} &> 2^{\sum_{k=1}^{z(A,n)}\log k + z(A,n)} \\ &= 2^{\log(z(A,n)!)+z(A,n)} \\ &= 2^{z(A,n)}z(A,n)!. \end{aligned}$$

Así que,

$$\begin{aligned} &\limsup_n d_{\mathrm{LZ}}^{(s)}(A[0\ldots n-1]) \geq \\ &\liminf_n d_{\mathrm{LZ}}^{(s)}(A[0\ldots n-1]) \geq \\ &\liminf_n \frac{2^{sn}}{2^{z(A,n)}z(A,n)!} > 1, \end{aligned}$$

y $s > \dim_{\mathrm{LZ}}(A)$, lo cual demuestra el teorema. □

La caracterización obtenida en el Teorema 5.1.9 nos permite encajar la dimensión de Lempel-Ziv entre la dimensión en tiempo polinómico y la dimensión de estados finitos. Esto se debe a las caracterizaciones de ambas dimensiones en términos de compresores.

Por un lado, y tal como se veía en el Capítulo 4 y en [61], dada una secuencia infinita $A \in \mathbf{C}$ y un compresor polinómico que no trabaja desde cero C,

$$\dim_{\mathrm{p}}(A) \leq \liminf_{n\to\infty} \frac{|C(A[0\ldots n-1])|}{n}.$$

Como se ve en el Ejemplo 4.1.4, Lempel-Ziv es un ejemplo de compresor polinómico que no trabaja desde cero, luego

$$\dim_{\mathrm{p}}(A) \leq \liminf_{n\to\infty} \frac{|LZ(A[0\ldots n-1])|}{n}. \tag{5.1.2}$$

Por otro lado, en [40] se demuestra que, la dimensión de estados finitos coincide con el mejor ratio de compresión asintótico que se puede obtener utilizando compresores de estados finitos sin perdida de información (ILFSC). Es decir, dada una secuencia infinita A,

$$\dim_{\mathrm{FS}} = \rho_{\mathrm{FS}}(A),$$

donde

$$\rho_{\mathrm{FS}}(A) = \inf_{\{C \text{ es ILFSC}\}} \liminf_{n\to\infty} \frac{|C(A[0\ldots n-1])|}{n}.$$

Debido a la universalidad del algoritmo de compresión de datos de Lempel-Ziv frente a los compresores de estados finitos [96], se tiene que

$$\liminf_{n\to\infty} \frac{|\mathrm{LZ}_{78}(A[0\ldots n-1])|}{n} \leq \rho_{\mathrm{FS}}(A). \tag{5.1.3}$$

El siguiente resultado es una consecuencia de [40, 61] y de la Observación 5.1.3. Por el Teorema 5.1.9, la segunda parte es una reformulación de las desigualdades (5.1.2) y (5.1.3) en términos de dimensión.

Teorema 5.1.10. Sea $X \subseteq \mathbf{C}$, entonces

$$\dim_{\mathrm{p}}(X) \leq \dim_{\mathrm{LZ}}(X) \leq \dim_{\mathrm{FS}}(X).$$

En particular, para toda secuencia infinita $A \in \mathbf{C}$, se tiene que

$$\dim_{\mathrm{p}}(A) \leq \dim_{\mathrm{LZ}}(A) \leq \dim_{\mathrm{FS}}(A).$$

5.2. La catástrofe del bit

Una de las cuestiones que todavía siguen abiertas en torno al algoritmo de compresión de datos de Lempel-Ziv es si cumple la denominada catástrofe del bit.

Esta cuestión fue planteada inicialmente por Lutz y recogida por diversos autores en [44, 52]. Básicamente, lo que se denomina la catástrofe del bit es que el ratio de compresión de una secuencia infinita pueda cambiar sustancialmente cuando se añade un bit al inicio de la secuencia.

Es decir, en términos de dimensión, lo anterior se enuncia del siguiente modo: Una secuencia infinita $A \in \mathbf{C}$ cumple la catástrofe del bit si y sólo si para algún $b \in \{0,1\}$

$$\dim_{\mathrm{LZ}}(A) \neq \dim_{\mathrm{LZ}}(bA). \tag{5.2.1}$$

Intuitivamente, parece que la catástrofe del bit es algo que no debería cumplirse. El que un único bit pueda modificar la compresión de una secuencia con infinitos caracteres sería una debilidad del algoritmo de Lempel-Ziv.

Sin embargo, que la catástrofe del bit no se cumpla, significa algo más que el que un sólo bit no cambie la compresión. En efecto, si no se cumple la catástrofe del bit, se tendría (en términos de dimensión) que para toda secuencia infinita $A \in \mathbf{C}$ y todo bit $b \in \{0,1\}$,

$$\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(bA).$$

Reiterando esto, que no se cumpla la catástrofe del bit significa que, para toda secuencia finita $w \in \{0,1\}^*$ y toda secuencia infinita $A \in C$,

$$\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(wA). \tag{5.2.2}$$

Es decir, que el ratio de compresión de una secuencia infinita no varíe aunque se añada al inicio de la misma una secuencia finita, tan grande como se desee. Y esto parece una propiedad demasiado buena para un único algoritmo de compresión. El siguiente ejemplo sirve para ver hasta que punto la intuición o los experimentos pueden resultar engañosos ante la cuestión de si se cumple o no la catástrofe del bit.

Ejemplo 5.2.1. Sea $A = 101100111000\ldots1^n0^n\ldots \in \mathbf{C}$ que claramente se comprime mucho utilizando el algoritmo de Lempel-Ziv. Comparemos experimentalmente como comprime Lempel-Ziv la secuencia A y la secuencia $1A$ sobre prefijos cada vez más largos.

Sea $t(w)$ el número de frases en el análisis válido único de un $w \in \{0,1\}^*$. Sea $w_1 = A[0\ldots29]$, $w_2 = A[0\ldots109]$ y $w_3 = A[0\ldots239]$. Entonces,

$$\begin{array}{llll} t(w_1) = 10 & \Rightarrow & |\mathrm{LZ}(w_1)| = 35 & \qquad t(1w_1) = 13 \Rightarrow |\mathrm{LZ}(1w_1)| = 53 \\ t(w_2) = 20 & \Rightarrow & |\mathrm{LZ}(w_2)| = 101 & \qquad t(1w_2) = 29 \Rightarrow |\mathrm{LZ}(1w_2)| = 146 \\ t(w_3) = 30 & \Rightarrow & |\mathrm{LZ}(w_3)| = 151 & \qquad t(1w_3) = 45 \Rightarrow |\mathrm{LZ}(1w_3)| = 271 \end{array}$$

Usando estos pocos prefijos, en el caso de la secuencia A parece que, cuanto más largos son los prefijos, mejor comprime el algoritmo de Lempel-Ziv. Sin embargo, en el caso de los prefijos de la secuencia $1A$, parece que el algoritmo de Lempel-Ziv no comprime nada en absoluto. Así que, experimentalmente parecería que la secuencia A cumple la catástrofe del bit.

Mediante resultados que se verán en la Sección 5.3, se demostrará que ambas secuencias se comprimen igual de bien asintóticamente y por lo tanto, la secuencia A no cumple la catástrofe del bit, pese a lo que los resultados experimentales parecían indicar.

La siguiente proposición establece que las dimensiones en tiempo polinómico y de estados finitos de una secuencia infinita no varían al añadir una secuencia finita en su inicio. Este es un resultado en la línea de la ecuación (5.2.2), que nos permitirá establecer como consecuencia directa una condición para que no se cumpla la catástrofe del bit.

Proposición 5.2.2. Para toda secuencia infinita $A \in \mathbf{C}$ y toda secuencia finita $w \in \{0,1\}^*$, se tiene que

1. $\dim_{\mathrm{p}}(wA) = \dim_{\mathrm{p}}(A)$.
2. $\dim_{\mathrm{FS}}(wA) = \dim_{\mathrm{FS}}(A)$.

Teorema 5.2.3. Sea una secuencia infinita $A \in \mathbf{C}$ y una secuencia finita $w \in \{0,1\}^*$. Entonces

$$|\dim_{\mathrm{LZ}}(A) - \dim_{\mathrm{LZ}}(wA)| \leq \dim_{\mathrm{FS}}(A) - \dim_{\mathrm{p}}(A).$$

En particular, si $\dim_{\mathrm{FS}}(A) = \dim_{\mathrm{p}}(A)$, entonces para todo $w \in \{0,1\}^*$,

$$\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(wA),$$

y no se cumple la catástrofe del bit en la secuencia A.

