Universidad Nacional del Sur

Tesis de Doctorado en Ciencias de la Computación

Aleatoriedad de estado finito

Nicolás Alvarez

Bahía Blanca Argentina

2017

Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctorado en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 1 de abril de 2012 y el 6 de noviembre de 2017, bajo la dirección del Dr. Pablo Fillottrani, Profesor Asociado del Departamento de Ciencias e Ingeniería de la Computación y de la Dra. Verónica Becher, Profesora Asociada de Universidad de Buenos Aires.

. . . . . .

Nicolás Alvarez
naa@cs.uns.edu.ar
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca, 6 de noviembre de 2017

Índice general

1. Introducción **5**

2. Preliminares **9**

2.1. Alfabeto, palabras y secuencias 9

2.2. Definición de normalidad 9

3. Collares perfectos **13**

3.1. Introducción 13

3.2. Collares perfectos 14

3.2.1. Todo collar ordenado es perfecto 15

3.3. Caracterización y conteo de collares perfectos 16

3.3.1. De collares perfecto a circuitos eulerianos 17

3.3.2. El número de circuitos eulerianos en grafos astutos 19

3.3.3. Número de collares perfectos 21

3.4. Pruebas de tamaño finito y collares perfectos 22

4. Incompresibilidad en subshifts de tipo finito **27**

4.1. Introducción 27

4.2. Subshifts de tipo finito 29

4.3. Equivalencia entre definiciones de normalidad en SFTs 30

4.4. El Lema 'Hot Spot' en SFTs 35

4.5. Incompresibilidad en SFT 37

4.5.1. Compresibilidad de estado finito 37

4.5.2. Teorema Principal 39

5. Independencia de estado finito **47**

5.1. Introducción 47

5.2. Definición de independencia 48

5.2.1. k-autómatas 48

5.2.2. Independencia 51

5.3. Enunciado del Teorema de Caracterización 53
5.3.1. Frecuencias de estados 53
5.3.2. Selección 55
5.3.3. Mezcladores 56
5.3.4. Teorema de caracterización 57
5.4. Prueba del Teorema de Caracterización 57
5.4.1. De independencia a frecuencia de estados 57
5.4.2. De frecuencias de estados a independencia 62
5.4.3. Equivalencia entre independencia y propiedad de selección . . 65
5.4.4. Equivalencia entre frecuencia de estados y propiedad de mezcladores 67

6. Construcción de palabras independientes **69**
6.1. Introducción 69
6.2. Algoritmo 70
6.3. Prueba de correctitud del algoritmo 74
6.4. Complejidad computacional 76

Bibliografía **79**

Capítulo 1

Introducción

Esta es una tesis en Ciencias de la Computación. Trata sobre aleatoriedad de secuencias infinitas de símbolos. El concepto de aleatoriedad fue formalizado en el campo de la Teoría de la Computación en base a nociones de computabilidad, específicamente en base a máquinas de Turing. Una secuencia es aleatoria si todos sus prefijos son, esencialmente, algorítmicamente incompresibles. La llamamos aleatoriedad pura por estar basada en el modelo de cómputo más general, las máquinas de Turing. Variando el poder del modelo de cómputo utilizado, se obtienen nociones análogas.

En el caso de la aleatoriedad pura, la teoría está bien elaborada y relaciona el concepto de aleatoriedad con test infinitos, desarrolla el concepto de secuencias aleatoriamente independientes e incluso consigue definir instancias de secuencias aleatorias. Vemos la necesidad de explorar estos conceptos para la aleatoriedad basada en autómata finitos, y dar resultados análogos a los ya obtenidos para la noción de aleatoriedad pura.

Consideramos la noción de aleatoriedad basada en autómatas finitos, a la que llamamos aleatoriedad de estado finito. Una secuencia es aleatoria de estado finito si ningún autómata finito logra comprimir infinitos prefijos de la secuencia. Un teorema importante (que resulta del trabajo de Schnorr y Stimm [51] y de Dai, Lutz, Lathrop y Mayordomo [25]) establece que las secuencias aleatorias de estado finito coinciden con las secuencias que Émile Borel llamó normales: una secuencia es normal si todos los bloques de símbolos de igual longitud aparecen con la misma frecuencia en el límite. Por ejemplo, si consideramos el alfabeto binario, el 0 y el 1 aparecen ambos con frecuencia 1/2, los bloques 00, 01, 10, 11 aparecen con frecuencia 1/4 y así siguiendo. El interés de esta noción es que captura exactamente la condición más básica de aleatoriedad.

A pesar que la definición de normalidad fue dada hace más de 100 años, existen muchos problemas abiertos. Aquí nos concentraremos específicamente en los siguiente interrogantes:

- Nos preguntamos cuál es la familia de secuencias que pasan todas las pruebas estadísticas finitas hasta un cierto tamaño dado. Para esto introducimos los collares perfectos en el Capítulo 3. Recordemos que un collar es la clase de equivalencia de una palabra finita bajo rotaciones. Los collares perfectos de orden k son aquellos en los que cada palabra de longitud k ocurre exactamente k veces en posiciones que son todas diferentes módulo k, para cualquier convención de la posición 0. Por ejemplo, el collar [00011011] es 2-perfecto mientras que [00011110] no lo es. Damos una biyección entre los collares perfectos y los ciclos eulerianos de un grafo asociado. Y realizamos un conteo de la cantidad de collares perfectos Mostramos que cada secuencia periódica infinita cuyo período coincide con un collar perfecto de orden k supera todos las pruebas estadísticas de tamaño menor o igual que k pero no todos los test de mayor tamaño. Sobre este problema, obtuvimos una respuesta completa a los problemas planteados.

- La aleatoriedad de secuencias infinitas también puede definirse en un espacio restringido de secuencias como los subshifts de tipo finito, este concepto captura los espacios con restricciones markovianas. Un conjunto S de palabras infinitas sobre un alfabeto dado es un subshift de tipo finito si existe un conjunto finito X de bloques prohibidos tal que S coincide con el conjunto de todas las palabras infinitas que no contienen ocurrencias de los bloques en X. Si bien la definición de normalidad para subshifts de tipo finito había sido desarrollada con anterioridad y estaba caracterizada por medio de martingalas ([4], [26]). Quedaba pendiente caracterizar esta noción mediante incompresibilidad por autómatas finitos. Damos una generalización de un teorema de Piatetski-Shapiro, que brinda una caracterización de normalidad en base a una condición más sencilla que la propia definición. Esta generalización extiende el resultado clásico para todo subshift de tipo finito. El resultado principal del Capítulo 4 es una caracterización de secuencias normales en estos espacios como aquellas incompresibles mediante autómatas finitos. Queda pendiente determinar si autómatas finitos más poderosos, como los no determinísticos, autómatas con contadores o bidireccionales, logran comprimir alguna secuencia normal. Tal vez la pregunta más desafiante que queda abierta es determinar si es posible lograr la caracterización por incompresibilidad

para secuencias normales en espacios shift más generales. Creemos que la demostración que dimos para el caso de subshifts de tipo finito se puede extender para shifts sóficos (un espacio shift es sófico si los bloques prohibidos constituyen un conjunto regular).

- La noción de independencia para aleatoriedad pura dice que dos secuencias son independientes cuando ninguna ayuda a comprimir la otra usando una máquina de Turing. Para definir independencia de estado finito, es necesario utilizar el mismo concepto que en aleatoriedad pura, pero ahora en base a incompresibilidad mediante autómatas finitos (concepto introducido en [11]). En el Capítulo 5 proponemos tres clases de autómatas finitos que sirven para caracterizar pares de palabras normales e independientes en función de las corridas sobre estos autómatas. Queda pendiente formular esta noción de independencia de estado finito en términos puramente combinatoriales, sin involucrar el uso de autómatas.

- Luego de desarrollar la noción de independencia de estado finito, nos encontramos con la pregunta ¿Es posible computar instancias concretas de secuencias normales e independientes? En el Capítulo 6 damos un algoritmo que resuelve afirmativamente esta pregunta. Y que es una adaptación de un algoritmo dado por Alan Turing en 1937. Desafortunadamente la complejidad de este algoritmo resulta doblemente exponencial, en el sentido que para producir los primeros n símbolos de la salida el algoritmo realiza una cantidad de operaciones en el orden de $O(2^{2^n})$. Queda abierto el problema de encontrar un algoritmo de complejidad polinomial. También queda abierto el problema de encontrar un algoritmo que reciba una secuencia normal y produzca otra secuencia normal e independiente a la dada.

Los resultados se obtuvieron en base a técnicas de matemática discreta. En particular, para el capítulo 3 extendimos el resultado clásico de N. de Bruijn que relaciona la definición combinatoria de las secuencias de Bruijn con caminos eulerianos en grafos apropiados. El conteo de collares perfectos se basa en una inversión de Möbius sobre una fórmula obtenida por análisis espectral de grafos.

Para dar los resultados de incompresibilidad de palabras normales en subshifts de tipo finito utilizamos el teorema de Perron-Frobenius que nos habilita a relacionar la codificación dentro del subshift de tipo finito con la codificación en un espacio sin restricciones. Junto a la cota dada por Lempel y Ziv ([57]) para la relación de compresión mediante autómatas finitos operando en el espacio sin restricciones en función de la entropía de bloques. Por otro lado, dimos un argumento

combinatorio elemental para demostrar que las secuencias que no son normales admiten compresión en el subshift.

Para la caracterización de pares de palabras normales e independientes adaptamos las técnicas utilizadas para demostrar la incompresibilidad de secuencias normales por medio de autómatas finitos, pero ahora considerando autómatas con dos cintas de entrada.

Por último, la correctitud del algoritmo que produce un par de palabras normales e independientes se basa en la cota efectiva de Hardy y Wright ([31]) de la cantidad de palabras de una longitud dada tengan exceso o defecto de un símbolo respecto del valor esperado. Se trata de un algoritmo iterativo que en cada paso refina su salida, de a un bit por vez, manteniendo una invariante que asegura que en el límite la salida es un par de palabras normales e independientes.

Capítulo 2

Preliminares

2.1. Alfabeto, palabras y secuencias

Un alfabeto es un conjunto finito de al menos dos símbolos. Una palabra sobre el alfabeto A es una secuencia de elemento de A. A^ℓ es el conjunto de todas las palabras de ℓ símbolos, $A^* = \bigcup_{\ell \geq 0} A^\ell$ es el conjunto de todas las palabras finitas y A^ω es el conjunto de todas las palabras infinitas sobre A.

Denotamos con λ a la palabra vacía. Para una palabra finita w, denotamos con $|w|$ a su longitud. Numeramos a las posiciones de una palabra comenzando en 1. Para una palabra x finita o infinita y un par de posiciones $i \leq j$, notamos con $x[i..j]$ a la palabra formado por los símbolos de x en las posiciones desde i hasta j, en orden. Dadas dos palabras finitas w y u, definimos el número $|w|_u$ de *ocurrencias* de u en w como

$$|w|_u = |\{i : w[i..i + |u| - 1] = u\}|$$

y el número $\|w\|_{u,r}$ de *ocurrencias alineadas con desplazamiento* r como

$$\|w\|_{u,r} = |\{i : w[i..i + |u| - 1] = u \text{ y } i = r \text{ mód } |u|\}|$$

. El número $\|w\|_u$ de *ocurrencias alineadas* viene dado por

$$\|w\|_u = \|w\|_{u,1}$$

Por ejemplo, $|aaaa|_{aa} = 3$, $\|aaaa\|_{aa} = 2$ y $\|aaaa\|_{aa,2} = 1$.

2.2. Definición de normalidad

Como ya dijimos en la introducción, la aleatoriedad basada en autómatas finitos (a la que llamamos aleatoriedad de estado finito) coincide con el concepto de

normalidad. Decimos que una palabra x en A^ω es normal cuando todos los bloques de la misma longitud aparecen con la misma frecuencia. La formalización de este concepto se debe a Émile Borel [18]. Según cómo se contabilicen las apariciones de bloques, se obtienen tres definiciones diferentes.

A. Normalidad alineada: x es normal si para todo $\ell \in \mathbb{N}, u \in A^\ell$

$$\lim_{n\to\infty} \frac{\|x[1 \dots n\ell]\|_u}{n} = |A|^{-\ell}$$

B. Normalidad alineada fuerte: x es normal si para todo $\ell, k \in \mathbb{N}, u \in A^\ell$

$$\lim_{n\to\infty} \frac{\|x[k \dots k-1+n\ell]\|_u}{n} = |A|^{-\ell}$$

C. Normalidad no alineada: x es normal si para todo $\ell \in \mathbb{N}, u \in A^\ell$

$$\lim_{n\to\infty} \frac{|x[1 \dots n]|_u}{n} = |A|^{-\ell}$$

Las tres definiciones son equivalentes, para una demostración de ésto ver el Teorema 4.2 y el Teorema 4.5 de [20] o en [9].

En el Capítulo 4 demostramos un resultado más general sobre la equivalencia entre las diferentes definiciones de normalidad.

Borel demostró que casi todos los números reales (con respecto a la medida de Lebesgue) satisfacen que su expansión fraccionaria en cada base entera b mayor o igual a 2 es una palabra normal para el alfabeto $\{0, 1, \dots b-1\}$.

Existen muchos interrogantes sobre normalidad que aún están abiertos. Entre los más famosos se encuentra la pregunta de Borel sobre si las constantes matemáticas usuales tales como π, e o $\sqrt{2}$ son normales en alguna base, así como la conjetura de que los números algebraicos irracionales son absolutamente normales.

Uno de los primeros ejemplos de un número normal se debe a Champernowne [23], que demostró que la expansión decimal de

$$0{,}12345678910112131415161718192021222324252627\dots$$

es una palabra normal en el alfabeto $\{0, \dots 9\}$. La construcción se puede hacer en cualquier base entera, obteniendo una palabra normal en el alfabeto correspondiente a esa base.

La construcción de Champernowne se ha generalizado de muchas maneras interesantes. También hay otros métodos para obtener ejemplos de secuencias

normales, una lista de referencias aparece en [20] y se pueden encontrar ejemplos elegantes con pruebas de normalidad breves pero completas en [9] .

Cabe mencionar que la normalidad es una propiedad esperable de una secuencia aleatoria, pero no es suficiente para asegurar verdadera aleatoriedad. Decimos entonces que la normalidad es una noción débil de aleatoriedad.

Capítulo 3

Collares perfectos

3.1. Introducción

Fijemos un alfabeto finito $\mathcal{A}$ y sea $|\mathcal{A}|$ su cardinalidad. Una palabra es una secuencia finita de símbolos en el alfabeto. Una rotación es la operación que mueve el símbolo final de una palabra a la primera posición mientras se desplazan todos los símbolos restantes a la posición siguiente, o es la composición de esta operación con sí misma un número arbitrario de veces. Una palabra circular, o collar, es la clase de equivalencia de una palabra bajo rotaciones. En este capítulo presentamos la noción de *collares perfectos*. El material que presentamos aquí fue publicado en [6]:

> N. Alvarez, V. Becher, P. Ferrari and S. Yuhjtman. Perfect Necklaces, *Advances of Applied Mathematics* 80:48–61, 2016.

Definición. Un collar es (k, n)-*perfecto* si tiene longitud $n|\mathcal{A}|^k$ y cada palabra de longitud k ocurre exactamente n veces en posiciones que son diferentes modulo n para cualquier convención sobre el inicio del collar. Un collar es *perfecto* si es (k, k)-perfecto para algún k.

Los collares perfectos son una variante de los célebres collares de Bruijn [27]. Recordemos que un collar de Bruijn de orden k en el alfabeto $\mathcal{A}$ tiene longitud $|\mathcal{A}|^k$ y cada palabra de longitud k ocurre exactamente una vez. Por lo tanto, nuestros collares $(k, 1)$-perfectos coinciden con los collares de Bruijn de orden k. Para una presentación de los collares de Bruijn, incluyendo un relato histórico de su descubrimiento y redescubrimiento, véase [17]. Observemos que un collar de longitud $k|\mathcal{A}|^k$ admite k posibles descomposiciones en $|\mathcal{A}|^k$ palabras consecutivas de longitud k (sin solapamiento). Por lo tanto, un collar es (k, k)-perfecto si y sólo si tiene longitud $k|A|^k$ y cada

palabra de longitud k ocurre exactamente una vez en cada una de las k posibles descomposiciones.

Para cada k y n, damos una caracterización de collares (k, n)-perfectos en términos de circuitos eulerianos en ciertos grafos. (Corolario 3.8). Damos una fórmula cerrada para el número de collares (k, n)-perfectos (Teorema 3.14). Estos son los resultados más elaborados en este capítulo.

Demostramos que toda secuencia aritmética cuya diferencia es coprima con el tamaño del alfabeto induce un collar perfecto (Teorema 3.3). En particular, la concatenación de todas las palabras de una misma longitud en orden lexicográfico produce un collar perfecto (Corolario 3.4). Esto proporciona un ejemplo curioso de collar perfecto para cualquier longitud de palabra.

Las propiedades combinatorias de la concatenación en orden lexicográfico de todas las palabras de una misma longitud fueron consideradas por primera vez, hasta donde sabemos, por E. Barbier [7, 8] (véase también [3]). Más tarde, Champernowne utilizó esta idea para la construcción de un numero real normal en base 10. Champernowne trabajó con el alfabeto $\mathcal{A} = \{0, 1, \ldots, 9\}$ y para cada k, acotó el número de ocurrencias de cada palabra de longitud menor o igual a k en la concatenación en orden lexicográfico de todas las palabras de longitud k. Pero ni Barbier ni Champernowne mencionaron que cada palabra de longitud k se produce en esta secuencia exactamente k veces, una vez en cada una de las k diferentes rotaciones.

3.2. Collares perfectos

Sea $\theta : \mathcal{A}^* \to \mathcal{A}^*$ el *operador de rotación*, tal que para cada posición i, $(\theta w)(i) = w((i + 1) \mod |w|)$. Notamos con θ^n la aplicación del operador de rotación n veces hacia la derecha, y con θ^{-n}, la rotación n veces hacia la izquierda. Como ya se dijo, un collar es la clase de equivalencia de una palabra bajo rotaciones. Para denotar un collar, escribimos $[w]$ donde w es cualquiera de las palabras en la clase de equivalencia. Por ejemplo, si $\mathcal{A} = \{0, 1\}$,

[000] contiene una sola palabra 000, porque para cualquier n, $\theta^n(000) = 000$.

[110] contiene tres palabras $\theta^0(110) = 110$, $\theta^1(110) = 101$ y $\theta^2(110) = 011$.

Ejemplo 3.1. Sea $\mathcal{A} = \{0, 1\}$. Para mayor claridad, agregamos espacios en los ejemplos.

Para palabras de longitud 2 existen dos collares perfectos:

[00 01 10 11],

[00 10 01 11].

El siguiente es un collar perfecto para palabras de longitud 3:

[000 110 101 111 001 010 011 100].

Los siguientes no son collares perfectos,

[00 01 11 10],

[000 101 110 111 010 001 011 100].

Los llamados *códigos Gray* no forman collares perfectos, por ejemplo,

[000 001 011 010 110 111 101 100].

3.2.1. Todo collar ordenado es perfecto

Definición. Dado un alfabeto ordenado $\mathcal{A}$ y un entero positivo k, el *k-collar ordenado* tiene longitud $k|\mathcal{A}|^k$ y se obtiene por la concatenación de todas las palabras de longitud k en orden lexicográfico.

Para $\mathcal{A} = \{0, 1\}$, los k-collares ordenados para k igual 1, 2 y 3 son los siguientes:

[01],

[00 01 10 11],

[000 001 010 011 100 101 110 111].

Probaremos que para toda longitud de palabra, su correspondiente collar ordenado es perfecto. Decimos que una biyección $\sigma : \mathcal{A}^k \to \mathcal{A}^k$ es un ciclo si para cada $w \in \mathcal{A}^k$ el conjunto $\{\sigma^j(w) : 0 \leq j < |\mathcal{A}|^k\}$ equivale a $\mathcal{A}^k$. Para una palabra w escribimos $w(i \dots j)$ para denotar la subcadena de w desde la posición i hasta la j.

Lema 3.2. Sea $\mathcal{A}$ un alfabeto finito, $\sigma : \mathcal{A}^k \to \mathcal{A}^k$ un ciclo y v cualquier palabra en $\mathcal{A}^k$. Sea $s = \sigma^0(v)\sigma^1(v) \dots \sigma^{|\mathcal{A}|^k-1}(v)$. El collar $[s]$ es perfecto si y sólo si para todo ℓ tal que $0 \leq \ell < k$, para todo $x \in \mathcal{A}^\ell$ y para todo $y \in \mathcal{A}^{k-\ell}$, hay un único $w \in \mathcal{A}^k$ tal que $w(k - \ell \dots k - 1) = x$ y $(\sigma(w))(0 \dots k - \ell - 1) = y$.

Demostración. Supongamos que $[s]$ es (k, k)-perfecto. Tomemos un ℓ tal que $0 \leq \ell < k$, $x \in \mathcal{A}^\ell$ y $y \in \mathcal{A}^{k-\ell}$. Consideremos $\theta^{-\ell}s$, la *$-\ell$-ésima* rotación de s. Dado que $[s]$ es (k, k)-perfecto, xy ocurre exactamente una vez en la descomposición de $\theta^{-\ell}s$ en palabras consecutivas de longitud k. Por lo tanto, existe una única palabra w en la descomposición de s en palabras consecutivas de longitud k cuyos últimos ℓ símbolos son iguales a x y tal que los primeros $k - \ell$ símbolos de $\sigma(w)$ son iguales a y.