Demostración. Sea $A \in \mathbf{C}$ y $w \in \{0,1\}^*$.

i) Si $\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(wA)$ es obvio.

ii) Si $\dim_{\mathrm{LZ}}(A) > \dim_{\mathrm{LZ}}(wA)$ entonces por el Teorema 5.1.10 y la Proposición 5.2.2 se tiene que,

$$\begin{aligned} \dim_{\mathrm{LZ}}(A) - \dim_{\mathrm{LZ}}(wA) &\leq \dim_{\mathrm{FS}}(A) - \dim_{\mathrm{p}}(wA) \\ &= \dim_{\mathrm{FS}}(A) - \dim_{\mathrm{p}}(A). \end{aligned}$$

iii) Si $\dim_{\mathrm{LZ}}(wA) > \dim_{\mathrm{LZ}}(A)$ de un modo similar se tiene que,

$$\begin{aligned}\dim_{\mathrm{LZ}}(A) - \dim_{\mathrm{LZ}}(wA) &\leq \dim_{\mathrm{FS}}(wA) - \dim_{\mathrm{p}}(A) \\ &= \dim_{\mathrm{FS}}(A) - \dim_{\mathrm{p}}(A).\end{aligned}$$

□

Definición 5.2.4. Sea la clase LZbit el conjunto de todas las secuencias infinitas S que no cunplen la catástrofe del bit, es decir, aquellas secuencias infinitas A's para las cuales, $\forall w \in \{0,1\}^*$,

$$\dim_{\mathrm{LZ}}(A) = \dim_{\mathrm{LZ}}(wA).$$

5.2.1. Algunos resultados sobre la catástrofe del bit

Es una técnica habitual, a la hora de estudiar el análisis valido único de una secuencia w, utilizar un árbol G_w de grado 2. Este árbol cumple la propiedad de que cada frase en el análisis único valido es un nodo del árbol. Las aristas de G_w se construyen del siguiente modo: si $w_i = w_j b$, se dibuja una arista en G_w uniendo el nodo que representa w_i con el nodo que representa w_j. Veamos un ejemplo,

Ejemplo 5.2.5. Sea la secuencia del Ejemplo 1.3.21, es decir, $w = 0100110001001$ cuyo análisis válido único es

$$\begin{array}{ccccccc} 0 & 1 & 00 & 11 & 000 & 10 & 01 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{array}$$

Entonces, el árbol asociado al análisis de w es el árbol de la figura 5.1.

Sea $A \in \mathbf{C}$ una secuencia infinita y n número natural. Sea G_n el árbol asociado al análisis válido único de $A[0 \ldots n-1]$ y $m(n)$ la altura de G_n. Sea n_k el número de nodos de G_n en el nivel k. Entonces, el número de frases del análisis válido único de $A[0 \ldots n-1]$ es $\sum_{k=1}^{m(n)} n_k$ o bien $\sum_{k=1}^{m(n)} n_k + 1$ (este último caso es cuando una frase del análisis válido único aparece repetida). Notar que $\forall w \in \{0,1\}^*$, $|\mathrm{LZ}_{78}(w)| \leq t(w) \log t(w)$. Por lo tanto, para calcular el ratio de compresión de Lempel-Ziv 78 (es decir, la dimensión de Lempel-Ziv de una secuencia infinita) podemos suponer, sin perdida de generalidad, que $t(A[0 \ldots n-1) = \sum_{k=1}^{m(n)} n_k$ y por lo tanto que,

$$\dim_{\mathrm{LZ}}(A) \leq \liminf_n \frac{(\sum_{k=1}^{m(n)} n_k)(\log \sum_{k=1}^{m(n)} n_k)}{n}.$$

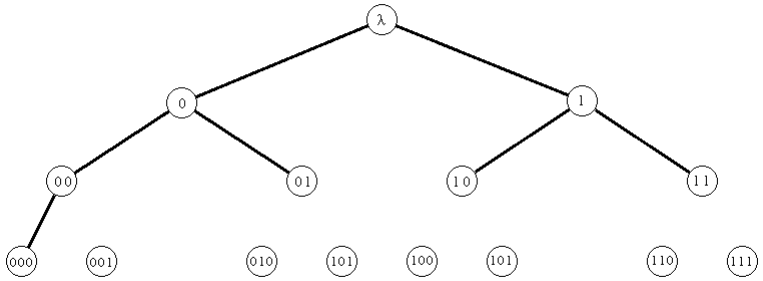

Figura 5.1: Árbol asociado al análisis válido único de $w = 0100110001001$.

Además, se tiene que la longitud de la secuencia es exactamente la suma de longitudes de cada una de sus frases, luego $n = \sum_{k=1}^{m(n)} k \cdot n_k$ y por lo tanto

$$\dim_{\mathrm{LZ}}(A) \leq \liminf_n \frac{(\sum_{k=1}^{m(n)} n_k)(\log \sum_{k=1}^{m(n)} n_k)}{\sum_{k=1}^{m(n)} k \cdot n_k}. \tag{5.2.3}$$

Veamos cómo comprime Lempel-Ziv en una secuencia infinita A según los árboles que generen los análisis válidos únicos.

Caso 1. Suponer que $\forall k$ se tiene que $n_k = C$ constante. En ese caso, utilizando la ecuación (5.2.3) se tiene que

$$\dim_{\mathrm{LZ}}(A) \leq \liminf_n \frac{2(C \cdot m(n)) \log(C \cdot m(n))]}{Cm(n)(m(n)-1)} = 0,$$

y por lo tanto, el algoritmo de Lempel Ziv 78 comprime mucho. La interpretación de que $n_k = C$ en cuanto al análisis de Lempel-Ziv nos dice que existen un número constante C de frases de cada longitud. Un ejemplo de secuencia de este tipo sería:

$$A = 01001100011100001111\cdots$$

y también cualquier otra que pueda parecer algo más aleatoria como

$$A = 00011000100100010010000100010011 0011\cdots$$

En ambos ejemplos $C = 2$, pero con cualquier otra constante se cumple también, de modo que se pueden generar secuencias que puedan parecer relativamente complicadas y sin embargo muy compresibles por Lempel-Ziv 78.

Caso 2. Suponer que $\forall k$ se cumple que $n_k = C \cdot k$. En ese caso, utilizando la ecuación (5.2.3) se tiene que

$$\dim_{\mathrm{LZ}}(A) \quad \leq \quad \liminf_n \frac{\frac{C}{2}m(n)(m(n)+1)\log(\frac{C}{2}m(n)(m(n)+1)}{\frac{c}{6}m(n)(m(n)+1)(2m(n)+1)} = 0,$$

y por lo tanto, el algoritmo de Lempel Ziv 78 comprime mucho.

Caso 3. Suponer que para todo nivel k suficientemente grande, $n_k = 2^k$. Entonces utilizando la ecuación (5.2.3) se tiene que

$$\dim_{\mathrm{LZ}}(A) = \liminf_n \frac{(2^{m(n)+1} - 2)\log(2^{m(n)+1} - 2)}{(m(n) - 1)2^{m(n)+1} + 2} = 1,$$

y por lo tanto Lempel Ziv 78 no comprime nada este tipo de secuencias. La interpretación de que $n_k = 2^k$ es que todas las secuencias aparecen como frases en el análisis de Lempel-Ziv. Un ejemplo de estas secuencias son las secuencias de Coperland-Erdös, por ejemplo la secuencia:

$$A = 0100011011000001010011100101110111\cdots$$

Caso 4. Suponer que para todo nivel k suficientemente grande, $n_k = C \cdot 2^k$ con C constante. Entonces utilizando la ecuación (5.2.3) se tiene que

$$\dim_{\mathrm{LZ}}(A) = \liminf_n \frac{C \cdot (2^{m(n)+1} - 2)\log(C \cdot (2^{m(n)+1} - 2))}{C \cdot [(m(n) - 1)2^{m(n)+1} + 2]} = 1,$$

y por lo tanto Lempel Ziv 78 no comprime nada este tipo de secuencias.

5.3. Secuencias altamente compresibles e incompresibles por LZ

En esta sección se utilizará la caracterización de la dimensión de Lempel-Ziv en términos del ratio de compresión para encontrar familias de secuencias que son altamente compresibles por LZ y que además no cumplen la catástrofe del bit. También se encontrarán ejemplos concretos de secuencias que son incompresibles por LZ_{78} y que tampoco cumplen la catástrofe del bit.

Definición 5.3.1. Sea secuencia infinita $S \in \mathbf{C}$,

1. Se dice que S es altamente compresible por LZ si

$$\liminf_{n\to\infty} \frac{|\mathrm{LZ}_{78}(S[0\ldots n-1])|}{n} = 0$$

Es decir, en términos de la dimensión de Lempel-Ziv, $\dim_{\mathrm{LZ}}(S) = 0$.

2. Se dice que S es incompresible por LZ si

$$\liminf_{n\to\infty} \frac{|\mathrm{LZ}_{78}(S[0\ldots n-1])|}{n} = 1$$

Es decir, en términos de la dimensión de Lempel-Ziv, $\dim_{\mathrm{LZ}}(S) = 1$.

El siguiente resultado es una consecuencia inmediata del Teorema 5.1.10.