Por el contrario, supongamos que $[s]$ no es (k, k)-perfecto. Entonces, existe algún ℓ, $0 \leq \ell < k$, tal que la descomposición de $\theta^{-\ell}(s)$ contiene dos palabras iguales de longitud k. Esto contradice que para cada $x \in \mathcal{A}^\ell$ y cada $y \in \mathcal{A}^{k-\ell}$, existe un $w \in \mathcal{A}^k$ tal que $w(k - \ell \dots k - 1) = x$ y $(\sigma(w))(0 \dots k - \ell - 1) = y$.

□

Teorema 3.3. Consideremos el alfabeto $\mathcal{A} = \{0, \ldots, b-1\}$ donde b es un entero mayor o igual a 2, una longitud de palabra k y un entero positivo r coprimo con b. Identificamos los elementos de $\mathcal{A}^k$ con el conjunto de enteros módulo b^k de acuerdo a su representación en base b. Definimos una palabra de longitud kb^k como la yuxtaposición de los elementos de $\mathcal{A}^k$ correspondientes a la sucesión aritmética $0, r, 2r, \ldots, (b^k-1)r$. El collar asociado es perfecto.

Demostración. Dado que r es coprimo con b, el operador "sumar r" define un ciclo $\sigma : \mathcal{A}^k \to \mathcal{A}^k$. Debemos comprobar que satisface la condición del Lema 3.2. Para cualquier w tal que $w(k-\ell \ldots k-1) = x$ tenemos $\sigma(w)(k-\ell \ldots k-1) = \tilde{x}$, donde, abusando de la notación, $\tilde{x} = x + r \mod b^\ell$. Dado que la palabra $y\tilde{x}$ aparece sólo una vez en el ciclo, esto fija un único $w = \sigma^{-1}(y\tilde{x})$ con $w(k-\ell \ldots k-1) = x$ y $(\sigma(w))(0 \ldots k-\ell-1) = y$. □

Corolario 3.4. Para cualquier alfabeto ordenado $\mathcal{A}$ y cualquier longitud de palabra k, el k-collar ordenado es perfecto.

Demostración. Tomar $r = 1$ en el Teorema 3.3. □

La siguiente proposición es inmediata, así que la enunciamos sin demostración.

Proposición 3.5. Los siguientes operadores $\phi : \mathcal{A}^* \to \mathcal{A}^*$ están bien definidos en los collares y conservan la perfección. Es decir, para cada k y n y para cada $s \in \mathcal{A}^*$, si $[s]$ es (k,n) - perfecto entonces $[\phi s]$ es (k,n)-perfecto.

1. El operador que permuta los dígitos: $\phi(x_0 \ldots x_{kb^k-1}) = (\pi x_0 \ldots \pi x_{kb^k-1})$ para cualquier permutación $\pi : \mathcal{A} \to \mathcal{A}$

2. El operador de reflexión: $\phi(x_0 \ldots x_{kb^k-1}) = (x_{kb^k-1} \ldots x_0)$.

3.3. Caracterización y conteo de collares perfectos

Para caracterizar y contar los collares (k,n)-perfectos en el alfabeto $\mathcal{A}$ consideramos circuitos eulerianos en un grafo dirigido apropiado, definido en base a $\mathcal{A}$, k y n.

Recordemos que un circuito euleriano en un grafo es un camino que comienza y termina en el mismo nodo y que utiliza todos los arcos exactamente una vez. Una presentación exhaustiva del material sobre grafos que usamos en esta sección puede ser leído en las monografías [30, 55, 24]. Para el material sobre combinatoria de

palabras vea los libros [39, 40]. Notamos $m \mid n$ cuando m divide n y $\gcd(m, n)$ para el máximo divisor común entre m y n.

Definición. Sea $\mathcal{A}$ un alfabeto con cardinalidad b, sea s una longitud de palabra y sea n un entero positivo. Definimos el *gráfico astuto* $G_{s,n}$ como el gráfico dirigido, con nb^s nodos, cada nodo es un par (u, v), donde u está en $\mathcal{A}^s$ y v es un número entre 0 y $n-1$. Hay una arista de (u, v) a (u', v') si los últimos $s-1$ símbolos de u coinciden con los primeros $s-1$ símbolos de u' y $(v+1) \mod n = v'$. Observe que $G_{s,n}$ es fuertemente regular (todos los nodos tienen un grado de entrada y un grado de salida igual a b) y es fuertemente conexo (existe un camino entre cada nodo y cada uno de los nodos restantes).

Para cualquier tamaño de alfabeto, el grafo astuto $G_{k-1,1}$ coincide con un gráfico de Bruijn de palabras de longitud $k-1$; por lo tanto, los circuitos eulerianos en $G_{k-1,1}$ corresponden exactamente a los collares de Bruijn de orden k. Aunque todo circuito euleriano en el grafo astuto $G_{k-1,n}$ resulta en un collar (k, n)-perfecto, cada collar (k, n)-perfecto puede provenir de varios circuitos eulerianos en este gráfico.

3.3.1. De collares perfecto a circuitos eulerianos

De aquí en adelante, fijamos un alfabeto $\mathcal{A}$ y escribimos b para denotar su cardinalidad.

Definición. Dado un collar de longitud ℓ, $[a_0, a_1, \ldots a_{\ell-1}]$, definimos su *período* como el mínimo entero L tal que para todo entero no negativo j, $a_{j \mod \ell} = a_{(j+L) \mod \ell}$. Observe que el periodo L siempre existe, y necesariamente $L \mid \ell$. Si el período coincide con la longitud, decimos que el collar es *irreducible.*

Definición. Sean m, n números enteros positivos. Definimos $d_{m,n} = \prod p_i^{\alpha_i}$ donde $\{p_i\}$ es el conjunto de primos que dividen a m, y α_i es el exponente de p_i en la factorización de n.

Proposición 3.6. El periodo L de un collar (k, n)-perfecto satisface lo siguiente:

1. $L = jb^k$ para algún $j \mid n$.
2. $d_{b,n} \mid j$
3. El collar irreducible correspondiente de longitud $L = jb^k$ es (k, j)-perfecto.

Demostración. Sea $[s]$ (k, n)-perfecto, con $s = a_0 \ldots a_{nb^k-1}$.

1. Dado que $[s]$ tiene longitud nb^k, sabemos $L \mid nb^k$. Comprobemos que $b^k | L$. Como $[s]$ tiene un período L, $[a_0 \ldots a_{L-1}]$ es un collar donde todas las palabras de

longitud k ocurren el mismo número de veces. En caso contrario, sería imposible que se produzcan el mismo número de veces en $[s]$. Si cada palabra de longitud k ocurre j veces en $[a_0 \dots a_{L-1}]$, entonces $L = jb^k$. Como $jb^k \mid nb^k$, concluimos que $j \mid n$.

2. La palabra $a_0 \dots a_{k-1}$ ocurre en la posición 0 en s, pero también en las posiciones $L,\ 2L, \dots,\ (n/j-1)L$. Estas posiciones son de la forma qjb^k donde $0 \leq q < n/j$. Estos números deben ser todos diferentes modulo n. Equivalentemente, los n/j números de la forma qb^k, donde $0 \leq q < n/j$, son todos diferentes módulo n/j. Esta última condición se cumple exactamente cuando $\gcd(b^k, n/j) = 1$, que a su vez es equivalente a $\gcd(b, n/j) = 1$, que es equivalente a $d_{b,n}|j$

3. Como se argumenta en el punto 1, en el collar $[a_0 \dots a_{L-1}]$ cada palabra de longitud k ocurre el mismo número de veces. Si las posiciones de dos ocurrencias de una palabra dada fueran iguales modulo j entonces serían iguales modulo n, pero esto es imposible porque $[s]$ es (k,n)-perfecto. □

Proposición 3.7. Sea N un collar (k,j)-perfecto. Si n es tal que $d_{b,n} \mid j \mid n$ entonces el collar de longitud nb^k que se obtiene repitiendo N exactamente n/j veces es (k,n)-perfecto.

Demostración. Sea $\tilde{N}$ el collar que se obtiene repitiendo N exactamente n/j veces. Entonces cada palabra de longitud k se produce en $\tilde{N}$ exactamente $j \times n/j = n$ veces.

Tomemos cualquier palabra w de longitud k y sean $q_1, \dots, q_j$, enteros entre 0 y $jb^k - 1$, las posiciones de las ocurrencias de w en N para alguna convención sobre la posición inicial.

Entonces, w ocurre en $\tilde{N}$ en las posiciones $q_i + jb^k t$, donde $0 \leq t < n/j$. Supongamos que $q_{i_1} + jb^k t_1 \equiv q_{i_2} + jb^k t_2 \pmod{n}$.

Tomando módulo j concluimos que $i_1 = i_2$ porque N es (k,j)-perfecto. Entonces tenemos que $b^k t_1 \equiv b^k t_2 \pmod{n/j}$. Dado que $d_{b,n} \mid j$ tenemos $\gcd(b, n/j) = 1$, así que $t_1 \equiv t_2 \pmod{n/j}$, lo cuál implica que $t_1 = t_2$. □

Corolario 3.8. Dado un alfabeto de b símbolos, con $b \geq 2$. Sean k y n números enteros positivos. Un circuito euleriano en el grafo astuto $G_{k-1,n}$ induce un collar (k,n)-perfecto. Cada collar (k,n)-perfecto de período jb^k corresponde a j circuitos Eulerianos diferentes en $G_{k-1,j}$. Por lo tanto, el número de circuitos Eulerianos en el grafo astuto $G_{k-1,n}$ es

$$e(n) = \sum_{d_{b,n}|j|n} j\ p(j),$$

donde $p(j)$ es el número de collares (k,j)-perfectos irreducibles.

3.3.2. El número de circuitos eulerianos en grafos astutos

Sea G un grafo dirigido con n nodos. La matriz de adyacencia de un grafo G es la matriz $A(G) = (a_{i,j})_{i,j=1}^{n}$ donde $a_{i,j}$ es el número de aristas entre el nodo i y el nodo j. El polinomio característico [24] de un gráfico G se define como

$$\mathcal{P}(G;x) = \text{determinante}(xI - A(G)),$$

donde I es la matriz de identidad de dimensión $n \times n$.

El teorema BEST (por los autores Bruijn, van Aardenne-Ehrenfest, Smith y Tutte) da una fórmula para el número de circuitos eulerianos en grafos dirigidos.

Lema 3.9 (BEST Theorem [30])**.** Sea G un grafo conexo regular con n nodos. Sea w un nodo de G y sea $r_w(G)$ el número de árboles de cubrimiento orientados hacia w. El número de circuitos eulerianos en G es

$$r_w(G) \cdot \prod_{v=1}^{n} (degree(v) - 1)!$$

Lema 3.10 (Hutschenreurther, Proposition 1.4 [24])**.** Sea G un multigrafo regular con n nodos y grado b. Para un nodo arbitrario w, el número de árboles de cubrimiento $r_w(G)$ es

$$r_w(G) = \frac{1}{n} \frac{\partial}{\partial x} \mathcal{P}(G;x)|_{x=b}.$$

donde $\frac{\partial}{\partial x}$ es la derivada con respecto a x.

Dado un grafo G, su grafo de línea $\Gamma(G)$ es un grafo tal que cada nodo de $\Gamma(G)$ representa un arco de G; y dos nodos de $\Gamma(G)$ son adyacentes si y sólo si sus correspondientes arcos comparten un nodo común en G.

Lema 3.11 ([24])**.** Para todo grafo dirigido G, regular y conectado,

$$\mathcal{P}(\Gamma(G);x) = x^{m-n} \mathcal{P}(G;x),$$

donde $\Gamma(G)$ es el grafo de línea de G, m es el número de arcos de G y n es el número de nodos de G.

En el siguiente lema notamos λ para la palabra vacía, es decir, la única palabra en $\mathcal{A}^0$.

Lema 3.12. Sea b cualquier tamaño de alfabeto, k una longitud de palabra, y j un entero tal que $gcd(b,k)|j|k$. Sea $G_{0,j}$ el grafo con nodos $\{(\lambda,0),(\lambda,1),\ldots(\lambda,j-1)\}$, , con b arcos de (λ,i) a $(\lambda,i+1 \mod j)$. Entonces, $\mathcal{P}(G_{0,j};x) = x^j - b^j$.

Demostración.
Es fácil comprobar que $\mathcal{P}(G_{0,j}; x) = det(xI - A(G_{0,j}))$, que es igual a $x^j - b^j$. □

Lema 3.13. Supongamos que tenemos un alfabeto de b símbolos con $b \geq 2$. Sea k una longitud de palabra y sea j un entero positivo tal que $\gcd(b, k)|j|k$. El número de circuitos eulerianos en el gráfico astuto $G_{k-1,j}$ es $(b!)^{jb^{k-1}}b^{-k}$.

Demostración. Escribimos $\Gamma(G)$ para denotar el grafo de línea de G. Observar que para cada s positivo y para cada j, $G_{s,j} = \Gamma(G_{s-1,j})$. En esta prueba, el valor j permanecerá fijo. Dado que $G_{k-1,j}$ tiene jb^{k-1} nodos, cada uno con un grado de entrada b (y también grado de salida b), por el Lema 3.9 el número de circuitos eulerianos en $G_{k-1,j}$ es

$$r(G_{k-1,j}) \cdot \prod_{v=1}^{jb^{k-1}} (degree(v) - 1)! = r(G_{k-1,j}) \cdot (b-1)!^{jb^{k-1}}.$$

El resto de la prueba se dedica a determinar $r(G_{k-1,j})$ usando el Lema 3.10.

$$\begin{aligned}
\mathcal{P}(G_{k-1,j}; x) &= \mathcal{P}(\Gamma(G_{k-2,j}); x) \\
&= x^{b^{k-1}j - b^{k-2}j}\mathcal{P}(G_{k-2,j}; x) \\
&= x^{j(b^{k-1}-b^{k-2})}\mathcal{P}(\Gamma(G_{k-3,j}); x) \\
&= x^{j(b^{k-1}-b^{k-2})}x^{j(b^{k-2}-b^{k-3})}\mathcal{P}(G_{k-3,j}; x) \\
&= x^{j(b^{k-1}-b^{k-3})}\mathcal{P}(G_{k-3,j}; x) \\
&= \dots \\
&= x^{j(b^{k-1}-b^0)}\mathcal{P}(G_{0,j}; x) \\
&= x^{j(b^{k-1}-1)}(x^j - b^j). \\
\frac{\partial}{\partial x}\mathcal{P}(G_{k-1,j}; x) &= \frac{\partial}{\partial x}x^{j(b^{k-1}-1)}(x^j - b^j) \\
&= (jb^{k-1} - j)x^{jb^{k-1}-j-1}(x^j - b^j) + x^{jb^{k-1}-j}jx^{j-1}. \\
\frac{\partial}{\partial x}\mathcal{P}(G_{k-1,j}; x)|_{x=b} &= b^{jb^{k-1}-j}jb^{j-1}.
\end{aligned}$$

Finalmente, por el Lema 3.10,

$$r(G_{k-1,j}) = \frac{1}{jb^{k-1}}\frac{\partial}{\partial x}\mathcal{P}(G_{k-1,j}; x)|_{x=b} = \frac{1}{jb^{k-1}}b^{jb^{k-1}-j}jb^{j-1} = b^{jb^{k-1}-k}.$$

Por lo tanto, el número total de circuitos eulerianos en $G_{k-1,j}$ es

$$b^{jb^{k-1}-k}((b-1)!)^{jb^{k-1}} = b!^{jb^{k-1}}b^{-k}.$$ □

3.3.3. Número de collares perfectos

Recordemos la Definición 3.3.1, $d_{b,n} = \prod p_i^{\alpha_i}$, donde $\{p_i\}$ es el conjunto de primos que dividen a ambos b y n, y α_i es el exponente de p_i en la factorización de n. La función de Euler $\varphi(n)$ cuenta los enteros positivos menores o iguales a n que son coprimos con n.

Teorema 3.14. Supongamos que tenemos un alfabeto de b símbolos, con b $geq2$. Sea k y n números enteros positivos. El número de collares (k, n)-perfectos es

$$\frac{1}{n} \sum_{d_{b,n}|j|n} e(j)\varphi(n/j)$$

donde $e(j) = (b!)^{jb^{k-1}} b^{-k}$ es el número de circuitos eulerianos en el grafo $G_{k-1,j}$ y φ es la función de Euler.

Demostración. Sea $p(j)$ el número de collares (k, j)-perfectos irreducibles. Luego, el número de collares (k, n)-perfectos es

$$\sum_{d_{b,n}|j|n} p(j).$$

Sea $e(j)$ el número de circuitos eulerianos del grafo astuto $G_{k-1,j}$. Por el Corolario 3.8, para cada j tal que $d_{b,n}|j|n$,

$$e(j) = \sum_{d_{b,n}|\ell|j} \ell \; p(\ell).$$

Notar que $d_{b,n} = d_{b,j}$. Para simplificar la notación, en el resto de la prueba abreviamos $d_{b,n}$ como d. Entonces, escribiendo j como un múltiplo de d, obtenemos que para todo m tal que $md|n$,

$$e(md) = \sum_{i|m} id \; p(id).$$

Sea $g(m) = e(md)$ y $f(m) = p(md)\ md$. Notando μ a la función de Möbius, obtenemos

$$\begin{aligned} f(m) &= \sum_{i|m} \mu(m/i)\ g(i). \\ p(md)\ md &= \sum_{i|m} \mu(m/i)\ e(id). \\ p(md) &= \frac{1}{md} \sum_{i|m} \mu(m/i)\ e(id). \end{aligned}$$

$$\begin{aligned}\sum_{d|j|n} p(j) &= \sum_{m|n/d} \frac{1}{md} \sum_{i|m} \mu(m/i)\ e(id) \\ &= \sum_{i|n/d} e(id) \sum_{i|m|n/d} \frac{1}{md}\ \mu(m/i) \\ &= \sum_{d|j|n} e(j) \sum_{j|q|n} \frac{1}{q}\ \mu(q/j).\end{aligned}$$

Aplicando inversión de Möbius,

$$\sum_{j|q|n} \frac{1}{q}\ \mu(q/j) = \sum_{r|n/j} \frac{1}{jr}\ \mu(r) = \frac{1}{n} \sum_{r|n/j} \frac{n/j}{r}\ \mu(r) = \frac{1}{n}\ \varphi(n/j).$$

Hemos utilizado la identidad $\varphi(m) = \sum_{r|m} \frac{m}{r}\mu(r)$, la cuál es simplemente la inversión de $m = \sum_{r|m} \varphi(r)$.

Por el Lema 3.13, el número $e(j)$ de circuitos eulerianos en el grafo astuto $G_{k-1,j}$ es $(b!)^{jb^{k-1}} b^{-k}$. □

3.4. Pruebas de tamaño finito y collares perfectos

"Dada una familia finita de pruebas de aleatoriedad hay una secuencia infinita x que pasa a todas ellas, pero que será rechazada por una nueva prueba más refinada", propuso Norberto Fava.

Nuestro intento de formalizar esta afirmación nos condujo a pruebas de tamaño finito y secuencias periódicas perfectas. El resultado se resume en la Proposición 3.15.

Sea $(X_0, X_1, \dots)$ una secuencia de variables aleatorias con valores en un alfabeto dado $\mathcal{A}$ de al menos dos símbolos. Decimos que la secuencia es *aleatoria* si las variables están uniformemente distribuidas en $\mathcal{A}$ y son mutuamente independientes. Para probar si una muestra $(x_0, \dots, x_{n-1}) \in \mathcal{A}^n$ viene de una secuencia aleatoria, consideramos la siguiente configuración de prueba de hipótesis. Como es usual, notamos con $\mathbb{R}$ el conjunto de números reales.

(a) La *hipótesis*

$$H_0 : \ (X_0, X_1, \dots) \text{ es aleatoria}$$

(b) Un *tamaño de prueba* k y una *función de prueba* $t : \mathcal{A}^k \to \mathbb{R}$. Notamos

$$\tau = E_0\big[t(X_0, \dots, X_{k-1})\big] = |\mathcal{A}|^{-k} \sum_{(y_0,\dots,y_{k-1}) \in \mathcal{A}^k} t(y_0, \dots, y_{k-1}),$$

donde E_0 es la esperanza asociada con la hipótesis H_0.

(c) Una función $T_n : \mathcal{A}^n \to \mathbb{R}$ definida por

$$T_n(x_0, \dots, x_{n-1}) = \Big| \frac{1}{n} \sum_{i=0}^{n-1} t(x_i, \dots, x_{i+k-1}) - \tau \Big|$$

considerando la condición de periodicidad $x_{n+j} = x_j$. Es decir, $T_n(x_0, \dots, x_{n-1})$ es la diferencia en valor absoluto entre la media empírica de t para la muestra y el valor esperado de t bajo la hipótesis H_0.

(d) Un *error* $\varepsilon > 0$ y una *regla de decisión*

> Si $T_n(x_0, \dots, x_{n-1}) > \varepsilon$ entonces rechazar que la muestra $(x_0, \dots, x_{n-1})$ cumpla la hipótesis H_0.

En este caso decimos que *la prueba t rechaza la muestra* $(x_0, \dots, x_{n-1})$.

Decimos que esta prueba es de tamaño k porque el rechazo se decide en base a una función sobre la media empírica de t, que es una función de k coordenadas consecutivas. Las pruebas de tamaño finito incluyen a la prueba de frecuencia, la prueba de bloques, cantidad de rachas, máxima racha de unos, etc. Existen muchas pruebas (no finitas), como la prueba de la de transformada de Fourier discreta, la prueba Kolmogorov-Smirnov y muchas otras. Estas pruebas también utilizan alguna función $\tilde{T}_n$ de la muestra, no necesariamente basado en la media empírica de t. La característica en común es el uso de la distribución de $\tilde{T}_n(X_1, \dots, X_n)$ bajo H_0 para calcular la probabilidad de rechazo.

Las pruebas para H_0 se utilizan para verificar si una secuencia de números producido por un generador de números aleatorios puede considerarse aleatorio; ver Knuth [35] y la batería de pruebas propuestas por L'Ecuyer y Simard [36]. Un buen recuento de la historia de las pruebas de hipótesis es dado por Lehmann [37].

En las pruebas de hipótesis convencionales, el tamaño de la muestra n se mantiene fijo. Suponiendo H_0 y repitiendo la prueba j veces con datos independientes, la proporción de veces en que la hipótesis es rechazada converge cuando $j \to \infty$ a la probabilidad condicionada a H_0 que $T_n(X_0, \dots, X_{n-1}) > \varepsilon$. En cambio, nosotros tomamos una secuencia infinita, testeamos sus primeros n elementos y registramos el rechazo para cada n mientras $n \to \infty$.

Sea $x = (x_0, x_1, \dots)$ una secuencia infinita de símbolos en $\mathcal{A}$. Fijamos un tamaño de prueba k, una función de prueba t de tamaño k y un T_n dado por (c). Decimos que x *pasa la prueba* t si

$$\lim_{n\to\infty} T_n(x_0, \dots, x_{n-1}) = 0. \qquad (*)$$

Es decir, para cada $\varepsilon > 0$ hay un $n(x, \varepsilon)$ tal que para todo $n > n(x, \varepsilon)$ tenemos

$$T_n(x_0, \ldots, x_{n-1}) \leq \varepsilon.$$

En otras palabras, fijando la función de prueba t de tamaño k y el error ε, la prueba t rechaza $(x_0, \ldots, x_{n-1})$ a lo sumo un número finito de n's. Cuando $(*)$ no se cumple, decimos que t *rechaza a* x.

La secuencia $(X_0, X_1, \ldots)$ de variables aleatorias independientes e idénticamente distribuidas en $\mathcal{A}$ pasa cualquier prueba de tamaño finito t casi seguramente (con probabilidad 1). Esto es lo mismo que decir que el conjunto de números reales en $[0, 1]$ cuya representación en base $|\mathcal{A}|$ pasa todas las pruebas finitas tiene una medida de Lebesgue igual a 1.

Decimos que la secuencia infinita x es (k, m)-*perfecta* si x es periódica con período $m|\mathcal{A}|^k$ y el collar $[x_0 \ldots x_{m|\mathcal{A}|^k-1}]$ es (k, m)-perfecto. Recordemos que los collares $(k, 1)$-perfectos son exactamente los collares de Bruijn de orden k, por lo que la siguiente proposición considera a las secuencias infinitas de Bruijn de orden k como un caso especial: si x es de Bruijn de orden k hay una prueba de tamaño $k + 1$ que rechaza a x.

Proposición 3.15. Supongamos que el alfabeto $\mathcal{A}$ tiene al menos dos símbolos. Sea m un entero positivo y x una secuencia infinita (k, m)-perfecta. Entonces, se cumple lo siguiente:

1. La secuencia infinita x pasa todas las pruebas de tamaño $j \leq k$.

2. Para todo $h > k + \log_{|A|} m$, existe una prueba t de tamaño h tal que t rechaza a x.

Demostración. Sea b la cantidad de símbolos en $\mathcal{A}$. El período de x tiene longitud mb^k.

1. Sea t una prueba de tamaño k. Para cualquier entero positivo ℓ, por periodicidad,

$$\begin{aligned} T_{mb^k\ell} &= \Big|\frac{1}{mb^k\ell} \sum_{i=0}^{mb^k\ell-1} t(x_i, \ldots, x_{i+k-1}) - \tau\Big| \\ &= \Big|\frac{\ell}{mb^k\ell} \sum_{i=0}^{mb^k-1} t(x_i, \ldots, x_{i+k-1}) - \tau\Big| \\ &= 0 \end{aligned}$$

porque x es (k, m)-perfecta y por la definición de τ en (b).

Tomemos $j \in \{0, \ldots, mb^k - 1\}$ y usemos la anterior igualdad para obtener

$$\begin{aligned}(mb^k\ell + j)T_{mb^k\ell+j} = j\,T_j &\leq j \operatorname{máx}|t - \tau| \\ &\leq mb^k \operatorname{máx}|t - \tau|,\end{aligned}$$

donde $\operatorname{máx}|t - \tau| = \operatorname{máx}_{z_0,\ldots,z_{k-1}} |t(z_0, \ldots, z_{k-1}) - \tau|$. Luego,

$$\begin{aligned}T_{mb^k\ell+j} &\leq \frac{mb^k}{mb^k\ell + j} \operatorname{máx}|t - \tau| \\ &\leq \frac{1}{\ell} \operatorname{máx}|t - \tau| \underset{\ell\to\infty}{\longrightarrow} 0.\end{aligned}$$

Esto muestra que x pasa la prueba t. Sea $\tilde{t}$ una prueba de tamaño $j < k$. Para ver que x también pasa $\tilde{t}$, definimos t de tamaño k como

$$t(x_0, \ldots, x_{k-1}) = \tilde{t}(x_0, \ldots, x_{j-1}).$$

2. Sea h un entero tal que $h > k + \log_b m$. Luego $b^h > mb^k$ y existen más palabras $w = w_0 \ldots w_{h-1} \in \mathcal{A}^h$ que los mb^k posibles lugares de comienzo. Por lo tanto, existe al menos una palabra $\tilde{w}$ de longitud h que no está presente en la secuencia x y el test t que considera como indicador a $\tilde{w}$ rechaza x. □

Pruebas finitas y números normales. Como enunció Borel (ver [20]), un número real es simplemente normal en base b^k exactamente cuando cada bloque de longitud k se produce en la expansión de x en base b con frecuencia asintótica b^{-k}. Por lo tanto, un número real es simplemente normal en base b^k si su expansión en base b pasa todas las pruebas de tamaño menor o igual a k. Obtuvimos el resultado que para cada k y b, y para cualquier m, toda secuencia (k, m)-perfecta en el alfabeto $\{0, 1, \ldots, b-1\}$ es la expansión en base b de un número que es simplemente normal en base b^k.

Borel define la normalidad en base b como simple normalidad en las bases b^k, para todo entero positivo k. De aquí en adelante, un número es normal en base b si su expansión en base b pasa todas las pruebas estadísticas de tamaño finito.

Entonces, cada instancia de un número normal en una base dada proporciona un ejemplo de una secuencia que pasa todas las pruebas de tamaño finito. Existen varios ejemplos conocidos, como [23, 14] y las referencias en [20].

Pruebas infinitas y secuencias algorítmicamente aleatorias. Martin-Löf definió un conjunto de pruebas basadas en términos de computabilidad [43], que incluyen a todas las pruebas de tamaño finito. Las secuencias infinitas que

pasan todas esas pruebas se llaman secuencias aleatorias Martin-Löf o secuencias algorítmicamente aleatorias. Debido a la naturaleza de la definición, las secuencias algorítmicamente aleatorias no son computables pero algunas de ellas pueden ser definidas en el primer nivel de la jerarquía aritmética [28]. Dado que para cada k y m, toda secuencia (k, m)-perfecta es rechazada por alguna prueba de Martin-Löf, las secuencias (k, m)-perfectas no son algorítmicamente aleatorias.

Capítulo 4

Incompresibilidad en subshifts de tipo finito

4.1. Introducción

Los subshifts de tipo finito son espacios de palabras infinitas que tienen un conjunto finito F de bloques prohibidos, es decir que sus palabras son aquellas que no contienen ninguna ocurrencia de bloques de F. Un ejemplo sencillo es el *shift de la razón áurea* definido en el alfabeto $\{0, 1\}$ y donde el conjunto de bloques prohibidos es $F = \{11\}$, es decir que sus elementos son las palabras infinitas binarias que no contienen dos 1s consecutivos. Recibe este nombre porque sus palabras se corresponden exactamente con la representación estándar en base $\phi = (1 + \sqrt{5})/2$ (razón áurea) de los números reales en $[0, 1]$. Un número $0 \leq r \leq 1$ se escribe en base ϕ como $\beta_1\beta_2\beta_3\ldots$, si $r = \sum_{i=1}^{\infty} \beta_i\phi^{-i}$ y donde se pide que no haya dos 1s consecutivos para asegurar que la representación sea única. Llamamos shift completo al espacio de todas la palabras infinitas de símbolos en un alfabeto dado, es decir cuando el conjunto F es vacío. En adelante nos referiremos a los subshifts de tipo finito como SFT por sus siglas en inglés.

La normalidad de secuencias infinitas se puede definir para subshifts de tipo finito en función de la medida de Parry correspondiente, que es la única medida de máxima entropía para un subshift de tipo finito.

La representación en base ϕ es un caso particular de los sistemas de numeración en base Pisot, que determinan palabras infinitas en un subshift de tipo finito. Recientemente Madritsch, Scheerer y Tichy [42] dieron un algoritmo polinomial para computar un número tal que todas sus representaciones en bases de Pisot son normales.

Dado que la normalidad en el shift completo puede ser caracterizada por incompresibilidad mediante autómatas finitos, nos preguntamos si podemos dar una caracterización similar para subshifts de tipo finito. Para este resultado se utilizan autómatas finitos aumentados con una cinta de salida a los que llamamos transductores, que no pierden información. Esto significa que se puede recuperar la entrada a partir de la salida y el estado actual. El teorema de caracterización dice que una palabra es normal si y sólo si ningún transductor sin pérdida de información logra comprimir infinitos prefijos de la secuencia. Este teorema aparece en el trabajo de Lempel y Ziv [57], y también se lo cita como consecuencia de los trabajos de Schnorr y Stimm [51] y de Dai, Lathrop, Lutz, y Mayordomo [25].

Las secuencias normales en subshifts de tipo finito son aquellas en las que ninguna martingala adaptada a la medida de Parry y computable mediante autómatas finitos logra una ganancia infinita. Este resultado aparece en [4], en una forma más general para martingalas adaptadas a cualquier medida de Markov. Dada la relación directa entre predecibilidad y compresibilidad, era esperable lograr una caracterización de normalidad en SFT mediante incompresibilidad. Este es el resultado principal de este capítulo y lo presentamos como el Teorema 4.5.

La dificultad para lograr la prueba de caracterización radica en dos puntos. Primero, para comprimir una palabra debemos imponer la restricción que los autómatas den su salida como una palabra en el subshift considerado. Esto nos quita libertad a la hora de elegir un método de compresión de estado finito. Por otro lado, en el caso del shift completo basta con encontrar un bloque que tenga una frecuencia distinta de la esperada para lograr compresión mediante una aplicación directa del algoritmo de Huffman [33]. Esto no es cierto en el caso de subshifts de tipo finito, donde el hecho de que la frecuencia de un bloque de longitud ℓ difiera de la esperada no implica que la entropía de bloques de longitud ℓ esté por debajo de la necesaria para comprimir.

También en este capítulo demostramos que tres formulaciones combinatorias de normalidad en SFT resultan equivalentes. Este resultado se presenta como Teorema 4.3. Ademas, en el Teorema 4.4 damos una generalización de un teorema de Piatetski-Shapiro [47] que brinda una caracterización de normalidad en base a una condición más sencilla que las tres definiciones anteriores. De esta manera, entre los Teoremas 4.3, 4.4 y 4.5, presentamos cinco formulaciones alternativas y equivalentes de normalidad en SFT.

4.2. Subshifts de tipo finito

El *shift completo* es el conjunto A^ω de todas las secuencias infinitas $(x_n)_{n\geq 0}$ de símbolos en A. El shift σ es la función de A^ω a A^ω que mapea cada secuencia $(x_n)_{n\geq 0}$ a la secuencia $(x_n)_{n\geq 1}$, obtenida eliminando el primer símbolo.

Sea $F \subset A^*$ un conjunto de palabras finitas que llamaremos *bloques prohibidos*. El subshift X_F es el subconjunto de A^ω compuesto por las secuencias que no contienen ninguna aparición de bloques prohibidos en F. Más formalmente, es el conjunto

$$X_F = \{x : x[m \dots n] \notin F \text{ para todos los enteros } 0 \leq m \leq n\}$$

Un *espacio shift* de A^ω o simplemente un *shift* es un subconjunto X de A^ω que está cerrado para la topología producto e invariante bajo el operador shift, es decir $\sigma(X) = X$. Esto equivale a la existencia de un subconjunto $F \subset A^*$ de bloques prohibidos tales que $X = X_F$. Se dice que el espacio shift es de *tipo finito* si $X = X_F$ para algún conjunto finito F de bloques prohibidos. Para un SFT X y $\ell \in \mathbb{N}$, denotamos con $\mathcal{B}_\ell(X)$ el conjunto de bloques de longitud ℓ que ocurren en elementos de X. Por simplicidad, siempre asumimos que $F \subset A^2$, es decir que los bloques prohibidos son todos de longitud 2. Esta simplificación está justificada por el hecho de que siempre es posible codificar una palabra infinita en X, donde la máxima longitud de un bloque prohibido es ℓ, a una palabra en otro SFT X^ℓ definido en un alfabeto de tamaño $|\mathcal{B}_\ell(X)|$ y en el que los bloques prohibidos son de longitud 2. Los llamamos SFT de 1 paso. Un SFT de 1 paso X puede ser descripto por un grafo dirigido que consiste de $|A|$ nodos, uno por cada símbolo del alfabeto, y hay un arco entre los nodos a y b exactamente cuando ab no es un bloque prohibido. Llamamos $G(X)$ a este grafo y $M(X)$ a su matriz de adyacencia. Un SFT X es irreducible cuando $G(X)$ es fuertemente conexo, es decir que para todo par de nodos a y b existe un camino dirigido de a hacia b. En adelante, consideramos siempre SFTs irreducibles.

La *entropía topológica* para un espacio shift X viene dada por

$$h(X) = \lim_{l\to\infty} \frac{\log |\mathcal{B}_\ell(X)|}{\ell}$$

De la Teoría de Perron-Frobenius [52] se desprende que para un SFT irreducible $h(X) = \log \lambda(M(X))$ donde $\lambda(M(X))$ es el mayor autovalor de la matriz $M(X)$ y comúnmente se lo conoce como *valor de Perron*.

Consideramos medidas de probabilidad sobre el conjunto A^ω, utilizando la σ-álgebra inducida por los cilindros $[u]$ con $u \in A^*$, donde el cilindro correspondiente a u es definido como

$$[u] = \{x \in A^\omega : x[1 \dots |u|] = u\}$$

En lo siguiente, notaremos la medida del cilindro $[u]$ como $\mu(u)$. La medida estará bien definida si $\mu(\lambda) = 1$ y $\sum_{a \in A} \mu(ua) = \mu(u)$ para toda $u \in A^*$. Una medida es invariante si $\mu(\sigma^{-1}(X) = \mu(X)$ para todo conjunto medible X.

Existe una noción de entropía para sistemas dinámicos, y en particular para espacios shifts, llamada *entropía métrica* o *entropía de Kolmogorov-Sinai*, que asigna a cada medida invariante μ una entropía $h(\mu)$. Una medida μ invariante en X se dice que es de máxima entropía si para toda medida μ' invariante en X, $h(\mu) \geq h(\mu')$.

Diremos que μ es una medida de Markov si $\mu(ua|u) = \mu(u_{|u|}a \mid u_{|u|})$. Toda medida de Markov puede definirse en función de una matriz estocástica P y una distribución estacionaria π, esto es un vector línea tal que $\pi P = \pi$. Llamamos a tal medida $\mu_{\pi,P}$ y la definimos como

$$\mu_{\pi,P}(a_1 a_2 \dots a_k) = \pi_{a_1} P_{a_1 a_2} \dots P_{a_{k-1} a_k}$$

La entropía métrica de una medida de Markov es dada por

$$h(\mu_{\pi,P}) = \sum_{i,j \in A} \pi_i P_{ij} \log\left(\frac{1}{P_{ij}}\right)$$

con la convención que $0 \log(1/0) = 0$.

Un resultado importante en SFTs dado por Parry [45] establece que la medida de máxima entropía para un SFT X dado es única y resulta ser una medida de Markov. Llamamos a esta medida *medida de Parry* para el SFT X y la notamos con μ^X. La entropía de μ^X es igual a la entropía topológica de X.

4.3. Equivalencia entre definiciones de normalidad en SFTs

La normalidad en un SFT X puede ser dada para cualquier medida μ de Markov sobre la σ-álgebra inducida por los cilindros X. Consideramos tres formulaciones de normalidad y demostramos que son equivalentes en el Teorema 4.3.

- *Normalidad alineada:* $x \in X$ es normal si para todo $\ell \in \mathbb{N}$, $w \in A^\ell$

$$\lim_{n\to\infty} \frac{\|x[1\dots n\ell]\|_w}{n} = \mu(w)$$

- *Normalidad alineada fuerte:* $x \in X$ si para todo $\ell, k \in \mathbb{N}$, $w \in A^\ell$

$$\lim_{n\to\infty} \frac{\|\sigma^k(x)[1\dots n\ell]\|_w}{n} = \mu(w)$$

- *Normalidad no alineada* $x \in X$ si para toda $w \in A^*$

$$\lim_{n\to\infty} \frac{|x[1\dots n]|_w}{n} = \mu(w)$$

Para la noción clásica de normalidad (i.e. normalidad en el shift completo), las tres definiciones son equivalentes. Para una demostración de esto, ver Teorema 4.2 y Teorema 4.5 de [20].

Antes de demostrar el resultado de equivalencia en SFT necesitamos un par de resultados auxiliares. El siguiente lema afirma que la obtención de una cota superior o inferior apropiada para las frecuencias asintóticas de todas las palabras de una cierta longitud es suficiente para demostrar que las frecuencias límite se corresponden con la medida esperada.

Lema 4.1. *Dada una medida* μ*, una palabra infinita* $x \in A^\omega$ *y una función* $\mathrm{freq}(x, n, w)$ *que puede puede ser o bien* $\frac{|x[1\dots n]|_w}{n}$ *o bien* $\frac{\|x[1\dots n\ell]\|_w}{n}$ *para* $w \in A^\ell$*.*

Las siguientes tres afirmaciones son equivalentes:

(1) $\lim_{n\to\infty} \mathrm{freq}(x, n, w) = \mu(w)$ *para toda* $w \in A^\ell$*.*

(2) $\limsup_{n\to\infty} \mathrm{freq}(x, n, w) \leq \mu(w)$ *para toda* $w \in A^\ell$*.*

(3) $\liminf_{n\to\infty} \mathrm{freq}(x, n, w) \geq \mu(w)$ *para toda* $w \in A^\ell$*.*

Demostración. (1) $\implies$ (2) y (1) $\implies$ (3) Esto es claro porque cuando existe un límite, tanto su límite inferior como su límite superior coinciden con el límite.

(2) $\implies$ (1)

Supongamos que $\limsup_{n\to\infty} \mathrm{freq}(x, n, w) \leq \mu(w)$ para toda $w \in A^\ell$. Para cualquier $v \in A^\ell$, sabemos que

$$\begin{aligned}
\liminf_{n\to\infty} \mathrm{freq}(x, n, v) &= \liminf_{n\to\infty} \left(1 - \sum_{w\in A\setminus\{v\}} \mathrm{freq}(x, n, w)\right) \\
&= 1 - \limsup_{n\to\infty} \sum_{w\in A\setminus\{v\}} \mathrm{freq}(x, n, w)
\end{aligned}$$

$$\begin{aligned} &\geq 1 - \sum_{w \in A \setminus \{v\}} \limsup_{n\to\infty} \text{freq}(x, n, w) \\ &\geq 1 - \sum_{w \in A \setminus \{v\}} \mu(w) \\ &= \mu(v) \end{aligned}$$

Concluimos que para toda $v \in A^\ell$,

$$\limsup_{n\to\infty} \text{freq}(x, n, v) \leq \mu(v) \leq \liminf_{n\to\infty} \text{freq}(x, n, v)$$

.

y sabiendo que el límite inferior es menor o igual al límite superior

$$\lim_{n\to\infty} \text{freq}(x, n, v) = \limsup_{n\to\infty} \text{freq}(x, n, v) = \limsup_{n\to\infty} \text{freq}(x, n, v) = \mu(v)$$

para toda $v \in A^\ell$.

(3) $\implies$ (1) La prueba es casi idéntica a la del caso anterior. □

El siguiente lema nos será de utilidad para manejar frecuencias límite sobre un cierto subconjunto de prefijos con longitud múltiplo de alguna constante.