Proposición 5.3.2. Sea una secuencia infinita $S \in \mathbf{C}$,

i) Si $\dim_{\mathrm{FS}}(S) = 0$, entonces S es altamente compresible por LZ y S no cumple la catástrofe del bit.

ii) Si $\dim_{\mathrm{p}}(S) = 1$, entonces S es incompresible por LZ y S no cumple la catástrofe del bit. Notemos que este segundo caso tiene la mayoría de las secuencias (LZbit tiene p-medida 1).

Veamos algunos ejemplos concretos en los que aplicar este último resultado.

Definición 5.3.3. Sea una secuencia infinita $S \in \mathbf{C}$ y $m \in \mathbb{Z}^+$. Entonces,

1. El *factor set* $F_m(S)$ el el conjunto de todas las secuencias finitas de longitud m que aparecen en S, es decir
$$F_m(S) = \{w \in \{0,1\}^m \mid w \text{ aparece en } S\}.$$
2. La *factor complexity function*, $p_S : \mathbb{N} \to \mathbb{N}$, se define como el número de elementos en el factor set para cada m, es decir $p_S(m) = |F_m(S)|$.

En [40] se prueba que la dimensión de estados finitos de secuencias con $\mathrm{p}_S(m) = 2^{o(m)}$ es igual a cero. Así pues, tenemos el siguiente resultado.

Corolario 5.3.4. Cada secuencia infinita $S \in \mathbf{C}$ con $\mathrm{p}_S(m) = 2^{o(m)}$ es altamente compresible por LZ y además $S \in$ LZbit.

En particular, usando este resultado en el Ejemplo 5.2.1 visto en la seccion anterior, la secuencia S es altamente compresible y $S \in$ LZbit (no cumple la catástrofe del bit) puesto que S satisface $\mathrm{p}_S(m) = m(m+1)$.

Otras aplicaciones del Corolario 5.3.4 son las siguientes

Corolario 5.3.5.
1. Si S es la expansión binaria de un número racional, S es altamente compresible por LZ_{78} y $S \in$ LZbit.
2. Las secuencias de Sturmians, las secuencias Morphic y las secuencias Automatic son altamente compresibles por LZ_{78} y están en LZbit (Ver [40]).
3. Cada $S \in$ REG es altamente compresible por LZ_{78} y $S \in$ LZbit.

Todos los ejemplos vistos hasta este momento son secuencias altamente compresibles por compresores de estados finitos y por lo tanto no cumplen la catastrofe del bit. Sin embargo, existen secuencias que son altamente compresibles por el algoritmo de Lempel Ziv pero no lo son para ningún compresor de estados finitos. Un ejemplo es la secuencia $S = s_0 s_1 s_2 s_3 \ldots$, es decir la secuencia formada por la concatenación de todas las secuencias de $\{0,1\}^*$ ordenadas en orden lexicográfico (ver [96]). Para este tipo de secuencias, no se conoce si la catástrofe del bit se cumple o no.

Capítulo 6

Dimensión y teoría del aprendizaje computacional

La teoría de aprendizaje computacional estudia desde un punto de vista formal el rendimiento y los recursos necesarios en aprendizaje automático (*machine learning*). Esta formalización se remonta a 1984 cuando Valiant introdujo el modelo de aprendizaje aproximadamente correcto (PAC learning) [94]. Este modelo ha sido desde entonces ampliamente estudiado y de él han surgido diversos modelos alternativos como el modelo de aprendizaje basado en preguntas de Angluin [6] o el modelo de aprendizaje on-line (*on-line mistake-bound learning model*) de Littlestone [58]. Los principales problemas abiertos en aprendizaje computacional están:

i) Relacionados con los límites de cada modelo de aprendizaje. En efecto, resulta especialmente interesante conocer que una cierta clase de conceptos no se puede aprender bajo determinado modelo y también resulta interesante el establecer una cota inferior de la complejidad de aprendizaje inherente a cada clase.

ii) Relacionados con la busqueda de nuevos algoritmos de aprendizaje más eficientes.

Este capítulo se centra en la relación entre la teoria de aprendizaje computacional y la dimensión efectiva. Por un lado, se tiene como objetivo obtener resultados de no aprendizaje a partir de resultados de dimensión. Por otro lado, se traducen algoritmos de aprendizaje en demostraciones de resultados en dimensión efectiva.

El principal antecedente de este capítulo es el trabajo de Watanabe y otros autores donde se investiga la medida de recursos acotados de clases que son aprendibles con un algoritmo PAC o

con un algoritmo basado en preguntas de equivalencia [57]. Mas concretamente, se prueba en su trabajo que: *i*) Las subclases de P/poly que pueden aprenderse con un algoritmo PAC tiene medida polinómica 0 si EXP $\not\subseteq$ AM; y *ii*) las subclases de P/poly que pueden aprenderse con preguntas de equivalencia tienen medida polinómica 0. De estos resultados, se obtienen hipótesis en medidas de recursos acotados que implicarían el no aprendizaje de la clase de circuitos Booleanos en tiempo polinómico. Por otro lado, en el contexto de dimensión efectiva, Hitchcock exploró en [34] la relación de dimensión con la logarithmic loss unpredictability.

Este capítulo proporciona nuevos resultados en línea con los citados anteriormente. Primero, en relación con el aprendizaje on-line, se obtiene una cota superior de la dimensión en tiempo polinómico de clases de conceptos que pueden aprenderse con algoritmos on-line en tiempo exponencial y con $\alpha 2^n$ errores. Es más, se demuestra que esta cota superior es óptima (en el sentido de que no se puede mejorar). Basándose en los resultados obtenidos en este capítulo, Hitchcock ha investigado en [37] el caso de tener un número de errores subexponencial y dimensión cero, con interesantes aplicaciones que desarrolla en [27].

En segundo lugar, en relación con el aprendizaje PAC, se demuestra en este capítulo que la dimensión en espacio polinómico de clases de conceptos que son PAC-aprendibles es cero. Esto proporciona una hipótesis basada en dimensión efectiva que implica la impredicibilidad inherente de una clase de conceptos (la no predictivilidad es una propiedad muy interesante en aprendizaje computacional que hace referencia a las clases que usando cualquier hipótesis no son PAC aprendibles). Más aún, existen conexiones entre resultados de aprendizaje PAC y construcciones en el campo de la criptografía [48] que se pueden reescribir con hipótesis de dimensión efectiva.

Finalmente, se estudia la dimensión de clases que se pueden aprender con algoritmos basados en preguntas de pertenencia. El principal resultado demuestra que la dimensión en espacio polinómico de clases de conceptos que pueden aprenderse con un algoritmo basado en preguntas de pertenencia es cero. Esto puede usarse para demostrar que, para clases que son complejas en el sentido de dimensión, es necesaria una representación compleja para poder aprender eficientemente esa clase con preguntas de pertenencia. Las definiciones formales de los modelos de aprendizajes de este capítulo se encuentran en la Sección 1.3.9. Los resultados de este capítulo han sido publicados junto con Ricard Gavalda, Elvira Mayordomo y Vinodchandran N. Variyam en [25].

6.1. Dimensión y aprendizaje on-line

Esta sección se centra en el modelo de aprendizaje on-line. En particular, se demuestra que la cota de error en el aprendizaje on-line proporciona una cota superior en la dimensión en tiempo polinómico y, en algunos casos, esta cota es muy ajustada.

La relación entre dimensión con *logarithmic loss unpredictability* fue explorada en [34]. Esta relación es intuitivamente muy cercana a la relación entre dimensión y aprendizajo on-line cuando restringimos que los ejemplos vengan dados en orden lexicográfico.

Basasándose en los resultados de esta sección (aunque sus publicaciones fueron previas) Hitchcock [37] exploró el caso de dimensión cero y cotas de error pequeñas para aprendizaje on-line, incluyendo reducciónes a esas clases.

El principal teorema de esta sección proporciona una cota superior para la dimensión en tiempo polinómico de clases de conceptos que pueden aprenderse on-line con $\alpha 2^n$ errores.

Teorema 6.1.1. Sea $\alpha \leq 1/2$ un número p-calculable. Sea $\mathcal{C}$ una clase de conceptos que se puede aprender con un algoritmo on-line que comete a lo más $\alpha 2^n$ errores, entonces

$$\dim_{\mathrm{p}}(\mathcal{C}) \leq \mathcal{H}(\alpha),$$

donde $\mathcal{H}$ es la entropía binaria de Shannon definida en el Capítulo 1 (Definición 1.3.13).

Demostración. Sea $\alpha < 1/2$ (el caso $\alpha = 1/2$ es trivial). Veamos que para todo $s > \mathcal{H}(\alpha)$, existe una s-gala que tiene éxito en $\mathcal{C}$.