Lema 4.2. *Dado $k \in \mathbb{N}$ y una función $freq(x, n, w)$ que puede ser o bien*

$$\frac{|x[1 \dots n]|_w}{n}$$

o bien

$$\frac{\|x[1 \dots n|w|]\|_w}{n}.$$

Las siguientes tres afirmaciones son válidas

(1) $\liminf_{n\to\infty} freq(x, n, w) = \liminf_{n\to\infty} freq(x, nk, w)$

(2) $\limsup_{n\to\infty} freq(x, n, w) = \limsup_{n\to\infty} freq(x, nk, w)$

(3) $\lim_{n\to\infty} freq(x, n, w) = \lim_{n\to\infty} freq(x, nk, w)$ *si tal límite existe.*

Demostración. (1) Como $\{nk\}_{n\geq 0}$ es una subsecuencia de $\{n\}_{n\geq 0}$, vale que

$$\liminf_{n\to\infty} freq(x, n, w) \leq \liminf_{n\to\infty} freq(x, nk, w).$$

Por otro lado,

$$\liminf_{n\to\infty} freq(x, n, w) \geq \liminf_{n\to\infty} freq(x, k\lfloor n/k \rfloor, w) \frac{k\lfloor n/k \rfloor}{n} = \liminf_{n\to\infty} freq(x, k\lfloor n/k \rfloor, w).$$

La última igualdad proviene del hecho que

$$\lim_{n\to\infty} \frac{k\lfloor n/k \rfloor}{n} = 1.$$

(2) La prueba es, mutatis mutandis, la misma que para el caso (1).

(3) Se deduce de (1) y (2). □

En las pruebas presentadas a continuación, usamos implícitamente el Lema 4.2 en varias ocasiones.

Ahora sí mostramos la equivalencia entre las tres definiciones de normalidad en SFTs y su medida de Parry, y en general para cualquier medida de Markov.

Teorema 4.3. *Fijada una medida de Markov μ y una secuencia x en el SFT X, las tres afirmaciones siguientes son equivalentes:*

(1) x presenta normalidad alineada

(2) x presenta normalidad alineada fuerte

(3) x presenta normalidad no alineada

Demostración. (1) $\implies$ (2) Será suficiente demostrar que si x presenta normalidad alineada entonces $\sigma(x)$ también presenta normalidad alineada.

Para cualquier $w \in A^\ell$, $k \geq \ell$ y $1 \leq i \leq k - \ell + 1$ se define $B(k, w, i) = \{v \in A^k \ : \ v[i \dots i + |w| - 1] = w\}$. Haremos uso del hecho que para una medida invariante $\mu(B(k, w, i)) = \mu(w)$.

Para todo $w \in A^\ell$ y $r \in \mathbb{N}$.

$$\begin{aligned}
\liminf_{n\to\infty} \frac{\|\sigma(x)[1\dots n\ell]\|_w}{n} &= \liminf_{n\to\infty} \frac{\|\sigma(x)[1\dots nr\ell]\|_w}{nr} \\
&\geq \liminf_{n\to\infty} \frac{1}{r} \sum_{k=0}^{r-2} \sum_{v\in B(r\ell,w,2+\ell k)} \frac{\|x[1\dots nr\ell]\|_v}{n} \\
&= \frac{1}{r} \sum_{k=0}^{r-2} \sum_{v\in B(\ell r,w,2+\ell k)} \mu(v) \\
&= \frac{r-1}{r}\mu(w)
\end{aligned}$$

Como esta última desigualdad vale para cualquier $r \in \mathbb{N}$.

$$\liminf_{n\to\infty} \frac{\|\sigma(x)[1\dots n\ell]\|_w}{n} \geq \mu(w)$$

y usando el Lema 4.1, concluimos que

$$\lim_{n\to\infty} \frac{\|\sigma(x)[1\dots n\ell]\|_w}{n} = \mu(w)$$

(2) $\implies$ (3). Notar que para todo $w \in A^\ell$,

$$|x[1\dots n]|_w = \sum_{i=0}^{\ell-1} \|\sigma^i(x)[1\dots n-i]\|_w$$

entonces

$$\lim_{n\to\infty} \frac{|x[1\dots n]|_w}{n} = \sum_{i=0}^{\ell-1} \lim_{n\to\infty} \frac{\|\sigma^i(x)[1\dots n-i]\|_w}{n} = \sum_{i=0}^{\ell-1} \mu(w)/\ell = \mu(w)$$

(3) $\implies$ (1). Para dos cadenas $v, w \in A^*$, definimos $\|v\|_{w,*} = \max_{i=1}^{|w|} \|v\|_{w,i}$. Y, dado un $\varepsilon > 0$ y un $k \in \mathbb{N}$, definimos un conjunto de palabras de longitud $k|w| - 1$ donde la frecuencia de ocurrencias alineadas de w es mala como

$$Bad(w, k, \varepsilon) = \{v \in A^{k|w|-1} : \|v\|_{w,*} > (k-1)(\mu(w) + \varepsilon)\}$$

A partir del teorema ergódico para cadenas de Markov [44, Teorema 1.10.2] sabemos que para todo $\delta > 0$ y para todo k suficientemente grande.

$$\mu(Bad(w, k, \varepsilon)) < \delta$$

Entonces, tomando k suficientemente grande

$$\begin{aligned}
\limsup_{n\to\infty} \frac{\|x[1\dots n\ell]\|_w}{n} &= \limsup_{n\to\infty} \frac{\|x[(k-1)\ell+1\dots n\ell]\|_w}{n} \\
&\leq \limsup_{n\to\infty} \frac{1}{n(k-1)\ell} \sum_{t=1}^{(n-1)\ell+1} \|x[t\dots t+k\ell-2]\|_{w,2-t} \\
&\leq \limsup_{n\to\infty} \frac{1}{n(k-1)\ell} \sum_{t=1}^{(n-1)\ell+1} \|x[t\dots t+k\ell-2]\|_{w,*} \\
&= \limsup_{n\to\infty} \sum_{v\in A^{k\ell-1}} \frac{|x[1\dots(n+k-1)\ell-1]|_v}{n\ell} \frac{\|v\|_{w,*}}{k-1} \\
&\leq \sum_{v\in A^{k\ell-1}} \left(\limsup_{n\to\infty} \frac{|x[1\dots(n+k-1)\ell-1]|_v}{n\ell}\right) \frac{\|v\|_{w,*}}{k-1}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{v\in A^{k\ell-1}} \left(\limsup_{n\to\infty} \frac{|x[1\ldots n\ell]|_v}{n\ell}\right) \frac{\|v\|_{w,*}}{k-1} \\
&= \sum_{v\in A^{k\ell-1}} \mu(v)\ \frac{\|v\|_{w,*}}{k-1} \\
&= \sum_{v\in A^{k\ell-1}\setminus Bad(w,k,\epsilon)} \mu(v)\ \frac{\|v\|_{w,*}}{k-1} \ + \sum_{v\in Bad(w,k,\epsilon)} \mu(v)\ \frac{\|v\|_{w,*}}{k-1} \\
&\leq (\mu(w)+\varepsilon) \sum_{v\in A^{k\ell-1}\setminus Bad(w,k,\epsilon)} \mu(v) + \sum_{v\in A^{k\ell-1}\setminus Bad(w,k,\epsilon)} \mu(v) \\
&\leq \mu(w)+\varepsilon+\delta
\end{aligned}
$$

La desigualdad de la segunda línea proviene de que cada ocurrencia alineada de w en una posición $j\ell+1$ con $k-1 \leq j < n$ se cuenta $(k-1)\ell$ veces como $\|x[t\ldots t+k\ell-2]\|_{w,2-t}$ para cada $(j+1-k)\ell+2 \leq t \leq j\ell+1$. Esta técnica fue utilizada por Cassels en [22].

Dado que la desigualdad vale para cualquier $\delta, \varepsilon > 0$, concluimos que

$$\limsup_{n\to\infty} \frac{\|x[1\ldots n\ell]\|_w}{n} \leq \mu(w)$$

y usando el Lema 4.1, logramos probar

$$\lim_{n\to\infty} \frac{\|x[1\ldots n\ell]\|_w}{n} = \mu(w). \quad \square$$

4.4. El Lema 'Hot Spot' en SFTs

El siguiente resultado es una adaptación en SFTs del teorema de Piatetski-Shapiro [47] y que fue redescubierto por Borwein y Bailey [19] quienes lo llamaron "Hot Spot Lemma". Cabe mencionar que el resultado de Piatetski-Shapiro fue extendido cambiando la constante C por una función sublineal, ver referencias en [20].

Teorema 4.4 (Lema "Hot Spot" para subshifts de tipo finito)**.** *Dado un SFT X y una medida de Markov μ sobre sus cilindros invariante bajo el operador shift.*

$$\lim_{n\to\infty} \frac{\|x[1..n\ell]\|_w}{n} = \mu(w) \quad \textit{para todo } w \in A^*$$

si y sólo si existe una constante C tal que

$$\limsup_{n\to\infty} \frac{\|x[1..n\ell]\|_w}{n} < C\mu(w) \quad \textit{para todo } w \in A^*$$

Demostración. Dados $w \in A^*$, $k \in \mathbb{N}$ y $\varepsilon > 0$, definimos

$$Bad(w,k,\varepsilon) = \left\{ v \in A^{k|w|} : \left| \frac{\|v\|_w}{k} - \mu(w) \right| > \varepsilon\mu(w) \right\}$$

y $Good(w,k,\varepsilon) = A^{k|w|} \setminus Bad(w,k,\varepsilon)$.

Por el teorema ergódico para cadenas de Markov [44], para todo $\varepsilon > 0$

$$\mu(Bad(w,k,\varepsilon)) < \varepsilon$$

para valores de k suficientemente grandes.

Supongamos que existe una constante C tal que

$$\limsup_{n\to\infty} \frac{\|x[1\ldots n|w|]\|_w}{n} < C \cdot \mu(w) \quad \text{para toda } w \in A^*.$$

Vamos a probar que

$$\limsup_{n\to\infty} \frac{\|x[1\ldots n|w|\|_w}{n} \leq \mu(w).$$

$$\begin{aligned}
\limsup_{n\to\infty} \frac{\|x[1\ldots n|w|]\|_w}{n} &= \limsup_{n\to\infty} \frac{\|x[1\ldots nk|w|]\|_w}{nk} \\
&= \limsup_{n\to\infty} \sum_{v\in A^{k|w|}} \frac{\|x[1\ldots nk|w|]\|_v}{n} \ \frac{\|v\|_w}{k} \\
&= \limsup_{n\to\infty} \left(\sum_{v\in Good(w,k,\varepsilon)} \frac{\|x[1\ldots nk|w|]\|_v}{n} \ \frac{\|v\|_w}{k} \right. \\
&\quad \left. + \sum_{v\in Bad(w,k,\varepsilon)} \frac{\|x[1\ldots nk|w|]\|_v}{n} \ \frac{\|v\|_w}{k} \right) \\
&\leq (1+\varepsilon)\mu(w) \ \limsup_{n\to\infty} \sum_{v\in Good(w,k,\varepsilon)} \frac{\|x[1\ldots nk|w|]\|_v}{n} \\
&\quad + \limsup_{n\to\infty} \sum_{v\in Bad(w,k,\varepsilon)} \frac{\|x[1\ldots nk|w|]\|_v}{n} \\
&\leq (1+\varepsilon)\mu(w) + \sum_{v\in Bad(w,k,\varepsilon)} \limsup_{n\to\infty} \frac{\|x[1\ldots nk|w|]\|_v}{n} \\
&\leq (1+\varepsilon)\mu(w) + \sum_{v\in Bad(w,k,\varepsilon)} C\mu(v) \\
&\leq (1+\varepsilon)\mu(w) + C\varepsilon
\end{aligned}$$

Como la última desigualdad es válida para todo $\varepsilon > 0$,

$$\limsup_{n\to\infty} \frac{\|x[1\ldots n|w|]\|_w}{n} \leq \mu(w).$$

y usando el Lema 4.1

$$\lim_{n\to\infty} \frac{\|x[1\ldots n|w|]\|_w}{n} = \mu(w). \quad \square$$

Utilizando la técnica introducida por Cassels [22], es posible realizar una demostración muy similar del lema en el caso de ocurrencias no alineadas. Basta con definir

$$Bad(w,k,\varepsilon) = \left\{ v \in A^{k|w|-1} : \left| \frac{|v|_w}{(k-1)|w|} - \mu(w) \right| < \varepsilon\mu(w) \right\}$$

y valerse del hecho que:

$$\lim_{n\to\infty} \frac{|x[1\ldots n]|_w}{n} = \lim_{n\to\infty} \frac{1}{(k-1)|w|} \sum_{v\in A^{k|w|-1}} \frac{|x[1\ldots n]|_v}{n} |v|_w.$$

4.5. Incompresibilidad en SFT

Decimos que una palabra infinita x de un SFT X es normal si

$$\lim_{n\to\infty} \frac{\|x[1\ldots n|w|]\|_w}{n} = \mu^X(w).$$

4.5.1. Compresibilidad de estado finito

Vamos a explorar cuáles son las condiciones para que una secuencia en un SFT sea incompresible por un transductor de estado finito, logrando un teorema de caracterización de palabras normales en un SFT análogo al resultado conocido para el shift completo (i.e. normalidad de Borel).

Consideramos *transductores deterministas*. Nos concentraremos en transductores que operan en tiempo real, es decir los que procesan exactamente un símbolo del alfabeto de entrada por cada transición. Comenzamos con la definición de un transductor.

Definición. Un *transductor* determinista es una tupla $C = \langle Q, A, B, \delta, q_0 \rangle$, donde

- Q is un conjunto finito de estados
- A y B son los alfabetos de entrada y salida, respectivamente
- $\delta : Q \times A \to B^* \times Q$ is la función de transición
- $q_0 \in Q$ es el estado inicial.

El transductor C procesa palabras infinitas sobre el alfabeto A: si en el estado p se procesa el símbolo a, C se mueve al estado q y genera como salida la palabra v,

donde $\langle v, q\rangle = \delta(p, a)$. Usamos, en este caso, la notación $p \xrightarrow{a|v} q$. Notar que v puede ser la palabra vacía.

Una *corrida finita* del transductor es una secuencia finita de transiciones consecutivas

$$p_0 \xrightarrow{a_1|v_1} p_1 \xrightarrow{a_2|v_2} p_2 \cdots p_{n-1} \xrightarrow{a_n|v_n} p_n$$

y escribimos $p_0 \xrightarrow{u|v} p_n$ donde $u = a_1a_2 \cdots a_n$ y $v = v_1v_2 \cdots v_n$.

Decimos que un estado q es *alcanzable* si existe una corrida finita desde el estado inicial a q.

Una *corrida infinita* del transductor es una secuencia de transiciones consecutivas

$$p_0 \xrightarrow{a_1|v_1} p_1 \xrightarrow{a_2|v_2} p_2 \xrightarrow{a_3|v_3} p_3 \cdots$$

y escribimos $p_0 \xrightarrow{x|y} \infty$ donde $x = a_1a_2a_3 \cdots$ y $y = v_1v_2v_3 \cdots$.

Una corrida infinita es *aceptadora* si $p_0 = q_0$. Esta es la condición de aceptación de Büchi cuando todos los estados son aceptadores. Escribimos $C(x)$ para referirnos a la palabra tal que $q_0 \xrightarrow{x|C(x)} \infty$.

En lo sucesivo, un transductor es un transductor determinista a menos que se indique lo contrario.

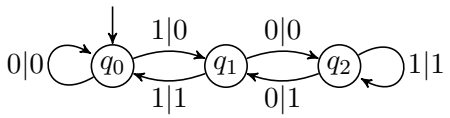

Figura 4.1: Un transductor que computa la división por 3 en base binaria

El transductor representado en la Figura 4.1 realiza la siguiente función de palabras binarias a palabras binarias. Si la entrada x es la expansión binaria de algún número real α en el intervalo unitario, entonces la salida es la expansión binaria de $\alpha/3$. Esta función no es uno-a-uno porque los racionales con denominador potencia de 2 tienen dos expansiones binarias posibles. Las dos expansiones binarias $01111\cdots$ y $10000\cdots$ de $1/2$ quedan mapeadas a la única expansión binaria $0010101\cdots$ de $1/6$.

Definición. Sea $C = \langle Q, A, B, \delta, q_0\rangle$ un transductor

1. C es *uno-a-uno* si la función $x \mapsto C(x)$ es uno-a-uno.

2. C es *sin pérdida de información* si para todo par de palabras diferentes u_1 y u_2, no sucede que $q_0 \xrightarrow{u_1|v} p$ y $q_0 \xrightarrow{u_2|v} p$ para alguna palabra v y algún estado p.

3. C es *acotado-a-uno* si la función $x \mapsto C(x)$ es acotada-a-uno.

La cualidad de ser *sin pérdida de información* está definida sobre la estructura del transductor mientras que ser uno-a-uno o acotado-a-uno depende de la función computada.

Definición (Incompresibilidad en un SFT)**.** Dado un SFT X y una secuencia $x \in X$, decimos que x es incompresible en X si no existe un transductor de estado finito $C : X \to X$ tal que

$$\liminf_{n\to\infty} \frac{|C(x[1 \dots n])|}{n} < 1.$$

Definición (Relación de compresión)**.**

- La relación de compresión de un transductor uno-a-uno C sobre una palabra finita $u \in A^*$ es
$$\rho_C(u) = \frac{|C(u)|}{|u|}$$
- La relación de compresión de C sobre una palabra infinita $x \in A^\omega$ es

$$\rho_C(x) = \liminf_{n\to\infty} \rho_C(x[1 \dots n]).$$

4.5.2. Teorema Principal

Se deduce de los resultados en [51, 25] que las palabras x con relación de compresión $\rho(x)$ igual a 1 son exactamente las palabras normales en el shift completo. Una prueba directa de este resultado aparece en [15]. En [10, 21] existen extensiones de esta caracterización para no determinismo y memoria extra.

Teorema 4.5 (Teorema Principal)**.** *Sea X un subshift de tipo finito y x una secuencia en X. La secuencia x es normal en X si y sólo es incompresible en X.*

El siguiente lema nos permite trabajar con transductores cuyo dominio es un SFT X y cuyo rango es el shift completo y luego extender los resultados para el caso en que tanto el dominio como el rango sean iguales a un SFT dado.

Lema 4.6. *Existe un transductor uno a uno $C : X \to X$ tal que $\rho_C(x) < 1$ si y sólo si existe un transductor uno a uno $C' : X \to \{0,1\}^\omega$ tal que $\rho_{C'}(x) < h(X)$.*

Demostración. De la teoría de Perron-Frobenius sabemos que dada una matriz primitiva M con valor de Perron λ, existe el límite $\lim_{n\to\infty} 1/\lambda^n \sum_{i,j} M_{ij}^{(n)}$. Por lo tanto, hay una constante c tal que para n suficientemente grande

$$1/\lambda^n \sum_{i,j} M_{ij}^{(n)} < c.$$

Tomemos $p, q \in \mathbb{N}$ tales que

$$h(X) < \frac{p}{q} < \frac{h(X)}{\rho_C(x)}.$$

Recordar que $\log \lambda = h(X)$. Dado que $\log \lambda < \frac{p}{q}$ sabemos que

$$\frac{\lambda^q}{2^p} < 1$$

y

$$\lim_{k\to\infty} \left(\frac{\lambda^q}{2^p}\right)^k = 0.$$

Entonces para toda constante $c > 0$, existe un $k_0 \in \mathbb{N}$ tal que $\lambda^{qk} c < 2^{pk}$ para todo $k \geq k_0$. Luego, existe un $k \in \mathbb{N}$ tal que $|\mathcal{B}_{qk}(X)| < \lambda^{qk} c < 2^{pk}$ y es posible hallar una función inyectiva del conjunto de palabras en $\mathcal{B}_{qk}(X)$ al conjunto $\{0,1\}^{pk}$.

Tomemos un transductor C' que simula la ejecución de C y codifica cada bloque de qk símbolos de la salida a un bloque en $\{0,1\}^{pk}$. Es fácil ver que C' es uno a uno y que

$$\liminf_{n\to\infty} \frac{|C'(x[1\ldots n])|}{n} = \frac{p}{q}\, \rho_C(x) < h(X).$$

Ahora, supongamos que existe un compresor $C' : X \to \{0,1\}^\omega$, con una relación de compresión $\rho_{C'} < h(X)$. Otra vez, usando teoría de Perron Frobenius, sabemos que $M_{0,0}^{(n)}/\lambda^n$ converge. Por lo tanto, hay una constante d tal que $M_{0,0}^{(n)} > d\lambda^n$ cuando n es suficientemente grande. Tomemos $p, q \in \mathbb{N}$ tales que

$$\frac{1}{h(X)} < \frac{p}{q} < \frac{1}{\rho_{C'}(x)}.$$

Vamos a demostrar que es posible implementar una codificación fija de bloques que mapea palabras finitas en el alfabeto binario a palabras finitas admisibles en el SFT X. Utilizamos palabras admisibles en X que están asociadas a caminos que comienzan y terminan en el estado 0, esto nos permite concatenarlas libremente.

Dado que $\frac{1}{\log \lambda} < \frac{p}{q}$, existe un k_0 tal que para todo $k \geq k_0$,

$$2^{qk} < d\lambda^{pk}.$$

Por lo tanto, existe un $k \in \mathbb{N}$ tal que $2^{qk} < M_{0,0}^{(pk)}$. Tomando un k que cumpla lo anterior, es posible definir un mapeo inyectivo desde el conjunto de palabras binarias de longitud qk a palabras admisibles en X de longitud pk asociadas a caminos que comienzan y terminan en el nodo 0, y que por lo tanto pueden ser concatenadas para formar palabras infinitas pertenecientes a X.