Sea $\epsilon = \frac{s-\mathcal{H}(\alpha)}{2}$ y sea la función $h_\alpha(x) = \alpha \log \frac{1}{x} + (1-\alpha)\log \frac{1}{1-x}$. Esta función es continua en $(0,1)$ y su valor mínimo es $\mathcal{H}(\alpha)$, que se obtiene cuando $x = \alpha$. Sea δ tal que $h_\alpha(\alpha+\delta) \leq \mathcal{H}(\alpha) + \epsilon$, y $\alpha + \delta < 1/2$.

Sea A un algoritmo on-line que aprende $\mathcal{C}$ con $\alpha 2^n$ errores. Para cada $z \in \{0,1\}^*$ de longitud entre 0 y 2^n denotamos con $h(z)$ la historia que corresponde a haber recibido los ejemplos ordenados $s^n_0 \dots s^n_{|z|-1}$ con sus correspondientes respuestas correctas $z[0] \dots z[|z|-1]$. Es decir, los ejemplos están en orden lexicográfico y las respuestas guardadas en los bits de z.

Definimos la siguiente s-gala $d : \{0,1\}^* \to [0,\infty)$ recursivamente como sigue:

i) $d(\lambda) = 1$.

ii) Para cada $n \in \mathbb{N}$ y cada w con $2^n - 1 \leq |w| < 2^{n+1} - 1$,

$$d(wb) = \begin{cases} (\alpha+\delta)2^s d(w) & \text{si } A(n, h(w[2^n-1 \dots |w|-1]), s_{|w|}) = \bar{b}, \\ (1-(\alpha+\delta))2^s d(w) & \text{si } A(n, h(w[2^n-1 \dots |w|-1]), s_{|w|}) = b. \end{cases}$$

Notar que d puede calcularse en tiempo polinómico puesto que A se ejecuta en tiempo polinómico en la longitud de la entrada (y esta incluye la historia).

Sea $L \in \mathcal{C}$ un concepto. Entonces,

$$\begin{aligned}
d(L[0\ldots 2^{n+1}-2]) &= d(L^{=0}\ldots L^{=n}) \\
&\geq \left[(\alpha+\delta)^{\mathrm{Mist}(n,L^{=n},A)}(1-(\alpha+\delta))^{2^n-\mathrm{Mist}(n,L^{=n},A)}\right] 2^{2^n s} d(L^{=0}\ldots L^{=n-1}) \\
&\geq \left[(\alpha+\delta)^{\alpha 2^n}(1-(\alpha+\delta))^{(1-\alpha)2^n}\right] 2^{2^n s} d(L^{=0}\ldots L^{=n-1}) \\
&= 2^{-h_\alpha(\alpha+\delta)2^n} 2^{2^n s} d(L^{=0}\ldots L^{=n-1}) \\
&= 2^{2^n(s-h_\alpha(\alpha+\delta))} d(L^{=0}\ldots L^{=n-1}) \\
&\geq 2^{2^n(s-\mathcal{H}(\alpha)-\epsilon)} d(L^{=0}\ldots L^{=n-1}) \\
&\geq 2^{\sum_{i=0}^{n} 2^i(s-\mathcal{H}(\alpha)-\epsilon)} = 2^{(2^{n+1}-1)\epsilon}
\end{aligned}$$

que tiende a infinito con n. Así pues $\mathcal{C} \subseteq S^\infty[d]$ y $\dim_{\mathrm{p}}(\mathcal{C}) \leq s$. □

Como corolario se obtiene que las clases de conceptos que se pueden aprender con $o(2^n)$ errores tienen p-dimensión 0. Este corolario fue generalizado posteriormente por Hitchcok en [37].

Corolario 6.1.2. Sea $\mathcal{C}$ una clase de conceptos que se puede aprender con un algoritmo on-line que comete $o(2^n)$ errores. Entonces,

$$\dim_{\mathrm{p}}(\mathcal{C}) = 0.$$

El siguiente corolario es una consecuencia inmediata de la Proposición 1.3.34 y mejora un resultado obtenido por Lindner, Schuler y Watanabe [57].

Corolario 6.1.3. Si los circuitos Booleanos se pueden aprender en tiempo polinómico (incluso en tiempo exponencial lineal) con $o(2^n)$ preguntas de equivalencia, entonces la clase de conceptos de circuitos Booleanos tiene p-dimensión 0.

A continuación se demuestra que el Teorema 6.1.1 es óptimo, en el sentido de que podemos encontrar una clase de conceptos que se puede aprender con $\alpha 2^n$ errores y que tiene p-dimensión $\mathcal{H}(\alpha)$.

Teorema 6.1.4. Sea $\alpha \leq 1/2$ un número p-calculable. Existe una clase de conceptos $\mathcal{C}_\alpha$ que es on-line aprendible con $\alpha 2^n$ errores tal que $\dim_{\mathrm{p}}(\mathcal{C}_\alpha) = \mathcal{H}(\alpha)$.

Demostración. Para cada α consideramos la clase de conceptos

$$\mathcal{C}_\alpha = \{L \in \mathbf{C} \mid \forall n, \#L^{=n} \leq \alpha 2^n\}$$

y en la línea de la demostración del Lema 5,1. en [67], se puede ver que $\dim_{\mathrm{p}}(\mathcal{C}_\alpha) = \mathcal{H}(\alpha)$. Considerar el algoritmo A que predice 0 todo el tiempo. El número de errores que hace este algoritmo para cualquier concepto de $\mathcal{C}_\alpha$ es como máximo $\alpha 2^n$. □

El último resultado de esta sección demuestra que los valores de la dimensión de las clases on-line aprendibles con $\alpha 2^n$ errores son densos en el intervalo $[0, \mathcal{H}(\alpha)]$.

Teorema 6.1.5. Sea $\alpha \leq 1/2$ un número p-calculable y sea $\beta \in [0, \mathcal{H}(\alpha))$ p-calculable. Entonces, existe una clase de conceptos $\mathcal{C}_\beta$ que es on-line aprendible con $\alpha 2^n$ errores tal que

$$\dim_{\mathrm{p}}(\mathcal{C}_\beta) = \beta.$$

Demostración. Sea $\beta > 0$ y sea $\mathcal{C}_\beta = \{L \in \mathbf{C} \,|\, \forall n, \#L^{=n} \leq \gamma 2^n\}$, donde γ es el menor valor tal que $\mathcal{H}(\gamma) = \beta$. Notar que, $\mathcal{H}(x)$ es una función continua, estrictamente creciente y simétrica para $x \leq 1/2$, así que $\gamma \leq \alpha$. Por el Teorema 6.1.4, $\dim_{\mathrm{p}}(\mathcal{C}_\beta) = \beta$ y $\mathcal{C}_\beta$ es on-line aprendible con $\gamma 2^n$ errores, así que también es on-line aprendible con $\alpha 2^n$ errores.

El caso $\beta = 0$ se cumple trivialmente con la misma definición de $\mathcal{C}_\beta$. □

Para finalizar, notar que todos los resultados de esta sección se pueden generalizar usando un modelo on-line que sólo tenga que aprender si los ejemplos vienen dados en orden lexicográfico.

6.2. Dimensión y Aprendizaje PAC

Esta sección se centrará en el aprendizaje PAC. Este modelo de aprendizaje se puede relacionar con la medida de recursos acotados [57] tal como mostraron Watanabe et al. Mostraremos en esta sección que también está relacionado con la dimensión pspace, generalizando parcialmente los resultados de [57].

El resultado principal de esta sección demuestra que la dimensión pspace de una clase de conceptos que es aprendible mediante un algoritmo PAC es cero. Este resultado puede usarse para demostrar que una clase $\mathcal{C}$ "grande" (en sentido de dimensión) no es aprendible PAC independientemente de la clase hipótesis $\mathcal{H}$ que se use (en términos de [48], $\mathcal{C}$ es inherentemente impredecible).

Finalmente se verán también resultados similares para dimensiones con plogon y p_2 como cotas de recursos, aunque en este caso se necesitará alguna hipótesis extra.

Notar que, como las cotas del aprendizaje PAC (tiempo y espacio) dependen del tamaño de la representación, únicamente las clases con representaciones subexponenciales ($\mathbf{size}_n(c) \in o(2^n)\ \forall c$) tienen un interés real.

Teorema 6.2.1. Sea $\mathcal{C}$ una clase de conceptos aprendible PAC con representaciones subexponenciales. Entonces

$$\dim_{\text{pspace}}(\mathcal{C}) = 0.$$

Además, si existe un algoritmo PAC que se ejecute en espacio $O(2^n)$ con un número de ejemplos $\xi(n)$ verificando $\sum_{i=0}^{n} \xi(i) \in o(2^n)$, entonces

$$\dim_{\text{pspace}}(\mathcal{C}) = 0.$$

Demostración. El primer resultado es un caso particular del segundo, así que probaremos este último.

Sea A el algoritmo PAC que nos asegura que $\mathcal{C}$ es PAC-aprendible. Sea D la distribución uniforme. Sea $s > \mathcal{H}(\epsilon)$ y sea $c \in \mathcal{C}_n$, entonces, el algoritmo A en la entrada (n, ϵ, δ) devuelve (con probabilidad $1-\delta$) una hipótesis h tal que h ϵ-aproxima c. Así pues, con probabilidad $1-\delta$,

$$\frac{\#\{x \in \{0,1\}^n | h(x) \neq c(x)\}}{2^n} = \text{err}_D(c,h) \leq \epsilon.$$

Sea Q_n la clase de todos los conjuntos de ejemplos posibles que $A(n, \epsilon, \delta)$ podría usar, es decir

$$Q_n = \{Q \subseteq \{0,1\}^n \,|\, \#Q \leq \xi(n)\}.$$

Ahora, sea $Q \in Q_n$ y w de longitud 2^n (es decir, la representación lexicográfica de un concepto $c \in \mathcal{C}_n$). Decimos que w *es buena para* A *con respecto* Q si

$$A^{c,Q}(n, \epsilon, \delta) = h \text{ con } \text{err}_D(c,h) \leq \epsilon,$$

donde la notación $A^{c,Q}$ hace referencia a la salida de A cuando a A se le proporciona como ejemplos los elementos de Q junto con la información de si estos están o no en c.

Intuitivamente, w es buena para A con respecto Q si podemos aprender aproximadamente el concepto c representado por w proporcionando a A los ejemplos de Q.

Sea $B_{n,Q}$ el conjunto de secuencias de longitud 2^n que son buenas para A con respecto Q y sea $d_{n,Q} : \{0,1\}^{\leq 2^n} \to [0, \infty)$ la función definida como sigue,

$$d_{n,Q}(v) = \frac{\#\{w \text{ buena para } A \text{ con respecto } Q \,|\, v \sqsubseteq w\}}{\#B_{n,Q}}.$$

Notar que, reutilizando espacio, $d_{n,Q}$ es calculable en espacio $O(2^n)$. A continuación definiremos la función d_n considerando todos los $Q \in Q_n$,

$$d_n(v) = \frac{\sum_{Q \in Q_n} d_{n,Q}(v)}{\#Q_n}.$$

Notar que d_n también es calculable en espacio $O(2^n)$. Además,

i) $d_n(\lambda) = 1$.

ii) d_n verifica $d_n(w0) + d_n(w1) = d_n(w)$ para todo $|w| < 2^n$.

iii) Si w es una secuencia de longitud 2^n entonces

$$d_n(w) = \sum_{Q \in Q_n(w)} \frac{1}{\#Q_n \cdot \#B_{n,Q}}, \tag{6.2.1}$$

donde $Q_n(w)$ está definida como

$$Q_n(w) = \{Q \in Q_n \mid w \text{ es buena para } A \text{ con respecto } Q\}.$$

Ahora definimos la función $d : \{0,1\}^* \to [0, \infty)$ como

$$d(w) = 2^{s|w|} \prod_{i=0}^{n} d_i(w^i),$$

donde $w = w^0 \dots w^n$ con $|w^i| = 2^i$ para todo $0 \leq i < n$ y $|w^n| \leq 2^n$.

Es fácil ver que d es una s-gala. Además, como cada d^i es calculable en espacio $O(2^i)$ (con $i \leq n$) e $i \leq \log |w|$, se tiene que $d \in$ pspace.

Finalmente se necesita el siguiente lema que proporcionará una cota superior del número de secuencias que son buenas para A con respecto Q.

Lema 6.2.2. Para todo $n \in \mathbb{N}$ y $Q \in Q_n$ tenemos que

$$\#B_{n,Q} \leq 2^{\mathcal{H}(\epsilon)2^n} 2^{\xi(n)},$$

donde ϵ es el parámetro de error en el algoritmo PAC A.

Demostración. Veamos cuántas hipótesis diferentes puede devolver el algoritmo A cuando usa los ejemplos de un conjunto Q fijo. Notar que cada $Q \in Q_n$ verifica que $\#Q \leq \xi(n)$, así A puede generar a lo más $2^{\xi(n)}$ hipótesis.

Sea h una de esas hipótesis y sea $\tilde{h} \in \{0,1\}^{2^n}$ su secuencia característica. Vamos a estimar el número de secuencias que son una ϵ-aproximación de $\tilde{h}$ del siguiente modo. Si

$$Approx(\epsilon, \tilde{h}) = \{w \in \{0,1\}^{2^n} \mid \#\{i \in \{0 \dots 2^n - 1\} \mid \tilde{h}[i] \neq w[i]\} \leq \epsilon 2^n\},$$

por la cota de Chernoff [26]

$$\#Approx(\epsilon, \tilde{h}) = \sum_{k=0}^{\epsilon 2^n} \binom{2^n}{k} \leq 2^{\mathcal{H}(\epsilon)2^n}.$$

Así que, para cada hipótesis h hay a lo mas $2^{\mathcal{H}(\epsilon)2^n}$ secuencias que ϵ-aproximan $\tilde{h}$ y por lo tanto, para cada $n \in \mathbb{N}$,

$$\#B_{n,Q} \leq 2^{\mathcal{H}(\epsilon)2^n} 2^{\xi(n)}.$$

□

Continuando con la demostración del Teorema 6.2.1, veamos que d tiene éxito en $\mathcal{C}$. Sea $L \in \mathcal{C}$ y sea $c \in \mathcal{C}_n$ el concepto representado por $L^{=n}$. Por un lado, como $A(n, \epsilon, \delta)$ devuelve con probabilidad $1 - \delta$ una ϵ-aproximación de c, tenemos que

$$\frac{\#Q_n(L^{=n})|}{\#Q_n} \geq 1 - \delta. \tag{6.2.2}$$

Por otro lado, por (6.2.1)

$$d_n(L^{=n}) = \sum_{Q \in Q_n(L^{=n})} \frac{1}{\#Q_n \cdot \#B_{n,Q}}.$$

Usando el Lema 6.2.2 en esta última ecuación tenemos que

$$\begin{aligned} d_n(L^{=n}) &\geq \sum_{Q \in Q_n(L^{=n})} \frac{1}{\#Q_n 2^{\mathcal{H}(\epsilon)2^n} 2^{\xi(n)}} \\ &\geq \frac{1-\delta}{2^{\mathcal{H}(\epsilon)2^n} 2^{\xi(n)}}, \end{aligned}$$

donde la última desigualdad se obtiene usando (6.2.2).

Así pues, para todo $n \in \mathbb{N}$,

$$\begin{aligned} d(L[0 \ldots 2^{n+1} - 2]) &\geq 2^{s(2^{n+1}-1)} \prod_{i=0}^{n} \frac{1-\delta}{2^{\mathcal{H}(\epsilon)2^i} 2^{\xi(i)}} \\ &= 2^{s(2^{n+1}-1)} \frac{(1-\delta)^{n+1}}{2^{\sum_{i=0}^{n} \mathcal{H}(\epsilon)2^i + \xi(i)}} \\ &= 2^{(s-\mathcal{H}(\epsilon))(2^{n+1}-1)} \frac{(1-\delta)^{n+1}}{2^{\sum_{i=0}^{n} \xi(i)}}, \end{aligned}$$

que tiende a infinito puesto que $s > \mathcal{H}(\epsilon)$. Finalmente, $\epsilon > 0$ es arbitrario y $\mathcal{H}(\epsilon)$ tiende a 0 cuando $\epsilon \to 0$ así, para todo $s > 0$, podemos definir una s-gala en pspace que tiene éxito en $\mathcal{C}$. □

Notar que el anterior teorema se cumple para cualquier clase de hipótesis que queramos considerar. Así, para todos los resultados positivos en el Ejemplo 1.3.31 (es decir, PAC aprendibles o propiamente PAC aprendibles) se puede usar el teorema anterior para obtener resultados de dimensión pspace.

Corolario 6.2.3. Las siguientes clases tienen dimensión pspace cero:

1. Conceptos en forma de conjunciones.
2. Conceptos de umbral lineal (perceptrones). De hecho, se demuestra en [27] que esta clase tiene también dimensión en tiempo polinómico cero.
3. La clase de conceptos en forma de umbrales lineales de umbrales lineales (es decir perceptrones multicapa con unidades ocultas).
4. Las clases k-DNF, k-CNF y k-listas decisionales (para cada k fijo).

El Teorema 6.2.1 puede usarse también a la inversa, obteniendo el siguiente resultado de no-aprendizaje independientemente de la clase de hipótesis que usemos.

Corolario 6.2.4. Sea $\mathcal{C}$ una clase de conceptos tal que $\dim_{\text{pspace}}(C) \neq 0$. Entonces, $\mathcal{C}$ es inherentemente impredecible, es decir, no existe ninguna clase de hipótesis para la cual $\mathcal{C}$ es PAC aprendible.

El Teorema 6.2.1 puede generalizarse a algoritmos PAC que usan un mayor número de ejemplos.