Podemos definir un transductor C uno a uno que simule C' y produzca bloques de pk símbolos por cada bloque de qk producidos por C. El transductor C obtenido mantiene la propiedad de ser uno-a-uno y

$$\liminf_{n\to\infty} \frac{|C(x[1\dots n])|}{n} = \frac{p}{q}\rho_{C'}(x) < 1.$$

□

Definición (Entropía de bloques)**.** Sean $u \in A^n$, $w \in A^\ell$ y $x \in A^\omega$.

- La frecuencia relativa de w en u es

$$P(w,u) = \frac{\ell}{n}\|u\|_w.$$

- La entropía de bloques ℓ-ésima de u es

$$h_\ell(u) = \frac{1}{\ell}\sum_{w\in A^\ell} P(w,u)\log\frac{1}{P(w,u)}.$$

- La entropía de bloques ℓ-ésima de x es

$$h_\ell(x) = \liminf_{k\to\infty} h_\ell(x[1\dots k\ell]).$$

- La entropía de bloques de x es

$$h(x) = \lim_{\ell\to\infty} h_\ell(x).$$

El siguiente resultado se debe a Lempel y Ziv y aparece dentro de la demostración del Teorema 3 en [57]. Por completitud damos la prueba en base a la presentación que hizo Sheinwald en [53].

Teorema 4.7. *Dado un alfabeto A y una palabra infinita $x \in A^\omega$. Para todo transductor sin pérdida de información $C : A^\omega \to \{0,1\}^\omega$:*

$$\rho_C(x) \geq h(x).$$

Demostración. Consideremos un transductor C con σ estados. Cuando C lee una palabra $w \in A^\ell$, produce una salida que depende de su estado actual. Denotamos con $L_C(w)$ la longitud de la salida más corta que produce C al leer w, donde el

mínimo se toma sobre los σ estados.

$$\rho_C(x[1\dots k\ell]) \geq \frac{1}{\ell}\sum_{w\in A^l} P(x[1\dots k\ell], w)\cdot L_C(w).$$

Luego,

$$h_\ell(x[1\dots k\ell]) - \rho_C(x[1\dots k\ell]) \quad \leq \quad \frac{1}{\ell}\sum_{w\in A^\ell} P(x[1\dots k\ell], w)\log\left(\frac{2^{-L_C(w)}}{P(x[1\dots k\ell], w)}\right).$$

Por la desigualdad de Jensen,

$$h_\ell(x[1\dots k\ell]) - \rho_C(x[1\dots k\ell]) \leq \frac{1}{\ell}\log\left(\sum_{w\in A^\ell} 2^{-L_C(w)}\right). \qquad (*)$$

Demostramos a continuación la siguiente desigualdad, que Lempel y Ziv llamaron desigualdad de Kraft generalizada,

$$\sum_{w\in A^\ell} 2^{-L_C(w)} \leq \sigma^2(1+\ell\, r_C)$$

donde r_C denota el máximo número de bits producidos por C en alguna de sus transiciones. Asociamos con cada $w \in A^\ell$ la salida más corta posible (cuya longitud será $L_C(w)$), en caso de exista más de una posibilidad tomamos una arbitraria. Dado que C es un transductor sin pérdida de información, para cada par de estados p, q y una salida dada v, puede existir a lo sumo una entrada que inicie la ejecución en p, termine en q y produzca v como salida. Esto implica que a lo sumo σ^2 palabras de A^ℓ pueden estar asociadas con la misma salida. Por lo tanto, el número k_j de palabras en A^ℓ para las cuales la salida más corta tiene longitud j es a lo sumo $\sigma^2 2^j$. Por lo tanto,

$$\sum_{w\in A^\ell} 2^{-L_C(w)} = \sum_{j=0}^{\ell r_C} k_j 2^{-j} \leq \sigma^2(1+\ell r_C).$$

A partir de (*) y la desigualdad de Kraft generalizada obtenemos

$$h_\ell(x[1\dots k\ell]) - \rho_C(x[1\dots k\ell]) \leq \frac{1}{\ell}\log(\sigma^2(1+\ell r_C)),$$

y tomando el limite de $\ell \to \infty$,

$$h(x) \leq \rho_C(x). \quad \square$$

Ahora sí podemos dar la demostración del Teorema Principal.

Prueba del Teorema 4.5. Supongamos que x es normal en X.

Sea $\phi : [0,1] \to \mathbb{R}$ la función definida como $\phi(p) = -p \log p$ usando la convención habitual que $0 \log 0 = 0$. Como $\phi(p)$ es una función continua y para toda palabra $w \in A^\ell$, $\lim_{k\to\infty} P(w, x[1 \dots k\ell]) = \mu(w)$

$$h_\ell(x) = \frac{1}{\ell} \sum_{w \in A^\ell} \phi(\mu(w)).$$

Para cualquier medida de Markov, $\sum_{w\in A^\ell} \phi(\mu(w)) = h(X_1, \dots, X_\ell)$, donde las X_i son variables aleatorias que siguen la distribución conjunta del correspondiente proceso de Markov.

$$\begin{aligned}
\lim_{\ell\to\infty} h_\ell(x) &= \lim_{\ell\to\infty} \frac{1}{\ell} h(X_1, \dots, X_\ell) \\
&= \lim_{\ell\to\infty} \frac{1}{\ell} \left(h(X_1) + \sum_{i=2}^{\ell} h(X_i|X_{i-1}) \right) \\
&= \lim_{\ell\to\infty} \frac{1}{\ell} \Big(h(X_1) + (\ell - 1) h(X_2|X_1) \Big) \\
&= h(X_2|X_1) \\
&= h(X).
\end{aligned}$$

Usando el Teorema 4.7, sabemos que no existe un transductor uno a uno $C' : X \to \{0,1\}^\omega$ con una relación de compresión menor a $h(X)$. Y a partir del Lema 4.6, concluimos que no existe transductor $C : X \to X$ tal que $\rho_C(x) < 1$.

Ahora, supongamos que x no es normal, entonces existe una palabra $w \in A^*$ tal que

$$\lim_{n\to\infty} \frac{\|x[1 \dots n|w|]\|_w}{n} \neq \mu^X(w)$$

o no existe tal límite.

Es posible seleccionar una subsecuencia de posiciones $1 \leq n_1 < n_2 < n_3 < \dots$ de modo que

$$\frac{\|x[1 \dots n_i|w|\|_v}{n_i}$$

converge para toda $v \in A^{|w|}$ y tal que el límite es diferente a $\mu(w)$ para $v = w$.

Sea $M = \#\mathcal{B}_{|w|}(X)$ y $B = \{1, 2, \dots M\}$. Podemos codificar x con una secuencia $y \in B^\omega$ tomando palabras alineadas de longitud $|w|$ en x y representándolas con un solo símbolo de B utilizando una función biyectiva $f : \mathcal{B}_{|w|}(X) \to B$. La secuencia y pertenece a un SFT Y con entropía $h(Y) = |w| \cdot h(X)$.

Para todo símbolo $a \in B$, existe el límite

$$\lim_{i\to\infty} \frac{|y[1\dots n_i]|_a}{n_i},$$

y para $b = f(w)$,

$$\lim_{i\to\infty} \frac{|y[1\dots n_i]|_b}{n_i} \neq \mu^Y(b).$$

Sea $n'_1, n'_2, \dots$ una subsecuencia de $n_1, n_2, \dots$ tal que

$$\frac{|y[1\dots n'_i]|_{ab}}{n'_i}$$

converge para todo $a, b \in B$.

Sea

$$\pi_a = \lim_{i\to\infty} \frac{|y[1\dots n'_i|_a}{n'_i}$$

$$\text{freq}(ab) = \lim_{i\to\infty} \frac{|y[1\dots n'_i|_{ab}}{n'_i}$$

$$P_{ab} = \begin{cases} \frac{\text{freq}(ab)}{\pi_a} & \text{si } \pi_a \neq 0 \\ \frac{1}{M} & \text{en caso contrario} \end{cases}$$

Dado un $k \in \mathbb{N}$, $a_1a_2\dots a_k \in B^k$, definimos una función de peso

$$W(a_1a_2\dots a_k) = \frac{1}{M}\prod_{i=1}^{k-1} P_{a_ia_{i+1}}.$$

Vamos a construir una codificación apropiada para B^k basada en los valores de W. Es necesario tener cierto cuidado para las palabras en B^k en las que W toma un valor nulo. Sea

$$S = \{u \in B^k : W(u) = 0\} \text{ y } T = B^k \setminus S.$$

Notar que para todo $u \in S$ existe algún $1 \leq i \leq k-1$ tal que $P_{u_iu_{i+1}} = 0$, lo cual implica que

$$\lim_{i\to\infty} \frac{|y[1\dots n'_i]|_{u_iu_{i+1}}}{n'_i} = 0,$$

y a su vez que

$$\lim_{i\to\infty} \frac{\|y[1\dots n'_i\|_u}{n'_i/|u|} = 0.$$

Si S no es vacío, definimos una función inyectiva

$$C_S : S \to \{0,1\}^L \text{ donde } L = \lceil \log|S| \rceil.$$

Para T, definimos un código libre de prefijos

$$C_T : T \to \{0,1\}^* \text{ tal que } |C_T(u)| = \left\lceil \log\left(\frac{1}{W(u)}\right)\right\rceil$$

la existencia de dicho código está garantizada por la desigualdad de Kraft, ya que $\sum_{u\in T} W(u) = 1$.

Sea $code(u) = \begin{cases} 0C_S(u) \text{ para } u \in S \\ 1C_T(u) \text{ para } u \in T \end{cases}$

Tomemos un transductor $C_y : Y \to \{0,1\}^\omega$ que lee palabras alineadas de k símbolos en B y las codifica usando la función *code*. Luego,

$$\begin{aligned}
\rho_{C_y}(y) &= \liminf_{n\to\infty} \frac{|C_Y(y[1\dots n])|}{n} \\
&\le \liminf_{i\to\infty} \frac{|C_Y(y[1\dots n_i'])|}{n_i'} \\
&= \liminf_{i\to\infty} \frac{1}{n_i'} \sum_{u\in B^k} \|y[1\dots n_i']\|_u \; code(u) \\
&= \liminf_{i\to\infty} \frac{1}{n_i'} \left(\sum_{u\in S} \|y[1\dots n_i']\|_u \cdot (L+1) + \sum_{u\in T} \|y[1\dots n_i']\|_u \cdot (1+|C_T(u)|) \right) \\
&= \liminf_{i\to\infty} \left((L+1) \sum_{u\in S} \cdot \frac{\|y[1\dots n_i']\|_u}{n_i'} + \frac{1}{n_i'} \sum_{u\in T} \|y[1\dots n_i']\|_u \cdot (1+|C_T(u)|) \right) \\
&= \liminf_{i\to\infty} \frac{1}{n_i'} \sum_{u\in T} \|y[1\dots n_i']\|_u \cdot (1+|C_T(u)|) \\
&= \liminf_{i\to\infty} \frac{1}{n_i'} \sum_{u\in T} \|y[1\dots n_i']\|_u \left(1 + \left\lceil \log \frac{1}{W(u)} \right\rceil\right) \\
&\le \liminf_{i\to\infty} \frac{1}{n_i'} \sum_{u\in T} \|y[1\dots n_i']\|_u \left(2 + \log \frac{M}{\prod_{j=1}^{k-1} P_{u_i u_{i+1}}}\right) \\
&= \frac{(2+\log M)\left\lfloor \frac{n_i'}{k} \right\rfloor}{n_i'} - \limsup_{i\to\infty} \frac{1}{n_i'} \sum_{u\in T} \|y[1\dots n_i']\|_u \cdot \sum_{j=1}^{k-1} \log(P_{u_i u_{i+1}}) \\
&\le \frac{2+logM}{k} - \limsup_{i\to\infty} \frac{1}{n_i'} \sum_{j=1}^{n_i'-1} \log P_{y_i y_{i+1}} \\
&= \frac{2+logM}{k} - \limsup_{i\to\infty} \sum_{a,b\in B} \frac{|y[1\dots n_i']|_{ab}}{n_i'} \log P_{ab} \\
&= \frac{2+logM}{k} - \sum_{a,b\in B} \pi_a P_{ab} \log P_{ab}
\end{aligned}$$

Como la última desigualdad es válida para todo $k \in \mathbb{N}$, y

$$-\sum_{a,b\in B} \pi_a P_{ab} \log P_{ab} = h(\mu_{\pi,P}) < h(\mu^Y) = h(Y) = |w| h(X).$$

concluimos que existe un transductor $C_y : Y \to \{0,1\}^\omega$, tal que $\rho_{C_y}(y) < |w|h(X)$.

Por últimos, definimos un transductor $C_x : X \to \{0,1\}^\omega$, que lee de la entrada palabras de longitud $|w|$, las mapea al alfabeto B usando la biyección $f : \mathcal{B}_{|w|}(X) \to B$ y simula luego el comportamiento del transductor C_y para producir una salida en el alfabeto binario.

$$\rho_{C_x}(x) = \frac{1}{|w|}\rho_{C_y(y)} < h(X),$$

lo cual implica, por el Lema 4.6, que existe un transductor $C : X \to X$, con relación de compresión $\rho_C(x) < 1$. □

Capítulo 5

Independencia de estado finito

5.1. Introducción

En [11] se introduce la noción de *independencia de estado finito* para cualquier par de palabras infinitas. En este capítulo caracterizamos esta noción de independencia específicamente para las palabras *normales*. Tal como define Émile Borel [18], al considerar un alfabeto con al menos dos símbolos, una palabra infinita x es normal si todos los bloques de símbolos de la misma longitud ocurren en x con la misma frecuencia límite. La palabra normal más conocida fue dada por Champernowne en [23],

0123456789101112131415161718192021222 3...

El libro de Bugeaud [20] da una presentación completa sobre normalidad e incluye una lista de referencias de varias construcciones conocidas de palabras normales. Borel mostró que, de hecho, casi todas las palabras son normales. Y en [11, Teorema 5.1] mostramos que casi todos los pares de palabras normales son independientes.

Uno de los resultados principales de este trabajo, expresado en el Teorema 5.1, da tres caracterizaciones para independencia de estado finito de palabras normales basado en diferentes tipos de autómatas finitos deterministas. La primera caracterización establece que dos palabras normales son independientes de estado finito cuando las frecuencias de los estados en la corrida de cualquier autómata finito determinista con dos cintas de entrada se determina sólo por el autómata, no por las palabras de entrada.

La segunda caracterización considera *selectores*, que son autómatas finitos con dos cintas de entrada y una cinta de salida. Los símbolos en la cinta de salida se obtienen como una selección de los símbolos en la primera cinta de entrada, mientras que los símbolos de la segunda cinta de entrada actúan como un oráculo.

La caracterización establece que dos palabras normales son independientes de estado finito exactamente cuando cualquier selector que los tenga como entrada produce también una palabra normal. Este resultado sobre la selección por autómatas finitos extiende el obtenido por Agafonov [1] y cae fuera de las reglas deterministas que preservan la normalidad dada por Kamae y Weiss [34].

La tercera caracterización dada en el Teorema 5.1 considera *mezcladores*, que son autómatas finitos con dos cintas de entrada y una cinta de salida tal que, después de la ejecución, la cinta de salida contiene todos los símbolos de las dos palabras normales pero mezcladas. Una presentación general de autómatas finitos mezcladores se puede leer en [48]. Usamos el término mezclar, no en el sentido de Diaconis Persis, porque los símbolos no se permutan. En cambio, la noción de mezclar que usamos supone dos palabras de entrada que dan como resultado una nueva palabra que intercala símbolos de cada una de ellas, preservando el orden en que aparecen en la entrada. La caracterización dada en el Teorema 5.1 demuestra que dos palabras normales son independientes de estado finito cuando todo autómata mezclador produce una palabra que también es normal.

5.2. Definición de independencia

5.2.1. k-autómatas

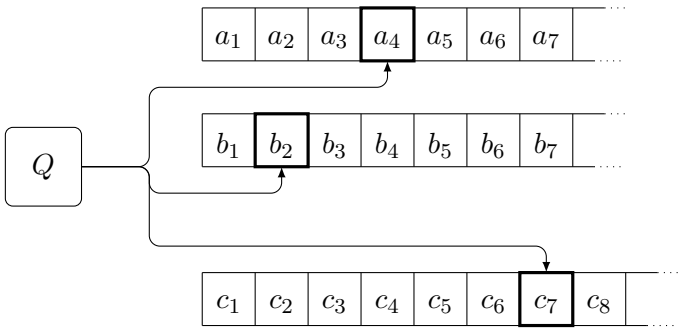

Figura 5.1: Funcionamiento de un 3-autómata.

Consideramos autómatas finitos que se ejecutan sobre una tupla de palabras infinitas sin condición de aceptación. Como se explica a continuación, el único requisito para que una corrida sea aceptadora es que todas sus etiquetas sean palabras infinitas. En particular, consideramos autómatas de k cintas, también conocidos como transductores de k cintas, para $k = 2$ y $k = 3$. Y los llamaremos k-autómatas. Una presentación exhaustiva de estos autómatas se puede encontrar en los libros [46, 49]. Usamos 2-autómatas para calcular funciones de palabras infinitas

a palabras infinitas. Y usamos 3-autómatas para calcular funciones ya sea de pares de palabras infinitas a palabras infinitas, o de palabras infinitas a pares de palabras infinitas.

Damos nombres como *compresores*, *selectores*, *mezcladores* o *divisores* a algunas subclases de estos autómatas para enfatizar su uso. Para simplificar la presentación, asumimos aquí que los alfabetos de entrada y salida de todos los autómatas son el mismo alfabeto A. Un *k-autómata* es una tupla $\mathcal{A} = \langle Q, A, \delta, I\rangle$, donde Q es el conjunto de estados, A es el alfabeto, δ es la relación de transición, e I el conjunto de estados iniciales. La relación de transición es un subconjunto de $Q \times (A \cup \{\lambda\})^k \times Q$. Por lo tanto, una transición es una tupla $\langle p, \alpha_1, \ldots, \alpha_k, q\rangle$ donde p es su *estado inicial*, $\langle \alpha_1, \ldots, \alpha_k\rangle$ es su *etiqueta* y q es su *estado final*. Cada α_i es aquí un símbolo a_i del alfabeto o la palabra vacía λ. Una transición se denota como $p \xrightarrow{\alpha_1,\ldots,\alpha_k} q$. Dos transiciones son *consecutivas* si el estado final de la primera es igual al estado inicial de la segunda.

Una *corrida finita* es una secuencia finita de transiciones consecutivas

$$q_0 \xrightarrow{\alpha_{1,1},\ldots,\alpha_{k,1}} q_1 \xrightarrow{\alpha_{1,2},\ldots,\alpha_{k,2}} q_2 \cdots q_{n-1} \xrightarrow{\alpha_{1,n},\ldots,\alpha_{k,n}} q_n$$

La *etiqueta* de una corrida es la concatenación de las etiquetas de sus transiciones. Más precisamente, es la tupla $\langle u_1, \ldots, u_k\rangle$ donde cada u_j para $1 \leq j \leq k$ es igual a $\alpha_{j,1}\alpha_{j,2}\cdots\alpha_{j,n}$. Tal corrida se denota brevemente como $q_0 \xrightarrow{u_1,\ldots,u_k} q_n$.

Una *corrida infinita* es una secuencia infinita de transiciones consecutivas

$$q_0 \xrightarrow{\alpha_{1,1},\ldots,\alpha_{k,1}} q_1 \xrightarrow{\alpha_{1,2},\ldots,\alpha_{k,2}} q_2 \xrightarrow{\alpha_{1,3},\ldots,\alpha_{k,3}} q_3 \cdots$$

La *etiqueta* de la corrida es la concatenación de las etiquetas de las transiciones. Más precisamente, es la tupla $\langle x_1, \ldots, x_k\rangle$ donde cada x_j para $1 \leq j \leq k$ es igual a $\alpha_{j,1}\alpha_{j,2}\alpha_{j,3}\cdots$. Observar que algunas etiquetas x_j pueden ser finitas aunque la corrida sea infinita, ya que algunas transiciones pueden tener etiquetas vacías.

La corrida es aceptadora si su primer estado q_0 es inicial y cada palabra x_j es infinita. Dicha corrida aceptadora se denota brevemente como $q_0 \xrightarrow{x_1,\ldots,x_k} \infty$. La tupla $\langle x_1, \ldots, x_k\rangle$ es aceptada si existe al menos una corrida aceptadora con etiqueta $\langle x_1, \ldots, x_k\rangle$.

En este trabajo sólo consideramos k-autómatas deterministas cuya función de transición está determinada por un subconjunto de las k cintas. Comenzamos con algunos definiciones. El *soporte* de una tupla $\langle \alpha_1, \ldots, \alpha_\ell\rangle$ en $(A \cup \{\lambda\})^\ell$ es el conjunto de las posiciones de la tupla donde aparecen símbolos de A, la *ℓ-etiqueta* de una

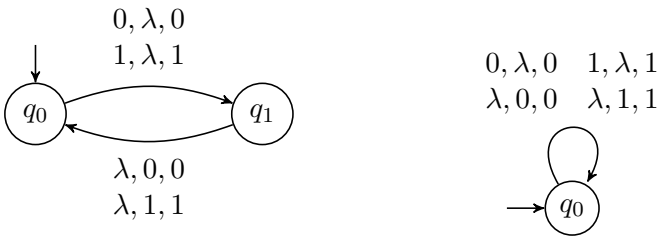

Figura 5.2: Un 3-autómata 2-determinista (izquierda) y un 3-autómata no determinista (derecha)

transición $p \xrightarrow{\alpha_1,\ldots,\alpha_k} q$ es la tupla $\langle \alpha_1, \ldots, \alpha_\ell \rangle$ y su *ℓ-soporte* es el soporte de su ℓ-etiqueta.

Decimos que un k-autómata es *ℓ-determinista*, con $1 \leq \ell \leq k$, si las siguientes dos condiciones se satisfacen:

1. el conjunto I de estados iniciales contiene un sólo elemento;
2. para todo estado q, todas las transiciones que comienzan en q tienen el mismo ℓ-soporte pero todas sus ℓ-etiquetas son diferentes.

Si el autómata es ℓ-determinista, llamamos al ℓ-soporte de todas las transiciones de un estado q el ℓ-soporte de q. Decimos que un autómata es *ℓ-completo* si para toda tupla $\alpha = \langle \alpha_1, \ldots, \alpha_\ell \rangle$ y todo estado q, existe una transición que comienza en q con ℓ-etiqueta igual a α. El ℓ-determinismo (la ℓ-completitud, respectivamente) garantiza que para cada tupla $\langle x_1, \ldots, x_\ell \rangle$ de palabras infinitas, existe a lo sumo (al menos, respectivamente) una corrida tal que las primeras ℓ componentes de su etiqueta son $\langle x_1, \ldots, x_\ell \rangle$. Sin embargo, esta corrida podría no ser aceptadora ya que una de sus etiquetas podría no ser infinita.

El 3-autómata de la izquierda en la figura 5.2 acepta una tripla $\langle x, y, z \rangle$ de palabras infinitas sobre el alfabeto $\{0, 1\}$ cuando z es la unión de los símbolos de x e y; recordar que la unión de dos palabras infinitas $x = a_1 a_2 a_3 \cdots$ e $y = b_1 b_2 b_3 \cdots$ es la palabra infinita $z = a_1 b_1 a_2 b_2 a_3 \cdots$. Este autómata es 2-determinista. El 3-autómata que se muestra en la derecha de la figura 5.2 acepta una tripla $\langle x, y, z \rangle$ de palabras infinitas sobre el alfabeto $\{0, 1\}$ siempre que z sea una intercalación de los símbolos en x y y. Este autómata es 2-no-determinista. De hecho, la primera condición sobre las transiciones no es satisfecha por las dos transiciones $q_0 \xrightarrow{0,\lambda,0} q_0$ y $q_0 \xrightarrow{\lambda,0,0} q_0$.

Sea $\mathcal{A}$ un k-autómata ℓ-determinista. Para cada tupla $\langle x_1, \ldots, x_\ell \rangle$ de palabras infinitas, existe al menos una tupla $\langle y_{\ell+1}, \ldots, y_k \rangle$ de palabras infinitas tales que la k-tupla $\langle x_1, \ldots, x_\ell, y_{\ell+1}, \ldots, y_k \rangle$ es aceptada por $\mathcal{A}$. El autómata $\mathcal{A}$ computa entonces una función parcial de $(A^\omega)^\ell$ a $(A^\omega)^{k-\ell}$ y denotamos la tupla $\langle y_{\ell+1}, \ldots, y_k \rangle$ como $\mathcal{A}(x_1, \ldots, x_\ell)$. Los 2-autómatas 1-deterministas también son

llamados transductores secuenciales en la literatura. Cuando un k-autómata es ℓ-determinista, sus transiciones se escriben

$$p \xrightarrow{\alpha_1,\ldots,\alpha_\ell|\beta_{\ell+1},\ldots,\beta_k} q$$

para remarcar que las primeras ℓ cintas constituyen la entrada y las $k-\ell$ restantes son cintas de salida.

Sea $\mathcal{A}$ un 2-autómata 1-determinista. Decimos que $\mathcal{A}$ es un *compresor* si la función (parcial) $x \mapsto \mathcal{A}(x)$ que mapea x a la salida $\mathcal{A}(x)$ es inyectiva.

La relación de compresión de una palabra infinita x para un compresor $\mathcal{A}$ es definida a partir de la única posible corrida aceptadora

$$q_0 \xrightarrow{u_1|v_1} q_1 \xrightarrow{u_2|v_2} q_2 \xrightarrow{u_3|v_3} q_3 \cdots$$

donde $x = u_1u_2u_3\cdots$ de la siguiente manera

$$\rho_{\mathcal{A}}(x) = \liminf_{n\to\infty} \frac{|v_1v_2\ldots v_n|}{|u_1u_2\ldots v_n|}.$$

Esta relación de compresión para un autómata dado $\mathcal{A}$ puede tomar cualquier valor real no negativo. En particular, puede ser mayor a 1. Una palabra infinita x es *compresible* por un 2-autómata 1-determinista $\mathcal{A}$ si $\rho_{\mathcal{A}}(x) < 1$. La relación de compresión de una palabra dada x, $\rho(x)$, es el ínfimo de las relaciones de compresión alcanzables por todos los 2-autómatas 1-deterministas inyectivos, es decir,

$$\rho(x) = \text{ínf}\{\rho_{\mathcal{A}}(x) : \mathcal{A} \text{ es un 2-autómata 1-determinista inyectivo}\}$$

Para toda palabra infinita x, $\rho(x)$ es menor o igual a 1, dado que existe un compresor $\mathcal{A}_0$ que copia cada símbolo de la entrada a la salida, entonces $\rho_{\mathcal{A}_0}(x)$ es igual a 1. La relación de compresión de la palabra $x = 0^\omega$ es $\rho(x) = 0$ porque para todo número real positivo ε existe un compresor $\mathcal{A}$ tal que $\rho_{\mathcal{A}}(x) < \varepsilon$. Observe que en este caso la compresión igual a 0 no es alcanzable por ningún compresor $\mathcal{A}$. Se deduce de los resultados en [51, 25] que las palabras x con relación de compresión $\rho(x)$ igual a 1 son exactamente las palabras normales. Una prueba directa de este resultado aparece en [15].

5.2.2. Independencia

A grandes rasgos, dos palabras infinitas son independientes de estado finito si ninguna de ellas sirve para comprimir a la otra utilizando 3-autómatas 2-

deterministas. En este contexto, un *compresor* es un 3-autómata 2-determinista $\mathcal{A}$ tal que al fijar cualquier palabra infinita y, la función $x \mapsto \mathcal{A}(x, y)$ es inyectiva. Esto garantiza que conociendo la palabra y, x se puede recuperar a partir de $\mathcal{A}(x, y)$. Notar que no pedimos que la función $(x, y) \mapsto \mathcal{A}(x, y)$ sea inyectiva, lo que sería una suposición mucho más fuerte. Por ejemplo, el 3-autómata 2-determinista $\mathcal{C}$ que mapea las palabras infinitas x e y a la palabra infinita z tal que $z[i] = x[i] + y[i]$ mód $|A|$ para cada $i \geq 1$ es, de hecho, un compresor pero la función $(x, y) \mapsto \mathcal{C}(x, y)$ no es inyectiva.

Definición ([11])**.** Sea $\mathcal{A}$ un compresor. Por simplicidad en la presentación asumimos que trabajamos con un único alfabeto. Sin embargo, es posible tener tres alfabetos diferentes, uno para cada cinta de entrada y otro para la cinta de salida. La *relación de compresión condicional* de una palabra infinita x con respecto a y para el autómata $\mathcal{A}$ es definido a partir de la única corrida aceptadora

$$q_0 \xrightarrow{u_1,v_1|w_1} q_1 \xrightarrow{u_2,v_2|w_2} q_2 \xrightarrow{u_3,v_3|w_3} q_3 \cdots$$

tal que $x = u_1u_2u_3\cdots$ e $y = v_1v_2v_3\ldots$ como

$$\rho_{\mathcal{A}}(x/y) = \liminf_{n\to\infty} \frac{|w_1w_2w_3\cdots|}{|u_1u_2u_3\cdots|}.$$

En el caso que la cinta de entrada y la de salida tuvieran alfabetos A y B respectivamente de diferentes tamaños, la formula anterior debe ser multiplicada por $\log|A|/\log|B|$. Notar que la cantidad de símbolos de y leídos, es decir $|v_1v_2v_3\cdots|$, no es considerada para la definición de $\rho_{\mathcal{A}}(x/y)$.

La *relación de compresión condicional* de una palabra infinita x con respecto a una palabra infinita y, $\rho(x/y)$, es el ínfimo de las relaciones de compresión $\rho_{\mathcal{A}}(x/y)$ para todos los compresores $\mathcal{A}$ con entrada x y oráculo y

Definición ([11])**.** Dos palabras infinitas x e y, posiblemente sobre diferentes alfabetos, son *independientes de estado finito* si $\rho(x/y) = \rho(x)$, $\rho(y/x) = \rho(y)$ y las relaciones de compresión de x e y no son cero.

En lo que sigue, en lugar de escribir *independencia de estado finito* simplemente escribimos *independencia*.

Notar que las relaciones de compresión de x e y deben ser distintas de 0. Esto significa que una palabra x tal que $\rho(x) = 0$ no es independiente de ninguna palabra. Sin este requisito, dos palabras x e y tal que $\rho(x) = \rho(y) = 0$ serían independientes. En particular, cada palabra x con $\rho(x) = 0$ sería independiente de sí misma.

De la definición de independencia se deduce que si las palabras infinitas x e y son independientes, cada sufijo de x es independiente de cada sufijo de y.

5.3. Enunciado del Teorema de Caracterización

5.3.1. Frecuencias de estados

Primero presentamos las definiciones para caracterizar independencia en términos de frecuencias de estados en corridas de 2-autómatas 2-deterministas con palabras normales como entrada.

Sea $\mathcal{A}$ un 2-autómata 2-determinista y sean x e y dos palabras infinitas, posiblemente sobre diferentes alfabetos. Sea γ la corrida de $\mathcal{A}$ sobre x e y

$$q_1 \xrightarrow{\bar{a}_1,\bar{b}_1} q_2 \xrightarrow{\bar{a}_2,\bar{b}_2} q_3 \xrightarrow{\bar{a}_3,\bar{b}_3} q_4 \cdots$$

donde cada $\bar{a}_i$ y cada $\bar{b}_i$ es un símbolo o la palabra vacía y cada $q_i \xrightarrow{\bar{a}_i,\bar{b}_i} q_{i+1}$ es una transición de $\mathcal{A}$.

Con algún abuso de notación, sea $|\gamma[1..n]|_q$ el número de apariciones del estado q en las primeras n transiciones de γ; es decir la cardinalidad del conjunto

$$\{i : 1 \leq i \leq n,\ q_i = q\}.$$

De igual manera, para cada transición $\tau = p \xrightarrow{\bar{a},\bar{b}} q$ sea $|\gamma[1..n]|_\tau$ el número de ocurrencias de τ en las primeras n transiciones de γ; es decir, la cardinalidad del conjunto

$$\{i : 1 \leq i \leq n,\ q_i \xrightarrow{\bar{a}_i,\bar{b}_i} q_{i+1} = \tau\}.$$

Asociamos con cada 2-autómata 2-determinista y 2-completo $\mathcal{A}$ una cadena de Markov descrita por una matriz estocástica M. Sean A y B los alfabetos para la primera y segunda cinta de $\mathcal{A}$. El conjunto de estados de la cadena de Markov es el conjunto de estados Q de $\mathcal{A}$. La dimensión de la matriz M es $|Q| \times |Q|$ y sus filas y columnas son indexadas por estados de Q. Para dos estados p y q, la entrada (p, q) de M es la suma de los pesos de todas las transiciones de p a q donde los pesos son los siguientes. El peso de una transición de la forma $p \xrightarrow{a,\lambda} q$ (respectivamente $p \xrightarrow{\lambda,b} q$) es $1/|A|$ (respectivamente $1/|B|$), mientras que el peso de una transición de la forma $p \xrightarrow{a,b} q$ es $1/(|A||B|)$.

Si el autómata $\mathcal{A}$ es fuertemente conexo entonces la cadena de Markov es irreducible. Por [52, Teorema 1.5], existe una única distribución estacionaria, es

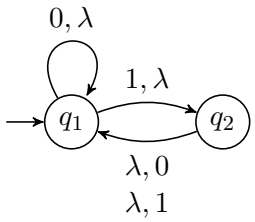

Figura 5.3: Un 2-autómata 2-determinista

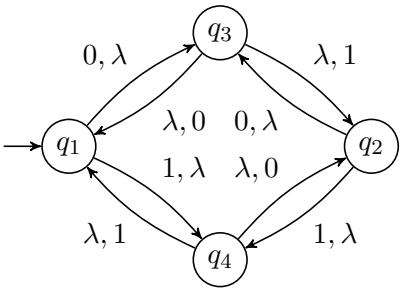

Figura 5.4: Otro 2-autómata 2-determinista

decir, un vector de línea π tal que $\pi M = \pi$ y $\sum_{q \in Q} \pi(q) = 1$. Por definición, se la llama *distribución estacionaria* asociada al autómata $\mathcal{A}$. Por ejemplo, la matriz de la cadena de Markov asociada con el 2-autómata de la Figura 5.3 es la siguiente

$$M = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix}$$

y la distribución estacionaria es $\pi(q_1) = 2/3$ y $\pi(q_2) = 1/3$.

El Teorema 5.1 establece que las frecuencias de estados en una corrida sobre palabras normales independientes viene dada por la distribución estacionaria asociada con el autómata. Esto significa que las frecuencias de los estados no dependen de las palabras de entrada. Esta afirmación es análoga a [51, Lema 4.5] pero para 2-autómatas 2-deterministas.

El siguiente ejemplo muestra que cuando se utilizan como entrada dos palabras normales pero no independientes, entonces la frecuencia de los estados en la ejecución depende de las palabras dadas. Considere el 2-autómata 2-determinista y 2-completo de la Figura 5.4. La matriz de la cadena de Markov asociada es

$$M = \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

y la distribución estacionaria está dada por $\pi(q_1) = \pi(q_2) = \pi(q_3) = \pi(q_4) = 1/4$. Si las palabras de entrada x e y son tales que $x = y$, la corrida nunca visita el estado q_2 y, por lo tanto, la frecuencia de este estado a lo largo de la corrida no es igual a $1/4$.

5.3.2. Selección

Presentamos la definición de selector que usamos para caracterizar independencia de palabras normales, que se dará en el Teorema 5.1. Dada una palabra infinita normal, el problema de selección consiste en hallar el modo de seleccionar símbolos de la palabra para que la palabra formada por los símbolos seleccionados satisfaga una cierta propiedad. Uno de los primeros resultados obtenidos por Wall [56] muestra que seleccionando los símbolos de una palabra normal en las posiciones dadas por una progresión aritmética vuelve a generar una palabra normal. Agafonov [1] extendió el resultado de Wall y demostró que cualquier selección mediante autómatas finitos preserva la normalidad (una prueba completa puede leerse en [15, Teorema de Agafonov] o se puede encontrar una versión más general en [10, Teorema 7.1]). Las selecciones admitidas por Agafonov deben ser realizadas por un 2-autómata 1-determinista imparcial. Imparcial significa que la decisión de seleccionar o no el siguiente símbolo solo depende del estado actual y no del siguiente símbolo.

Otras formas de selección mediante autómatas finitos no preservan normalidad. Por ejemplo [10, Theorem 7.3] muestra que la regla de selección bidireccional "seleccionar símbolos entre dos ceros", no preserva normalidad.

Para caracterizar la propiedad de independencia, consideramos la selección mediante un autómata finito sobre una palabra infinita, condicionado a otra palabra infinita que se puede usar en el proceso de selección como oráculo.

Definición. Un *selector* es un 3-autómata 2-determinista tal que cada una de sus transiciones es de alguno de los tipos $p \xrightarrow{a,\lambda|a} q$ (tipo I), $p \xrightarrow{a,\lambda|\lambda} q$ (tipo II), o $p \xrightarrow{\lambda,b|\lambda} q$ (tipo III) para dos símbolos $a, b \in A$. El selector es *imparcial* si todas las transiciones que comienzan en un estado dado tiene el mismo tipo.

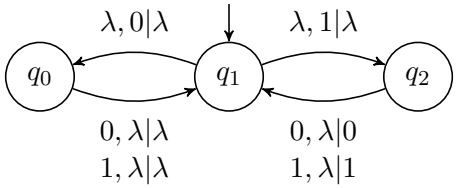

Figura 5.5: Un selector imparcial

Una transición de tipo $p \xrightarrow{a,\lambda|a} q$ (tipo I) copia un símbolo de la primera entrada x a la cinta de salida. Una transición de los tipos $p \xrightarrow{a,\lambda|\lambda} q$ (tipo II) o $p \xrightarrow{\lambda,b|\lambda} q$ (tipo III)

omite un símbolo de la primera entrada x o la segunda entrada y. Esto implica que la palabra de salida $z = \mathcal{S}(x, y)$ se obtiene seleccionando símbolos de x. Lo cual justifica la terminología.

Dado que un selector es 2-determinista, todas las transiciones que comienzan en un dado estado o bien son de tipo I y II, o son de tipo III. Cuando el selector es imparcial, no es posible que el mismo estado tenga transiciones de tipo I y II. La elección de copiar o no el símbolo actual de la cinta de entrada solo depende del estado y no del símbolo.

El autómata que se muestra en la Figura 5.5 es un selector imparcial. Selecciona símbolos de la primera entrada x que se encuentran en posiciones donde hay un 1 en la segunda entrada y.

5.3.3. Mezcladores

Presentamos la definición de mezclador que utilizamos para caracterizar independencia de palabras normales en el Teorema 5.1. Una palabra infinita z es una mezcla de x e y si se puede factorizar como $z = u_1v_1u_2v_2u_3\cdots$ donde las secuencias de palabras $(u_i)_{i\geq 1}$ y $(v_i)_{i\geq 1}$ satisfacen $x = u_1u_2u_3\cdots$ y $y = v_1v_2v_3\cdots$. Nos limitamos a la mezcla de palabras en el mismo alfabeto obtenido por 3-autómatas 2-deterministas. Probaremos que si x e y son palabras normales, x e y son independientes exactamente cuando cualquier mezcla de ellas también es normal. La mezcla de los símbolos de x e y debe ser realizado por un autómata determinista e imparcial que lee x e y. Aquí, imparcial, significa que la elección del símbolo a insertar en la mezcla z, solo depende del estado actual y no de los símbolos leídos actualmente en las cintas de entrada.

Definición. Un *mezclador* es un 3-autómata 2-determinista tal que cada una de sus transiciones es de la forma $p \xrightarrow{a,\lambda|a} q$ (tipo I) o de la forma $p \xrightarrow{\lambda,a|a} q$ (tipo II).

Notar que el determinismo de un mezclador $\mathcal{S}$ implica que para cada uno de sus estados p, todas las transiciones que salen de p tienen el mismo tipo. Una transición de tipo I copia un símbolo de la primera entrada x a la salida y una transición de tipo II copia un símbolo de la segunda entrada y a la salida. Entonces la tercera palabra $z = \mathcal{S}(x, y)$ se obtiene como mezcla de x e y. Esto justifica la terminología.

Considere las palabras normales $x = \overline{0011010001}\cdots$, $y = \underline{01000110001}\cdots$ y sea $\mathcal{S}$ el mezclador de la Figura 5.6. Entonces, la palabra infinita $z = \mathcal{S}(x, y)$ tiene

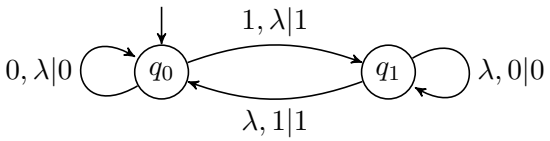

Figura 5.6: Un mezclador

la siguiente forma

$$z = \overline{00}\underline{1}\overline{01}\underline{\overline{1}}\underline{0001}\overline{0}\underline{\overline{1}}\underline{\overline{1}}\underline{\overline{0001}}\underline{0001}\cdots$$

agregamos líneas debajo o sobre los símbolos para marcar de qué palabra provienen.

5.3.4. Teorema de caracterización

Teorema 5.1 (Teorema de caracterización)**.** *Sean x e y dos palabras normales en los alfabetos A y B. Las siguientes afirmaciones son equivalentes.*

1. *Las palabras x e y son independientes.*
2. *Para todo 2-autómata 2-determinista y 2-completo fuertemente conexo $\mathcal{A}$ y para cada corrida infinita γ que tiene como entrada un sufijo de x y un sufijo de y, la frecuencia de cada estado q en γ es*

   $$\lim_{n\to\infty} \frac{|\gamma[1..n]|_q}{n} = \pi(q).$$

   donde π es la distribución estacionaria asociada con $\mathcal{A}$.
3. *Para todo selector imparcial $\mathcal{S}$, las salidas $\mathcal{S}(x,y)$ y $\mathcal{S}(y,x)$ también son normales.*

Además, si los alfabetos A y B son iguales, la siguiente afirmación también es equivalente.

4. *Para todo mezclador $\mathcal{S}$, el resultado de $\mathcal{S}(x,y)$ es también normal.*

Presentamos la prueba del teorema en la siguiente sección.