Teorema 6.2.5. Sea $\mathcal{C}$ una clase de conceptos que puede aprenderse con un algoritmo PAC que se ejecuta en espacio $O(2^n)$ con a lo más $\alpha 2^n$ ejemplos ($\alpha \leq 1$ pspace-calculable), entonces

$$\dim_{\text{pspace}}(\mathcal{C}) \leq \alpha.$$

Demostración. La demostración es análoga a la del último teorema, únicamente hay que usar $\sum_{i=0}^{n} \xi(i) \leq \alpha(2^{n+1}-1)$. De este modo,

$$d(L[0 \ldots 2^{n+1}-2]) \quad \geq \quad 2^{(s-\mathcal{H}(\epsilon)-\alpha)(2^{n+1}-1)}(1-\delta)^{n+1},$$

que tiende a infinito cuando $s > \mathcal{H}(\epsilon) + \alpha$. Finalmente, $\epsilon > 0$ es arbitrario y $\mathcal{H}(\epsilon)$ tiende a 0 cuando $\epsilon \to 0$. Así, para todo $s > \alpha$ podemos definir una s-gala en pspace tal que tiene éxito en $\mathcal{C}$. □

Finalmente se pueden obtener resultados similares para otras versiones de dimensión efectiva.

Teorema 6.2.6. Sea $\mathcal{C}$ una clase de conceptos que puede aprenderse mediante un algoritmo PAC con espacio de trabajo y número de ejemplos acotados por $p(n)$, siendo p un polinomio fijo. Entonces

$$\dim_{\text{plogon}}(\mathcal{C}) = 0.$$

Teorema 6.2.7. Sea $\mathcal{C}$ una clase de conceptos que puede aprenderse mediante un algoritmo PAC que se ejecuta en tiempo 2^n y con un número de ejemplos acotado por $p(n)$, siendo p un polinomio fijo. Entonces

$$\dim_{\text{p}_2}(\mathcal{C}) = 0.$$

Nota 6.2.8. Notar que las cotas polinómicas en el teorema anterior no implican representaciones de tamaño trivial (ni para la clase de conceptos ni para la clase de hipotésis), ya que la salida del algoritmo PAC no está acotada.

6.3. Dimensión y aprendizaje basado en preguntas de pertenencia

Esta sección se centra en el modelo de aprendizaje basado en preguntas de pertenencia. El principal resultado demuestra que la dimensión pspace de las clases de conceptos que se pueden aprender con un algoritmo basado en preguntas que se ejecuta en espacio $O(2^n)$ y hace a lo más $o(2^n)$ preguntas es cero. Esto implica que clases "grandes" en el sentido de dimensión necesitan representaciones largas para poder ser aprendibles mediante preguntas de pertenencia.

Finalmente se demuestra un resultado más fuerte bajo condiciones de aprendizaje más restrictivas.

Teorema 6.3.1. Sea $\mathcal{C}$ una clase de conceptos aprendible con un algoritmo basado en preguntas de pertenencia que se ejecuta en espacio $O(2^n)$ y hace a lo más $o(2^n)$ preguntas. Entonces,

$$\dim_{\text{pspace}}(\mathcal{C}) = 0.$$

Demostración. Sea A el algoritmo que nos asegura que $\mathcal{C}$ es aprendible con $o(2^n)$ preguntas. Sea $q(n)$ el máximo número de preguntas que hace A en una entrada n.

Sea w una secuencia de longitud 2^n. Decimos que w es buena para A si el algoritmo A en la entrada n devuelve una hipótesis h equivalente a w y el número total de preguntas hechas por A esta acotado por $q(n)$.

Sea B_n el conjunto de todas las secuencias de longitud 2^n que son buenas para A. Notar que, si sólo se permiten preguntas de pertenencia, entonces el número de diferentes salidas de $A(n)$ está acotado por $2^{q(n)}$, así que $\#B_n \leq 2^{q(n)}$.

Sea $d_n : \{0,1\}^{\leq 2^n} \to [0,\infty)$ la función definida del siguiente modo,

$$d_n(v) = \frac{\#\{w \text{ buena para } A \mid v \sqsubseteq w\}}{\#B_n}.$$

Notar que, reutilizando espacio, d_n puede calcularse en espacio $O(2^n)$. También, si w tiene longitud 2^n y es buena para A, $d_n(w) = 1/\#B_n$.

Ahora, la s-gala $d : \{0,1\}^* \to [0,\infty)$ se define como

$$d(w) = 2^{s|w|} \prod_{i=0}^{n} d_i(w^i),$$

donde $w = w^0 \dots w^n$ con $|w^i| = 2^i$ para todo $0 \leq i < n$ y $|w^n| \leq 2^n$.

Es fácil ver que d es una s-gala. También, como cada d_i es calculable en espacio $O(2^i)$ (con $i \leq n$) e $i \leq \log |w|$, $d \in$ pspace.

Finalmente, veamos que d tiene éxito en $\mathcal{C}$. Sea $L \in \mathcal{C}$ y $n \in \mathbb{N}$, entonces $L^{=n}$ es buena para A y

$$d_n(L^{=n}) \geq \frac{1}{\#B_n} \geq \frac{1}{2^{q(n)}}.$$

Así que, para todo $n \in \mathbb{N}$,

$$d(L[0 \dots 2^{n+1} - 2]) \geq 2^{s(2^{n+1}-1)} \prod_{i=0}^{n} \frac{1}{2^{q(i)}}$$

que tiende a infinito cuando $s > 0$. □

Si se permite que el número de preguntas en el Teorema 6.3.1 sea $\alpha 2^n$, entonces α es una cota superior para la dimensión pspace de $\mathcal{C}$.

Teorema 6.3.2. Sea $\mathcal{C}$ una clase de conceptos aprendible con un algoritmo basado en preguntas de pertenencia que se ejecuta en espacio $O(2^n)$ y hace a lo más $\alpha 2^n$ preguntas ($\alpha \leq 1$, α pspace calculable), entonces

$$\dim_{\text{pspace}}(\mathcal{C}) \leq \alpha.$$

Demostración. La demostración es análoga al teorema anterior, únicamente hay que considerar que $q(n) = \alpha 2^n$. En este caso, $\#B_n \leq 2^{\alpha 2^n}$ y entonces

$$\begin{aligned} d(L[0 \dots 2^{n+1} - 2]) &\geq 2^{s(2^{n+1}-1)} \prod_{i=0}^{n} \frac{1}{2^{\alpha 2^i}} \\ &= 2^{(s-\alpha)(2^{n+1}-1)}. \end{aligned}$$

que tiende a infinito cuando $s > \alpha$. Así pues, $\dim_{\text{pspace}}(\mathcal{C}) \leq \alpha$. □

El próximo teorema demuestra que el Teorema 6.3.2 es óptimo.

Teorema 6.3.3. Sea $\alpha \in \mathbb{Q} \cap (0,1)$. Existe una clase de conceptos $\mathcal{C}_\alpha$ que es aprendible con $\alpha 2^n$ preguntas de pertenencia y tal que

$$\dim_{\text{pspace}}(\mathcal{C}_\alpha) = \alpha.$$

Demostración. Usaremos la construcción del Teorema 4,3. en [68]. Sea $L \in \mathbf{C}$ y sea $L = L_0 L_1 L_2 \dots$ una partición de L con $|L_i| = \alpha 2^i$. Definimos la secuencia $\tilde{L} \in \mathbf{C}$ como la concatenación de $\tilde{L}_i = L_i 0^{2^i - |L_i|}$ con $|\tilde{L}_i| = 2^i$.

Sea $\mathcal{C}_\alpha = \{\tilde{L} \,|L \in \mathbf{C}\}$, entonces es claro que esta clase se puede aprender con un algoritmo que hace $\alpha 2^n$ preguntas. Notar que sólo es necesario preguntar por los bits que provienen de las secuencias originales L_i, porque los otros bits son todos cero, y esos son exactamente $\alpha 2^n$ para cada n.

Veamos que $\dim_{\text{pspace}}(\mathcal{C}) = \alpha$.

Primero, veremos que $\dim_{\text{pspace}}(\mathcal{C}) \leq \alpha$. Sea $s \in [0,1]$ y definamos $d : \{0,1\}^* \to [0,\infty)$ como sigue:

i) $d(\lambda) = 1$.

ii) Sea $w = w_0 \dots w_m$ con $|w_i| = 2^i$ para todo $i \leq m$ y $|w_m| \leq 2^m$, entonces

$$d(wb) = \begin{cases} 2^{s-1} d(w) & \text{si } |w_m| < \alpha 2^m \\ 2^s d(w) & \text{si } |w_m| \geq \alpha 2^m \text{ y } b = 0. \\ 0 & \text{si } |w_m| \geq \alpha 2^m \text{ y } b = 1. \end{cases}$$

Es claro que esta función es una s-gala pspace-calculable. Sea $\tilde{L} \in \mathcal{C}$, entonces

$$\begin{aligned} d(\tilde{L}[0 \dots 2^n - 2]) &= d(\tilde{L}^{=0} \dots \tilde{L}^{=n}) = \\ &= \prod_{i=0}^{n} 2^{(s-1)\alpha 2^i} 2^{s(1-\alpha)2^i} \\ &= 2^{\sum_{i=0}^{n} 2^i (s-\alpha)} = 2^{(2^{n+1}-1)(s-\alpha)} \end{aligned}$$

que tiende a infinito cuando $s > \alpha$, así que $\dim_{\text{pspace}}(\mathcal{C}) \leq \alpha$.

Veamos que $\dim_{\text{pspace}}(\mathcal{C}) \geq \alpha$ usando una técnica de diagonalización. Sea $s < \alpha$ y sea d una s-gala pspace-calculable. Vamos a construir recursivamente una secuencia $\tilde{L} \in \mathcal{C}$ tal que d no tenga éxito en $\tilde{L}$. Suponer que $\tilde{L}[0 \dots n-1]$ ha sido ya construida y sea $\tilde{L}[0 \dots n-1] = L_0 \dots L_m$ donde $|L_i| = 2^i$ y $|L_m| \leq 2^m$, definimos entonces

$$\tilde{L}[n] = \begin{cases} b & \text{si } |\tilde{L}_m| < \alpha 2^m \text{ y } d(\tilde{L}[0 \dots n-1]b) \leq d(\tilde{L}[0 \dots n-1]\bar{b}) \\ 0 & \text{si } |\tilde{L}_m| \geq \alpha 2^m. \end{cases}$$

Es claro que $\tilde{L} \in \mathcal{C}$, veamos que d no tiene éxito en $\tilde{L}$. En el mejor caso, d gana 2^s del capital en los $(1-\alpha)2^i$ últimos bits de cada $\tilde{L}_i$, y pierde al menos 2^{s-1} del capital en los otros bits (este sería el caso cuando $d(\tilde{L}[0 \dots n-1]b) = d(\tilde{L}[0 \dots n-1]\bar{b})$). Así pues,

$$\begin{aligned} d[0 \dots 2^{m-1} - 2] &= \sum_{i=0}^{m} 2^{s(1-\alpha)2^i} 2^{(s-1)\alpha 2^i} \\ &= \sum_{i=0}^{m} 2^{2^i (s-\alpha)} \end{aligned}$$

que no tiende a infinito cuando $s < \alpha$. □

Como corolario del Teorema 6.3.1, se puede deducir que clases "grandes" (en el sentido de dimensión) requieren de representaciones largas para poder ser aprendibles mediante preguntas de pertenencia.

Corolario 6.3.4. Sea $\mathcal{C}$ una clase de conceptos tal que

$$\dim_{\text{pspace}}(\mathcal{C}) \neq 0.$$

Entonces $\mathcal{C}$ no tiene representaciones de tamaño $o(2^n)$ para las cuales sea aprendible con preguntas de pertenencia.

Demostración. Por definición, $\mathcal{C}$ es aprendible por un algoritmo A de forma que el tiempo de ejecución y el número total de preguntas están acotadas por un polinomio en n y en $\mathbf{size}_n(c)$. Así, si $\mathbf{size}_n \in o(2^n)$,el tiempo de ejecución (y entonces el espacio de trabajo) es $o(2^n)$ y $\dim_{\text{pspace}}(\mathcal{C}) = 0$, lo que nos lleva a contradicción. □

Finalmente, el siguiente teorema demuestra que el Teorema 6.3.1 es también cierto para plogon-dimensión cuando restringimos a cotas polinómicas.

Teorema 6.3.5. Sea $\mathcal{C}$ una clase de conceptos aprendible con un algoritmo basado en preguntas de pertenencia que se ejecuta en espacio polinómico en n y hace a lo más un número polinómico de preguntas. Entonces,

$$\dim_{\text{plogon}}(\mathcal{C}) = 0,$$

Tanto el espacio de salida como el tiempo en el algoritmo anterior no están restringidos. De este modo representaciones de tamaño no trivial pueden aprenderse bajo las condiciones del Teorema 6.3.5.

El teorema anterior es también cierto para p_2-dimensión.

Trabajo Futuro

Esta tesis deja varias líneas abiertas para seguir investigando. Más concretamente, en el capítulo de dimensión con escala en $\{0,1\}^*$, queda abierta la cuestión de si existe la posibilidad de intercambiar supertermgalas por termgalas (Sección 2.5) y si la hipótesis que se establece como suficiente para este intercambio es más débil que el encontrar una termgala óptima. Un resultado en esta línea permitiría establecer un puente entre dimensión con escala en $\{0,1\}^*$ y predicción. Además en este capítulo y en los posteriores (Capítulos 4, 5 y 6) se establecen caracterizaciones exactas entre diversas dimensiones y otras herramientas conocidas de Teoría de la Información (diversos tipos de complejidad de Kolmogorov y compresores). Gracias a estas caracterizaciones los resultados que vayan surgiendo en un futuro relacionados con dimensión podrán interpretarse como resultados relacionados con complejidad de Kolmogorov o compresión y viceversa.

Por otro lado, en el capítulo de dimensión es compresión es necesario establecer una condición algo artificial en los compresores (aunque resulta ser lo suficientemente general como para incluir los extensores o los compresores de Lempel-Ziv). Resultaría interesante estudiar si esa condición se puede relajar de algún modo, aunque para ello sería necesario cambiar sustancialmente las demostraciones aportadas en esta tesis. Por último, sería interesante generalizar los resultados obtenidos al caso de dimensión con escala. Una de las desigualdades es posible, pero no se conoce que ocurre con la otra desigualdad. Sería interesante estudiar si dicha desigualdad es posible o si por el contrario existe algún contraejemplo.

En referencia al capítulo de dimensión de Lempel-Ziv, parece natural intentar relacionar dicha dimensión con otras dimensiones definidas recientemente ([2]) que tienen relación directa con el algoritmo de Lempel-Ziv. Además, la catástrofe del bit sigue siendo una cuestión abierta para muchas familias de funciones.

Por último, en el capítulo de dimensión y aprendizaje se estudia principalmente la pspace-dimensión de clases que son aprendibles con PAC-learning, sin embargo quedan muchas cuestiones abiertas como por ejemplo la posible conexión entre PAC-learning y p-dimensión que iluminaría cuestiones abiertas sobre la aprendibilidad de lenguajes en tiempo exponencial (EXP).

Bibliografía

[1] R.L. Rivest A. Blum. Training a 3-node neural network is NP-complete. In *Proceedings of the 1988 Workshop on Computational LearningTheory*, pages 9–198, 1988.

[2] P. Albert, E. Mayordomo, P. Moser, and S. Perifel. Bounded pushdown dimension vs Lempel Ziv information density. Technical Report cs.CC/0704.2386, Computing Research Repository, 2007.

[3] K. Ambos-Spies and E. Mayordomo. Resource-Bounded Measure and Randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics, pages 1–47. Marcel Dekker, New York, N.Y., 1997.

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

[5] K. Ambos-Spies, C. Neis, and S. A. Terwijn. Genericity and measure for exponential time. In *Proceedings of the* 19^{th} *Symposium on Mathematical Foundations of Computer Science*, Berlin, 1994. Springer-Verlag.

[6] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[7] D. Angluin. ComputationalLearning Theory: Survey and Selected Bibliography. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 351–369. ACM, 1992.

[8] 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(3):671–705, 2007.

[9] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, second edition, 1995.

[10] 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.

[11] P. Billingsley. *Ergodic theory and Information*. John Wiley & Sons, Inc., New York, N.Y., 1965.

[12] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[13] H. Buhrman and L. Longpré. Compressibility and resource bounded measure. *SIAM Journal on Computing*, 31(3), 2002.

[14] H. Buhrman and L. Torenvliet. Complete Sets and Structure in Subrecursive Classes. In *Proceedings of Logic Colloquium '96*, pages 45–78. Springer-Verlag, 1998.

[15] H. Buhrman and D. van Melkebeek. Hard Sets Are Hard to Find. *Journal of Computer and System Sciences*, 59:327–345, 1999.

[16] 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.

[17] G. J. Chaitin. A Theory of Program Size Formally Identical to Information Theory. *Journal of the Association for Computing Machinery*, 22:329–340, 1975.

[18] A. Church. A Set of Postulates for the Foundation of logic. *Annals of Mathematics*, 25:839–864, 1933.

[19] A. Church. An Unsolvable Problem of Elementary Number Theory. *The American Journal of Mathematics*, 58:345–363, 1933.

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

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

[22] D. Doty, X. Gu, J. H. Lutz, E. Mayordomo, and P. Moser. Zeta-dimension. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 283–294. Springer-Verlag, 2005.

[23] D. Doty and J. Nichols. Pushdown dimension. *Theoretical Computer Science*, 381:105–123, 2007.