5.4. Prueba del Teorema de Caracterización

5.4.1. De independencia a frecuencia de estados

La afirmación (2) del Teorema 5.1 considera corridas sobre autómatas fuertemente conexos. Si el autómata no es fuertemente conexo, la corrida alcanzará una componente fuertemente conexa C maximal (que no tiene transiciones hacia

otras componentes), es decir, compuesta solamente de estados recurrentes según la terminología para cadenas de Markov. Por lo tanto, esta componente fuertemente conexa puede ser considerada como un 2-autómata 2-determinista. Además, si las palabras infinitas x e y son independientes, cada sufijo de x es independiente de cada sufijo de y. La afirmación (2) del Teorema 5.1 se puede aplicar a un sufijo de la corrida que solo visite estados de C. La frecuencia de ocurrencias de cada estado en C es dada por la distribución estacionaria de la cadena de Markov asociada con C.

El primer lema afirma que se puede suponer que los autómatas considerados obedecen cierta forma normal.

Lema 5.2. *Todo 2-autómata (2 -determinista) se puede transformar en otro que admite exactamente las mismas corridas infinitas y es tal que todas sus transiciones son de la forma $p \xrightarrow{a,\lambda} q$ o $p \xrightarrow{\lambda,b} q$ para algún par de símbolos a y b.*

Demostración. Introduciendo un nuevo estado q_a, las transiciones de la forma $p \xrightarrow{a,b} q$ pueden reemplazarse por dos transiciones $p \xrightarrow{a,\lambda} q_a$ y $q_a \xrightarrow{\lambda,b} q$. En cada corrida, la transición $p \xrightarrow{a,b} q$ será reemplazada por la corrida $p \xrightarrow{a,\lambda} q_a \xrightarrow{\lambda,b} q$. Además, si el autómata es 1-determinista, la transformación preserva esta característica. □

El siguiente lema da una relación entre la frecuencia de un estado y la frecuencia de las transiciones que comienzan en él. Es el primer paso hacia la caracterización a través de las frecuencias de los estados.

Lema 5.3. *Sea γ la corrida de un 2-autómata 2-determinista $\mathcal{A}$ sobre dos palabras normales independientes. Sea p un estado de $\mathcal{A}$ y σ y σ' dos transiciones que comienzan en p. Sea $(k_n)_{n\geq 0}$ una secuencia creciente de enteros tal que $\lim_{n\to\infty} |\gamma[1..k_n]|_p/k_n > 0$. Entonces*

$$\lim_{n\to\infty} \frac{|\gamma[1..k_n]|_\sigma}{|\gamma[1..k_n]|_p} = \lim_{n\to\infty} \frac{|\gamma[1..k_n]|_{\sigma'}}{|\gamma[1..k_n]|_p}.$$

Demostración. Para simplificar suponemos que A es el alfabeto binario $\{0,1\}$ pero la prueba puede extenderse fácilmente al caso general. Por el Lema 5.2, se puede suponer que cada transición de $\mathcal{A}$ es de la forma $p \xrightarrow{a,\lambda} q$ o $q \xrightarrow{\lambda,b} q$. En el resto de la prueba, las transiciones de la forma $p \xrightarrow{a,\lambda} q$ son llamadas de tipo I y las transiciones de la forma $p \xrightarrow{\lambda,b} q$ son llamadas de tipo II. Como el autómata es determinista, todos las transiciones que comienzan en cada estado q tienen el mismo tipo. Decimos que un estado q es de tipo I (respectivamente II) si todas las transiciones que comienzan en q son de tipo I (respectivamente II). Sin pérdida de generalidad, podemos suponer que todas las transiciones que comienzan en p, incluyendo σ y σ', son de tipo I.

Suponemos por contradicción que la igualdad requerida no se cumple y vamos a concluir en que x e y no son independientes. Probaremos que x puede ser comprimida dada y. Hay una falta de simetría entre x e y porque las transiciones σ y σ' son de tipo I. Al reemplazar $(k_n)_{n\geq 0}$ por una de sus subsecuencias, se puede asumir que, para toda transición τ, $\lim_{n\to\infty} |\gamma[1..k_n]|_\tau/k_n$ existe y que $\lim_{n\to\infty} |\gamma[1..k_n]|_\sigma/k_n \neq \lim_{n\to\infty} |\gamma[1..k_n]|_{\sigma'}/k_n$. Como la frecuencia de cada estado es igual a la suma de las frecuencias de las transiciones que comienzan en él, el límite $\lim_{n\to\infty} |\gamma[1..k_n]|_q/k_n$ existe para todo estado q. Llamamos a este límite $\pi(q)$.

Para cada transición τ que comienza en un estado q, se define $\pi(\tau)$ de la siguiente manera.

$$\pi(\tau) = \begin{cases} \lim\limits_{n\to\infty} \dfrac{|\gamma[1..k_n]|_\tau}{|\gamma[1..k_n]|_q} & \text{si } \lim_{n\to\infty} |\gamma[1..k_n]|_q/k_n \neq 0 \\ \dfrac{1}{2} & \text{en otro caso} \end{cases}$$

Dado que $\lim_{n\to\infty} |\gamma[1..k_n]|_\sigma/k_n \neq \lim_{n\to\infty} |\gamma[1..k_n]|_{\sigma'}/k_n$, $\pi(\sigma) \neq \pi(\sigma')$. Además, la siguiente igualdad es válida para todo estado q.

$$\sum_{\tau \text{ starts at } q} \pi(\tau) = 1.$$

Dado que x es normal, basta con mostrar que $\rho(x/y) < 1$. Sea ℓ una longitud de bloque que fijaremos posteriormente. Sea γ una corrida finita de longitud ℓ, es decir que γ es una secuencia $\tau_1\tau_2\cdots\tau_\ell$ de ℓ transiciones consecutivas. Definimos $\pi(\gamma)$ de la siguiente manera.

$$\pi(\gamma) = \begin{cases} \prod\limits_{\substack{\tau_i \text{ de tipo I} \\ 1\leq i\leq \ell}} \pi(\tau_i) & \text{si } \gamma \text{ tiene alguna transición de Tipo I} \\ 1 & \text{en otro caso} \end{cases}$$

Sea q un estado y $\bar{v}$ una palabra de longitud ℓ. Sea $\Gamma_{q,\bar{v}}$ el conjunto de corridas de longitud ℓ, comenzando en el estado q y que lee un prefijo de $\bar{v}$ en la segunda cinta,

$$\Gamma_{q,\bar{v}} = \{\gamma : \gamma = q \xrightarrow{u,v} q', v \sqsubset \bar{v}, |u| + |v| = \ell\}.$$

Notar que los conjuntos $\Gamma_{q,\bar{v}}$ no siempre son disjuntos. La palabra v leída de la segunda cinta durante la corrida γ puede ser prefijo de varias palabras $\bar{v}$. Si v es prefijo de $\bar{v}$ y $\bar{v}'$, entonces la corrida γ pertenece a $\Gamma_{q,\bar{v}}$ y $\Gamma_{q,\bar{v}'}$.

Veamos que para todo estado q y toda palabra $\bar{v}$,

$$\sum_{\gamma \in \Gamma_{q,\bar{v}}} \pi(\gamma) = 1.$$

Lo probamos por inducción en la longitud de la corrida finita, que llamamos ℓ. Si $\ell = 0$, la única corrida $\gamma \in \Gamma_{q,\bar{v}}$ es la corrida vacía, entonces $\pi(\gamma) = 1$. Ahora supongamos que $\ell \geq 1$. Distinguimos entre dos casos.

Primer caso: q es de tipo I. Sean $\tau_0 = q \xrightarrow{0,\lambda} q_0$ y $\tau_1 = q \xrightarrow{1,\lambda} q_1$ las dos transiciones que comienzan en el estado q. Y factorizamos $\bar{v}$ como $\bar{v} = \bar{v}'a$ donde $\bar{v}' = \bar{v}[1 \ldots \ell - 1]$ y a es el último símbolo de $\bar{v}$ El conjunto $\Gamma_{q,\bar{v}}$ es igual a la unión disjunta $\Gamma_{q,\bar{v}} = \tau_0\Gamma_{q_0,\bar{v}'} \cup \tau_1\Gamma_{q_1,\bar{v}'}$. El resultado es consecuencia de la hipótesis inductiva ya que $\pi(\Gamma_{q,\bar{v}}) = \pi(\tau_0)\pi(\Gamma_{q_0,\bar{v}'}) + \pi(\tau_1)\pi(\Gamma_{q_1,\bar{v}'}) = \pi(\tau_0) + \pi(\tau_1) = 1$.

Segundo caso: q es de tipo II. Factorizamos $\bar{v}$ como $\bar{v} = a\bar{v}'$, donde a es el primer símbolo de $\bar{v}$ y $\bar{v}' = \bar{v}[2 \ldots \ell - 1]$. La transición $\tau = q \xrightarrow{\lambda,a} q'$ es la primera de todas las corridas en $\Gamma_{q,\bar{v}}$ y $\Gamma_{q,\bar{v}} = \tau\Gamma_{q',\bar{v}'}$. El resultado es consecuencia de la hipótesis inductiva ya que $\pi(\Gamma_{q,\bar{v}}) = \pi(\Gamma_{q',\bar{v}'}) = 1$.

Dado que $\sum_{\gamma \in \Gamma_{q,\bar{v}}} \pi(\gamma) = 1$, existe, para cada estado q y cada palabra $\bar{v}$, un código libre de prefijos $P_{q,\bar{v}} = \{w_{\gamma,\bar{v}} : \gamma \in \Gamma_{q,\bar{v}}\}$ tal que $|w_{\gamma,\bar{v}}| \leq \lceil -\log \pi(\gamma) \rceil$ para toda corrida $\gamma \in \Gamma_{q,\bar{v}}$. Estas palabras pueden usarse para definir un compresor $\mathcal{C}$ que se ejecuta de la siguiente manera sobre dos entradas. Simula $\mathcal{A}$ y tiene ℓ símbolos de anticipación de la segunda cinta. Para cada corrida γ de longitud ℓ, el compresor produce $w_{\gamma,\bar{v}}$ en la tercera cinta. La elección de $w_{\gamma,\bar{v}}$ depende de la palabra de ℓ símbolos anticipados $\bar{v}$.

Finalmente mostramos que $\rho_{\mathcal{C}}(x/y) < 1$. La corrida γ de $\mathcal{A}$ sobre x e y puede ser factorizada como $\gamma = \gamma_1\gamma_2\gamma_3 \cdots$ donde cada corrida finita γ_i tiene longitud ℓ. La salida del compresor $\mathcal{C}$ es entonces $w_{\gamma_1,\bar{v}_1} w_{\gamma_2,\bar{v}_2} w_{\gamma_3,\bar{v}_3} \cdots$ donde las palabras $\bar{v}_1, \bar{v}_2, \bar{v}_3, \ldots$ están compuestas por los correspondientes ℓ símbolos de anticipación. Sean $\varepsilon, \delta > 0$ dos números reales positivos. Sea n un entero lo suficientemente grande como para que $|\gamma[1..k_n]|_\tau \leq (1+\delta)\pi(q)\pi(\tau)k_n$ para toda transición τ que comienza en q. Entonces,

$$\begin{aligned}
|w_{\gamma_1,\bar{v}_1} \cdots w_{\gamma_n,\bar{v}_n}| &\leq \sum_{i=1}^{n} \lceil -\log \pi(\gamma_i) \rceil \\
&\leq n + \sum_{i=1}^{n} -\log \pi(\gamma_i) \\
&\leq n + \sum_{\tau \text{ of type I}} |\gamma[1..\ell n]|_\tau \log \frac{1}{\pi(\tau)}
\end{aligned}$$

$$\leq \ell n \left[\frac{1}{\ell} + (1+\delta) \sum_{q \text{ of type I}} \pi(q) \sum_{\tau \text{ starts at } q} \pi(\tau) \log \frac{1}{\pi(\tau)} \right]$$

Luego, para todo estado estado q.

$$\sum_{\tau \text{ comienza en } q} \pi(\tau) \log \frac{1}{\pi(\tau)} \leq 1$$

y la desigualdad es estricta para $q = p$. Como $\pi(p) > 0$, para ε suficientemente pequeño, se pueden elegir δ y ℓ tal

$$\frac{1}{\ell} + (1+\delta) \sum_{q \text{ of type I}} \pi(q) \sum_{\tau \text{ starts at } q} \pi(\tau) \log \frac{1}{\pi(\tau)} \leq (1-\varepsilon) \sum_{q \text{ de tipo I}} \pi(q).$$

Y obtenemos

$$|w_{\gamma_1,\bar{v}_1} \cdots w_{\gamma_{k_n},\bar{v}_{k_n}}| \leq (1-\varepsilon)\ell k_n \sum_{q \text{ de tipo I}} \pi(q).$$

Dado que $\sum_{q \text{ de tipo I}} \pi(q)$ es el límite de la relación entre la cantidad de símbolos leídos de x y la longitud de la corrida, concluimos que $\rho_{\mathcal{C}}(x/y) < 1$. □

El siguiente lema muestra que, para un par de palabras normales x e y, si en la corrida de un 2-autómata 2-determinista, las transiciones que comienzan en cada estado q tienen la misma frecuencia. Entonces la frecuencia de cada estado es exactamente la mencionada en la afirmación (2) del Teorema 5.1.

Lema 5.4. *Sean x e y dos palabras normales tales que para todo 2-autómata 2-determinista $\mathcal{A}$ se cumple que si p es un estado de $\mathcal{A}$, σ y σ' son dos transiciones que comienzan en p y $(k_n)_{n\geq 0}$ es una secuencia creciente de naturales tal que $\lim_{n\to\infty} |\gamma[1..k_n]|_p/k_n > 0$ entonces*

$$\lim_{n\to\infty} \frac{|\gamma[1..k_n]|_\sigma}{|\gamma[1..k_n]|_p} = \lim_{n\to\infty} \frac{|\gamma[1..k_n]|_{\sigma'}}{|\gamma[1..k_n]|_p}.$$

Entonces la afirmación (2) del Teorema 5.1 es cierta.

Demostración. Sea $\mathcal{A}$ un 2-autómata 2-determinista fuertemente conexo Primero mostramos que si existe una corrida infinita γ en $\mathcal{A}$ sobre sufijos x' y y' de x e y, entonces $\mathcal{A}$ debe ser 2-completo. Supongamos por contradicción que $\mathcal{A}$ no es 2-completo. Digamos que $x = ux'$ y $y = vy'$. Al agregar algunos estados y transiciones a $\mathcal{A}$, construimos un 2-autómata determinista $\mathcal{A}'$ que primero lee u y v y luego se ejecuta como $\mathcal{A}$. La corrida de $\mathcal{A}'$ en x e y es igual a $\rho\gamma$ para alguna corrida finita ρ. Primero afirmamos que cada estado q en $\mathcal{A}$ tiene una frecuencia distinta de cero. Sea

P el subconjunto de estados $\{q : \liminf_{n\to\infty} \frac{|\gamma[1..k_n]|_p}{n} > 0\}$. La hipótesis implica que todos los estados alcanzables desde un estado en P también pertenecen a P. Dado que $\mathcal{A}$ es fuertemente conexo, todos los estados de $\mathcal{A}$ están en P. Ahora supongamos por contradicción que $\mathcal{A}$ no es completo. Se puede completar agregando transiciones que no se utilizan en la ejecución γ. Esto contradice la hipótesis.

Para probar la afirmación sobre las frecuencias de estados, es suficiente mostrar que para toda secuencia creciente de enteros $(k_n)_{n\ ge0}$ tal que $\lim_{n\to\infty} |\gamma[1..k_n]|_q/k_n$ existe, este límite es igual a $\pi(q)$. Sea $(k_n)_{n\geq 0}$ sea una secuencia de este tipo. Reemplacemos $(k_n)_{n\geq 0}$ por una de sus subsecuencias de modo que $\lim_{n\to\infty} |\gamma[1..k_n]|_q/k_n$ existe para todo estado q. Ya se ha demostrado en el párrafo anterior que estos límites no pueden ser 0.

Introducimos dos secuencias $(v_n)_{n\geq 0}$ y $(v'_n)_{n\geq 0}$ de vectores y una secuencia $(M_n)_{n\geq 0}$ de matrices. Para cada estado q, las entradas indexadas por q de los vectores están dadas por $v_n(q) = |\gamma[1..k_n]|_q/k_n$ y $v'_n(q) = |\gamma[2..k_n + 1]|_q/k_n$. Para cada par de estados p y q, la entrada (p, q) de M_n están dadas por $|\gamma[1..k_n]|_\tau/|\gamma[1..k_n]|_p$. Es fácil comprobar que $v_n M_n = v'_n$ vale para todo $n \geq 1$. Además, ambas secuencias $(v_n)_{n\geq 0}$ y $(v'_n)_{n\geq 0}$ convergen hacia el mismo vector $v(q) = \lim_{n\to\infty} |\gamma[1..k_n]|_q/k_n$. A partir la hipótesis, la secuencia $(M_n)_{n\geq 0}$ converge a la matriz M de la cadena de Markov asociada con $\mathcal{A}$. Tomando límites obtenemos $vM = v$. Por la unicidad de la distribución estacionaria de M, $v(q) = \pi(q)$ para todo estado q.

□

Prueba del Teorema 5.1, (1) implica (2). La prueba de que la afirmación (1) implica la afirmación (2) es consecuencia directa de los lemas 5.3 y 5.4. □

5.4.2. De frecuencias de estados a independencia

Para probar que la afirmación (2) implica la afirmación (1) en el Teorema 5.1 utilizaremos los siguientes lemas que consideran componentes fuertemente conexas de autómatas y distribuciones estacionarias. Una componente fuertemente conexa de un autómata se llama *final* si ninguna transición comienza en un estado de esta componente y finaliza en una componente distinta.

Lema 5.5. *Supongamos que se cumple la afirmación (2) del Teorema 5.1. Entonces, toda corrida infinita de un 2-autómata 2-determinista sobre x e y alcanza una componente fuertemente conexa final.*

Demostración. Supongamos que la corrida infinita γ en el 2-autómata 2-determinista $\mathcal{A}$ nunca alcanza una componente fuertemente conexa final. Entonces existe un sufijo

de la corrida que permanece en una componente fuertemente conexa no final C. Considere la restricción $\mathcal{A}'$ de $\mathcal{A}$ al conjunto de estados C. Como C no es final, $\mathcal{A}'$ no está completo. Esto es una contradicción. □

Lema 5.6. *Si $\mathcal{A}$ es fuertemente conexo, entonces la restricción de $\mathcal{A}_{k,\ell}$ al conjunto $Q \times A^k \times B^\ell$ también es fuertemente conexo.*

Demostración. Sean (q, u, v) y (q', u', v') dos estados en $Q \times A^k \times B^{ell}$. Existe una palabra w en $A^* \cup B^*$ y un estado r de $\mathcal{A}$ tal que $q \xrightarrow{uw,v} r$ o $q \xrightarrow{u,vw} r$ es una corrida finita en $\mathcal{A}$. Sin pérdida de generalidad, se puede suponer que $q \xrightarrow{uw,v} r$ es una corrida finita en $\mathcal{A}$. Como $\mathcal{A}$ es fuertemente conexo, existe una ejecución $r \xrightarrow{u'',v''} q'$. Entonces

$$(q, u, v) \xrightarrow{uwu''u',vv''v'} (q', u', v')$$

es una corrida en $\mathcal{A}_{k,\ell}$. □

Lema 5.7. *Si $\mathcal{A}$ es fuertemente conexo y π es su distribución estacionaria, entonces la distribución estacionaria de $\mathcal{A}_{k,\ell}$ viene dada por $\pi(q, u, v) = \pi(q)/|A|^k|B|^\ell$ para cada estado $(q, u, v) \in Q \times A^k \times B^\ell$.*

Demostración. Sea $M = (m_{p,q})$ la matriz $Q \times Q$ de $\mathcal{A}$. Cada entrada $m_{p,q}$ es igual a

$$|\{a : p \xrightarrow{a,\lambda} q\}|/|A| \text{ o } |\{b : p \xrightarrow{\lambda,b} q\}|/|B|$$

El vector π es el único vector que satisface $\pi M = \pi$ y $\sum_{q \in Q} \pi(q) = 1$. Sea (q, u, v) un estado fijo. Para cada transición $p \xrightarrow{a,\lambda} q$, existe una transición $(p, au', v) \xrightarrow{a',\lambda} (q, u, v)$ donde $u = u'a'$ (u' es el prefijo de longitud $k-1$ de u y a es su último símbolo. □

Prueba del Teorema 5.1, (2) implica (1). Sean x e y dos palabras normales tales que vale la afirmación (2) del Teorema 5.1. Vamos a probar que x e y son independientes. Es suficiente con probar que que x no puede ser comprimida con la ayuda de y, la prueba de que y no puede ser comprimida con la ayuda de x es la misma, intercambiando los roles de x e y.

Sea $\mathcal{C}$ un 3-autómata 2 determinista. Sea q_0 el estado inicial de $\mathcal{C}$. Sea γ la ejecución de $\mathcal{C}$ sobre x e y y sea z la salida de $\mathcal{C}$ en la corrida γ, es decir, $z = \mathcal{C}(x, y)$. Sea $\varepsilon > 0$ un número real positivo. Afirmamos que la relación de compresión $\rho_{\mathcal{C}}(x/y)$ satisface $\rho_{\mathcal{C}}(x/y) > 1 - \varepsilon$. Dado que esto es válido para todo $\varepsilon > 0$, se concluye que $\rho_{\mathcal{C}}(x/y) \geq 1$. Se puede asumir que $\mathcal{C}$ es fuertemente conexo. De lo contrario, la corrida sobre x e y en $\mathcal{C}$ alcanza, por Lema 5.5, una componente fuertemente conexa

de $\mathcal{C}$. Haciendo el mismo razonamiento con el sufijo de corrida en esta componente fuertemente conexa demostramos que los sufijos de x e y no son independientes, lo que implica que x e y tampoco son independientes.