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

[25] R. Gavalda, M. López-Valdés, E. Mayordomo, and N. V. Vinodchandran. Resource-bounded Dimension in Computational Learning Theory. *CoRR*, abs/1010.5470, 2010.

[26] R.L. Graham, D.E. Knuth, and O. Pataschnick. *Concrete Mathematics.* Addison Wesley, 1989.

[27] R. C. Harkins and J. M. Hitchcock. Dimension, halfspaces, and the density of hard sets. In *Proceedings of the* 13^{th} *Annual International Computing and Combinatorics Conference*, pages 129–139. Springer-Verlag, 2007.

[28] F. Hausdorff. Dimension and äusseres mass. *Math. Ann.*, 79:157–179, 1919.

[29] D. Haussler. Learning Conjunctive Concepts in Structural Domains. *Machine Learning*, 4:7–40, 1989.

[30] D. Haussler. Probably Approximately Correct Learning. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 1101–1108. AAAI Press, 1990.

[31] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of Models for Polynomial Learnability. *Information and Computation*, 95(2):129–161, 1991.

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

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

[34] J. M. Hitchcock. Fractal Dimension and Logarithmic Loss Unpredictability. *Theoretical Computer Science*, 304(1–3):431–441, 2003.

[35] J. M. Hitchcock. Gales Suffice for Constructive Dimension. *Information Processing Letters*, 86(1):9–12, 2003.

[36] J. M. Hitchcock. Small Spans in Scaled Dimension. *SIAM Journal on Computing*, 34:170–194, 2004.

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

[38] J. M. Hitchcock, M. Lopez-Valdes, and E. Mayordomo. Scaled dimension and the Kolmogorov complexity of Turing-hard sets. *Theory of Computing Systems*, 43(3–4):471–497, 2008.

[39] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. Scaled dimension and non-uniform complexity. *Journal of Computer and System Sciences*, 69:97–122, 2004.

[40] J. M. Hitchcock and N. V. Vinodchandran. Dimension, Entropy Rates, and Compression. In *Proceedings of the* 19^{th} *IEEE Conference on Computational Complexity*, pages 174–183, 2004.

[41] John M. Hitchcock, María López-valdés, and Elvira Mayordomo. Scaled dimension and the Kolmogorov complexity of Turing-hard sets. In *In Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science*, pages 476–487. Springer-Verlag, 2004.

[42] D. A. Huffman. Canonical forms for information-lossless finite-state logical machines. *IRE Trans. Circuit Theory CT-6 (Special Supplement)*, pages 41–59, 1959.

[43] Witold Hurewicz and Henry Wallman. *Dimension Theory (Princeton Mathematical Series; Vol 4)*. Princeton University Press, 1996.

[44] L.A. Pierce II and P.C. Shields. Sequences incompressible by SLZ (LZW) yet fully compressible by ULZ. *Numbers, Information and Complexity, Kluwer Academic Publishers*, pages 385–390, 2000.

[45] D. W. Juedes. Weakly complete problems are not rare. *Computational Complexity*, 5:267–283, 1995.

[46] D. W. Juedes and J. H. Lutz. The complexity and distribution of hard problems. *SIAM Journal on Computing*, 24(2):279–295, 1995.

[47] D. W. Juedes and J. H. Lutz. Completeness and Weak Completeness under Polynomial-Size circuits. *Information and Computation*, 125:13–31, 1996.

[48] M.J. Kearns and U.V. Vazirani. *An introduction to Computational Learning Theory.* MIT Press, 1994.

[49] Z. Kohavi. *Switching and Finite Automata Theory (Second Edition).* McGraw-Hill, 1978.

[50] A. A. Kurmit. *Information-Lossless Automata of Finite Order.* Wiley, 1974.

[51] L. Valiant L. Pitt. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[52] J. I. Lathrop and M. J. Strauss. A Universal Upper Bound on the Performance of the Lempel-Ziv Algorithm on Maliciously-Constructed Data. In B. Carpentieri, editor, *Compression and Complexity of Sequences '97*, pages 123–135. IEEE Computer Society Press, 1998.

[53] L. A. Levin. On the Notion of a Random Sequence. *Soviet Mathematics Doklady*, 14:1413–1416, 1973.

[54] L. A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, 10:206–210, 1974.

[55] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 2008.

[56] W. Lindner. On the polynomial time bounded measure of one-truth-table degrees and p-selectivity, 1993. Diplomarbeit, Technische Universität Berlin.

[57] W. Lindner, R. Schuler, and O. Watanabe. Resource-Bounded Measure and Learnability. *Theory Computer Systems*, 33(2):151–170, 2000.

[58] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[59] M. Lopez-Valdes. Lempel-Ziv Dimension for Lempel-Ziv compression. In *Proceedings of the 31th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 693–703. Springer-Verlag, 2006.

[60] M. Lopez-Valdes. Scaled Dimension of Individual Strings. In *Logical Approaches to Computational Barriers. Second Conference on Computability in Europe*, pages 206–214, 2006.

[61] M. Lopez-Valdes and E. Mayordomo. Dimension is Compresion. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 676–685. Springer-Verlag, 2005.

[62] J. H. Lutz. Almost Everywhere High Nonuniform Complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.

[63] J. H. Lutz. A Small Span Theorem for P/Poly-Turing Reductions. In *Proceedings of the Tenth IEEE Structure in Complexity Theory Conference*, pages 324–330. IEEE Computer Society Press, 1995.

[64] J. H. Lutz. Weakly Hard Problems. *SIAM Journal on Computing*, 24:1170–1189, 1995.

[65] 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.

[66] J. H. Lutz. Resource-bounded measure. In *Proceedings of the Thirteenth IEEE Conference on Computational Complexity*, pages 236–248, New York, 1998. IEEE Computer Society Press.

[67] J. H. Lutz. Dimension in complexity classes. *SIAM Journal on Computing*, 32:158–169, 2000.

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

[69] J. H. Lutz and E. Mayordomo. Measure, stochasticity, and the density of hard languages. *SIAM Journal on Computing*, 23:762–779, 1994.

[70] J.H. Lutz and E. Mayordomo. Dimensions of points in self-similar fractals. *SIAM Journal on Computing*, 38:1080–1112, 2008.

[71] B.B. Mandelbrot. *Fractals: Form, Chance and Dimension*. San Francisco: W.H. Freeman and Co, 1977.

[72] B.B. Mandelbrot. *The Fractal Geometry of Nature*. San Francisco: W.H. Freeman and Co, 1982.

[73] P. Martin-Löf. The Definition of Random Sequences. *Information and Control*, 9:602–619, 1966.

[74] E. Mayordomo. Almost every set in exponential time is P-bi-immune. *Theoretical Computer Science*, 136(2):487–506, 1994.

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

[76] E. Mayordomo. Measuring in PSPACE. In *Proceedings of the 7th International Meeting of Young Computer Scientists (IMYCS'92)*, volume 6, pages 93–100. Gordon and Breach Topics in Computer Science, 1994.

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

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

[79] R. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

[80] B. Ya. Ryabko. Coding of combinatorial sources and Hausdorff dimension. *Soviet Mathematics Doklady*, 30:219–222, 1984.

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

[82] D. Scheinwald. On the Lempel-Ziv proof and related topics. In *Proceedings of the IEEE*, volume 82, pages 866–871, 1994.

[83] C. P. Schnorr. Klassifikation der Zufallsgesetze nach Komplexität und Ordnung. *Z. Wahrscheinlichkeitstheorie verw. Geb.*, 16:1–21, 1970.

[84] C. P. Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5:246–258, 1971.

[85] C. P. Schnorr. Zufälligkeit und Wahrscheinlichkeit. *Lecture Notes in Mathematics*, 218, 1971.

[86] C. P. Schnorr. Process Complexity and Effective Random Tests. *Journal of Computer and System Sciences*, 7:376–388, 1973.

[87] L. Staiger. Kolmogorov Complexity and Hausdorff Dimension. *Information and Computation*, 103:159–94, 1993.

[88] L. Staiger. A Tight Upper Bound on Kolmogorov Complexity and Uniformly Optimal Prediction. *Theory of Computing Systems*, 31:215–29, 1998.

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

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

[91] A. Turing. On Computable Numbers with an Application to the "Entscheidungsproblem". In *Proceedings of the London Mathematical Society*, volume 2, pages 230–265, 1936.

[92] A. Turing. Rectification to "On Computable Numbers ...". In *Proceedings of the London Mathematical Society*, volume 4, pages 544–546, 1937.

[93] L. G. Valiant. Learning disjuctions of conjuctions. *Communications of the ACM*, 27(11):1134–1142, 1984.

[94] L. G. Valiant. A Theory of the Learnable. *Proceedings 9th IJCAI*, 1:560–566, 1985.

[95] J. Ziv and A. Lempel. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, 23:337–343, 1977.

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

[97] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25:83–124, 1970.