Sea k un número entero positivo a ser fijado luego. Dado que y es normal, existe una constante $K > 0$ tal que si $u \sqsubset x$, $v \sqsubset y$ y $w \sqsubset z$ (u, v y w son prefijos de x, y y z respectivamente) tales que

$$q_0 \xrightarrow{u,v|w} q$$

entonces $|v| \leq K|u|$. La corrida γ se factoriza en

$$q_0 \xrightarrow{u_1,v_1|w_1} q_1 \xrightarrow{u_2,v_2|w_2} q_2 \xrightarrow{u_3,v_3|w_3} \cdots$$

donde $|u_i| = k$ para cada entero $i \geq 1$. Considerar que las longitudes de cada palabra v_i y cada palabra w_i son arbitrarias. Nuestro objetivo es demostrar que para N suficiente grande $|w_1 \cdots w_N| \geq (1-\varepsilon)|u_1 \cdots u_N|$.

Sea ℓ el entero $\lceil kK/\varepsilon \rceil$. Por definición de ℓ, la cardinalidad del conjunto $\{i \leq N : |v_i| > \ell\}$ es menor que εN. De lo contrario, tendríamos $|v_1 \cdots v_N| > K|u_1 \cdots u_N|$ que contradice la definición de la constante K.

Los índices i tales que $|v_i| > \ell$ serán ignorados en el resto de la prueba. Sea v_i' el prefijo de longitud ℓ de la palabra infinita $v_i v_{i+1} v_{i+2} \cdots$. A menos que $|v_i| > \ell$, v_i es un prefijo de v_i'. Sea $v' \in B^\ell$ una palabra fija de longitud ℓ. La cardinalidad del conjunto

$$X_{v'} = \{u \in A^k : \exists p, q \;\; p \xrightarrow{u,v|w} q, v \sqsubset v' \text{ y } |w| < (1-\varepsilon)k\}$$

está acotada por $|Q|^2|A|^{k(1-\varepsilon)}$. El entero k se elige de tal manera que $|A|^k - |Q|^2|A|^{k(1-\varepsilon)}$ sea mayor que $(1-\varepsilon)|A|^k$. Para distinguir estados que pueden ocurrir en la secuencia $(q_i)_{i\geq 0}$, introducimos un nuevo autómata $\mathcal{A}'$. Su conjunto de estado es $Q \times \{0, \ldots, k-1\}$ y sus transiciones se definen de la siguiente manera.

$$\begin{aligned}
&(q,i) \xrightarrow{a,\lambda|w} (q', i+1 \text{ mód } k) && \text{si } q \xrightarrow{a,\lambda|w} q' \text{ in } \mathcal{A} \\
&(q,i) \xrightarrow{\lambda,b|w} (q', i) && \text{si } q \xrightarrow{\lambda,b|w} q' \text{ in } \mathcal{A}
\end{aligned}$$

Notar que la distribución estacionaria del nuevo autómata $\mathcal{A}'$ no satisface la propiedad $\pi(q,i) = \pi(q)/k$ porque algunos estados podrían ser inalcanzables. Sin embargo $\sum_q \pi(q,i) = 1/k$ para todo $0 \leq i < k$.

Sea $\mathcal{A}$ un 2-autómata 2-determinista y sean k y ℓ dos enteros positivos. Introducimos un nuevo autómata $\mathcal{A}_{k,\ell}$. Su conjunto de estados es $Q \times A^{\leq k} \times \{\lambda\} \cup Q \times A^k \times B^{\leq \ell}$ y sus transiciones se definen de la siguiente manera.

$$\begin{aligned}
(q, u, \lambda) &\xrightarrow{a,\lambda} (q, ua, \lambda) && \text{si } |u| < k \\
(q, u, v) &\xrightarrow{\lambda,b} (q, u, vb) && \text{si } |u| = k \text{ y } |v| < \ell \\
(q, au', v) &\xrightarrow{a',\lambda} (q, u'a', v) && \text{si } |u'| = k-1,\ |v| = \ell \text{ y } q \xrightarrow{a,\lambda} q' \text{ in } \mathcal{A} \\
(q, u, bv') &\xrightarrow{\lambda,b'} (q, u, v'b') && \text{si } |u| = k,\ |v'| = \ell - 1 \text{ y } q \xrightarrow{\lambda,b} q' \text{ in } \mathcal{A}
\end{aligned}$$

Notar que los estados de $Q \times A^{\leq k} \times \{\lambda\} \cup Q \times A^k \times B^{<\ell}$ son claramente transitorios (en la terminología de cadenas de Markov). El propósito de estos estados es reunir los primeros k símbolos de x y los primeros ℓ símbolos de y para llegar al estado (q_0, u, v) donde q_0 es el estado inicial de $\mathcal{A}$ y u y v son los prefijos de x e y de longitud k y ℓ respectivamente. Por los lemas 5.6 y 5.7, la longitud de la salida durante la corrida γ es al menos $(1-\varepsilon)^4 kN$,

$$\begin{aligned}
\sum_{i=1}^{N} |w_i| &\geq (1-\varepsilon) \sum_{i=1, |v_i| \leq \ell}^{N} |w_i| \\
&\geq \frac{(1-\varepsilon)^2 N}{|A|^k |B|^\ell} \sum_{v_i \in B^\ell} \sum_{u_i \in A^k} |w_i| \\
&\geq \frac{(1-\varepsilon)^2 N}{|A|^k |B|^\ell} \sum_{v \in B^\ell} \sum_{u \in A^k} (1-\varepsilon) k \\
&\geq \frac{(1-\varepsilon)^2 N}{|A|^k |B|^\ell} \sum_{v \in B^\ell} (1-\varepsilon) |A|^k (1-\varepsilon) k \\
&\geq (1-\varepsilon)^4 kN.
\end{aligned}$$

□

5.4.3. Equivalencia entre independencia y propiedad de selección

Prueba del Teorema 5.1, (1) implica (3). Debemos probar que la selección de una palabra normal x con un oráculo normal e independiente y conserva la propiedad de normalidad. Mutatis mutandis, esta prueba es la misma que la dada en [10, Teorema 7.1], pero ahora se deben considerar 2-autómatas 3-deterministas y la palabra normal y como oráculo consultivo. □

Prueba del Teorema 5.1, (3) implica (1). Supongamos que x e y no son independientes. Como ya se ha demostrado que las afirmaciones (1)

y (2) son equivalentes, se puede suponer que la afirmación (2) no se cumple. Por el Lema 5.4, existe un autómata 2-determinista con la siguiente propiedad. Sea γ la corrida de $\mathcal{A}$ sobre x e y. Existe un estado p de $\mathcal{A}$, dos transiciones σ y σ' que comienzan en p, una secuencia creciente $\lim_{n\to\infty} |\gamma[1..k_n]|_p/k_n > 0$ y

$$\lim_{n\to\infty} \frac{|\gamma[1..k_n]||_\sigma}{|\gamma[1..k_n]||_p} \neq \lim_{n\to\infty} \frac{|\gamma[1..k_n]||_{\sigma'}}{|\gamma[1..k_n]||_p}.$$

Dado que $\mathcal{A}$ es 2-determinista, se puede suponer que todas las transiciones que comienzan en q leen símbolos desde la misma cinta. El autómata $\mathcal{A}$ se puede convertir en un selector de la siguiente manera. Las transiciones que comienzan en q seleccionan el dígito leído, mientra que todas las demás transiciones no seleccionan dígitos. La desigualdad anterior muestra que la salida del selector obtenido no es simplemente normal. Esta es una contradicción con la hipótesis.

□

Terminamos esta sección con el siguiente resultado que muestra que la independencia entre dos palabras normales implica la independencia de una y una selección mediante autómatas finitos de la otra.

Proposición 5.8. Sean x e y dos palabras normales e independientes. Si y' se obtiene mediante la selección imparcial de y, entonces x e y' son también independientes.

Demostración. Probaremos que si x e y' no son independientes, entonces x e y tampoco lo son. Suponemos que x e y' no son independientes. Esto significa que x se puede comprimir con la ayuda de y' o que y' se puede comprimir con la ayuda de x. Supongamos primero que x puede ser comprimido por un compresor $\mathcal{C}$ con la ayuda de y'. Combinando este compresor con el selector $\mathcal{S}$ que selecciona y' de y obtenemos un compresor $\mathcal{C}'$ que comprime x con la ayuda y. De hecho, este compresor $\mathcal{C}'$ ignora los símbolos de y que no son seleccionados por $\mathcal{S}$ y simula $\mathcal{C}$ sobre los símbolos que son seleccionados por $\mathcal{S}$.

Supongamos ahora que y' puede ser comprimida mediante un compresor $\mathcal{C}$ con la ayuda de x. Afirmamos que y también se puede comprimir con la ayuda de x. El selector $\mathcal{S}$ que selecciona y' de y se usa como separador para dividir y en dos palabras: y' compuesta por los símbolos seleccionados, e y'' compuesta por los símbolos no seleccionados. Entonces, el compresor $\mathcal{C}$ se usa para comprimir y' con la ayuda de x en una palabra z. Finalmente, las palabras z y y'' se combinan en una palabra z' usando bloques de la misma longitud m. Cada bloque de longitud m contiene m símbolos de de z o m símbolos de y'' y un símbolo adicional que indica si el bloque

contiene símbolos de z o símbolos de y''. La combinación de todos estos autómatas produce un compresor que comprime y con la ayuda de x. □

5.4.4. Equivalencia entre frecuencia de estados y propiedad de mezcladores

Si dos palabras normales x e y están en alfabetos diferentes, entonces su mezcla $\mathcal{S}(x, y)$ no es necesariamente normal. Por ejemplo, si x e y son palabras en diferentes alfabetos, su unión no es normal; recordar que la unión de dos palabras infinitas $x = a_1a_2a_3\cdots$ e $y = b_1b_2b_3\cdots$ es la palabra infinita $z = a_1b_1a_2b_2a_3\cdots$. Por lo tanto, asumimos que ambas palabras utilizan el mismo alfabeto.

Intercambiando las cintas de entrada y salida de un mezclador $\mathcal{S}$ obtenemos un 3-autómata 1-determinista al que llamamos *separador* correspondiente a $\mathcal{S}$. Si la salida $z = \mathcal{S}(x, y)$ del mezclador $\mathcal{S}$ sobre las entradas x e y se utiliza como entrada para el separador correspondiente, se obtienen las salidas x e y en sus respectivas cintas. El hecho de que el separador correspondiente sea 1-determinista produce el siguiente lema que efectivamente requiere que los alfabetos de las dos cintas sean iguales.

Lema 5.9. *Sea $\mathcal{S}$ sea un mezclador y q uno de sus estados. Para toda palabra finita w, hay exactamente una corrida de longitud $|w|$ que comienza en q y produce w.*

Prueba del Teorema 5.1, (2) implica (4). Supongamos que x e y son normales. Sea γ la corrida del mezclador $\mathcal{S}$ sobre las entradas x e y, y sea ℓ una longitud dada. Para todo estado q de $\mathcal{S}$ y toda palabra w de longitud ℓ, existe por Lema 5.9 una única corrida $\sigma_{q,w}$ que comienza en el estado q y que produce w.

Para toda palabra w de longitud ℓ, el número de apariciones de w en el prefijo $z[1..n]$ de z viene dado por

$$|z[1..n]|_w = \sum_{q \in Q} |\gamma[1..n]|_{\sigma_{q,w}}.$$

Veamos que la corrida γ alcanza una componente fuertemente conexa final. Supongamos por contradicción que la corrida permanece en una componente fuertemente conexa con al menos una transición saliente a otra componente. Sea $\mathcal{A}$ el autómata formado por esta componente fuertemente conexa sin las transiciones de salida. Entonces, no hay una distribución estacionaria asociada con $\mathcal{A}$ y esto es una contradicción con la afirmación (2). Por lo tanto, se puede asumir sin pérdida de generalidad que $\mathcal{S}$ es fuertemente conexo. La proporción $|z[1..n]|_w/n$ está dada

por

$$\frac{|z[1..n]|_w}{n} = \sum_{q \in Q} \frac{|\gamma[1..n]|_q}{n} \frac{|\gamma[1..n]|_{\sigma_{q,w}}}{|\gamma[1..n]|_q}.$$

Mostremos que para cualquiera par de corridas finitas σ y σ' de una misma longitud ℓ que comienzan en un mismo estado q,

$$\lim_{n\to\infty} \frac{|\gamma[1..k_n]|_\sigma}{|\gamma[1..k_n]|_p} = \lim_{n\to\infty} \frac{|\gamma[1..k_n]|_{\sigma'}}{|\gamma[1..k_n]|_p}.$$

Consideremos el autómata $\mathcal{S}_\ell$ definido de la siguiente manera. El conjunto de estados de $\mathcal{S}_\ell$ es el conjunto de corridas de longitud ℓ en $\mathcal{S}$. Hay una transición de una corrida σ a una corrida σ' en $\mathcal{S}_\ell$ si existe una transición τ en $\mathcal{S}$ tal que σ' es el sufijo de longitud ℓ de $\sigma\tau$. Un cálculo sencillo muestra que la distribución estacionaria de $mathcalS_\ell$ está dada por $\pi_\ell(\sigma) = \pi(q)/2^\ell$ donde π es la distribución estacionaria de $\mathcal{S}$. Aplicando la afirmación (2) a $\mathcal{S}_\ell$ se obtiene la igualdad. Por lo tanto, la relación $|z[1..n]|_w/n$ también tiene un límite que depende de w. Dado que esto es válido para toda longitud ℓ, la palabra infinita z es normal. □

Prueba del Teorema 5.1, (4) implica (1). Supongamos
que x e y no son independientes y x es compresible con la ayuda de y. Sea $\mathcal{A}$ el compresor tal que $\rho_{mathcalA}(x/y) < \rho(x)$. Consideremos el mezclador $\mathcal{S}$ que imita $\mathcal{A}$ y copia cada dígito de x (respectivamente de y) en el momento que es leído por $\mathcal{A}$. Afirmamos que $\mathcal{S}(x, y)$ es compresible, por lo tanto, no es normal. Para comprimir $\mathcal{S}(x, y)$, primero definimos un separado $\mathcal{S}'$ intercambiando las entradas y salidas del mezclador $\mathcal{S}$. Luego, $\mathcal{S}'(\mathcal{S}(x, y)) = (x, y)$. Al componer $\mathcal{S}'$ con $\mathcal{A}$ podemos comprimir x usando y y obtener una palabra comprimida x'. Sea m sea el tamaño de bloque utilizado en esta compresión. Finalmente, las palabras y y x' se combinan en una palabra z intercalando un bloque de m símbolos de x con un bloque de m símbolos de y. Como la hipótesis asegura que x es compresible, también lo es la palabra z. A partir de esta palabra z podemos obtener (x', y), de la cual podemos obtener (x, y) y luego computar $S(x, y)$.

□

Capítulo 6

Construcción de palabras independientes

6.1. Introducción

El resultado principal de este capítulo, expresado en el Teorema 6.1, da un algoritmo para construir un par de palabras normales independientes de estado finito, basado en la caracterización en términos de mezcladores que se prueba en el Teorema 5.1. Este algoritmo genera un par de palabras normales independientes de estado finito (x, y); agregando, en cada paso, un nuevo símbolo que extiende alternativamente el prefijo de x computado o el prefijo de y computado. Lamentablemente, la complejidad computacional de este algoritmo es doblemente exponencial, lo que significa que para obtener el n-ésimo símbolo del par de palabras el algoritmo realiza un número de operaciones que es doblemente exponencial en n.

Nuestra construcción de un par de palabras normales independientes es similar la construcción de secuencias que representan la expansión fraccionaria de números absolutamente normales (un número es absolutamente normal si su expansión en cada base entera mayor o igual a 2 es una palabra normal). El algoritmo presentado aquí es una variante del algoritmo de Turing para el cómputo de números absolutamente normales [54, 13], que también tiene complejidad computacional doblemente exponencial.

Hubiéramos obtenido un algoritmo muy parecido si en lugar del algoritmo de Turing hubiéramos considerado la versión recursiva del algoritmo de Siepiński [12] cuya complejidad es también doblemente exponencial,

Sin embargo no hemos podido adaptar ninguno de los algoritmos de generación de números absolutamente normales que son más eficientes. En particular no es nada evidente cómo adaptar los algoritmos basados en herramientas analíticas

como los de Schmidt [50], Levin [38, 5], o el más reciente de Aistleitner, Becher, Scheerer y Slaman [2], que tienen complejidad simplemente exponencial. Tampoco logramos adaptar los algoritmos discretos modernos como la variante del algoritmo de Turing con complejidad casi cuadrática de Becher, Heiber y Slaman [16], o el algoritmo polinomial basado en martingalas de Figueira y Nies [29] o su versión poli-logarítmicamente lineal dada por Lutz y Mayordomo [41].

El problema de dar un algoritmo polinomial para generar dos palabras normales independientes permanece abierto. Tampoco hemos conseguido dar un algoritmo que dada una secuencia normal genere otra normal independiente.

6.2. Algoritmo

Teorema 6.1. *Para todo alfabeto A, hay un algoritmo que calcula un par de palabras normales independientes en el alfabeto A.*

Para probar el Teorema 6.1 damos un algoritmo explícito basado en la caracterización de palabras normales independientes en términos de mezcladores (afirmación (4) de Teorema 5.1). El algoritmo que presentamos aquí es una adaptación del algoritmo de Turing para calcular un número absolutamente normal [54, 13]. Pero en lugar de computar la expansión de un número que es normal en toda base entera, computamos un par de palabras infinitas normales de modo que cada mezcla de ellas sea normal. Comenzamos con definiciones auxiliares y algunas propiedades. Escribimos log para el logaritmo en base e y $\log_b$ para cualquier otra base b.

Definición. 1. Para un mezclador $\mathcal{S}$, un número real $\varepsilon > 0$, una palabra finita $\gamma \in A^*$, y un entero positivo n, definimos el conjunto

$$E_{\mathcal{S}}(\varepsilon, \gamma, n) = \left\{ (x, y) \in A^\omega \times A^\omega : \left| |\mathcal{S}(x, y)[1..n]|_\gamma - n/|A|^{|\gamma|} \right| < \varepsilon n \right\}.$$

2. Asumiendo una enumeración de todos los mezcladores $\mathcal{S}_1, \mathcal{S}_2, \ldots$, definimos el conjunto

$$F(\varepsilon, t, \ell, n) = \bigcap_{i=1}^{t} \bigcap_{r=1}^{\ell} \bigcap_{\gamma \in A^r} E_{\mathcal{S}_i}(\varepsilon, \gamma, n).$$

3. Para cada entero positivo n, sea $\ell_n = (\log_{|A|} n)/3$, $t_n = n$ y $\varepsilon_n = 2\sqrt{(\log n \log_{|A|} n)/n}$.

$$F_n = F(\varepsilon_n, t_n, \ell_n, n).$$

Lema 6.2 (Lema 8 en [54], adaptado del Teorema 148 en [32])**.** *Sean r y n enteros positivos. Para todo ε real tal que $6/\lfloor n/r \rfloor \leq \varepsilon \leq 1/|A|^r$ y para todo $\gamma \in A^r$, si $N(\gamma, i, n) = |\{w \in A^n : |w|_\gamma = i\}|$ entonces*

$$\sum_{0 \leq i \leq n/|A|^r - \varepsilon n} N(\gamma, i, n) + \sum_{n/|A|^r + \varepsilon n \leq i \leq n} N(\gamma, i, n) < 2|A|^{n+2r-2} r e^{-|A|^r \varepsilon^2 n/6r}.$$

Para una palabra $u \in A^*$ escribimos $[u]$ para denotar el conjunto de palabras infinitas que comienzan con u, y lo llamamos el cilindro determinado por u.

$$[u] = \{x \in A^\omega : x[1..|u|] = u\}.$$

Notamos con $[u] \times [v]$ al producto cartesiano de dos cilindros, y lo llamamos par de cilindros determinados por (u, v).

Proposición 6.3. Para todo mezclador $\mathcal{S}$, $n, r \in \mathbb{N}$, $\varepsilon > \mathbb{R}^{>0}$ tales que $6/\lfloor n/r \rfloor \leq \varepsilon \leq 1/|A|^r$ y todo $\gamma \in A^r$,

$$\mu(E_{\mathcal{S}}(\varepsilon, \gamma, n)) > 1 - 2|A|^{2r-2} r e^{-|A|^r \varepsilon^2 n/6r}.$$

Demostración. Consideremos el conjunto

$$P(\varepsilon, \gamma, n) = \left\{ w \in A^n : \left| |w|_\gamma - n/|A|^{|\gamma|} \right| < \varepsilon n \right\}.$$

Entonces,

$$\begin{aligned} E_{\mathcal{S}}(\varepsilon, \gamma, n) &= \bigcup_{w \in P(\varepsilon,\gamma,n)} \{([u], [v]) : |u| + |v| = n \text{ y } \forall x \in [u] \forall y \in [v],\ \mathcal{S}(x, y) \in [w]\} \\ &= \bigcup_{w \in P(\varepsilon,\gamma,n)} \mathcal{S}^{-1}([w]). \end{aligned}$$

Por lo tanto,

$$\mu(E_{\mathcal{S}}(\varepsilon, \gamma, n)) = \sum_{w \in P(\varepsilon,\gamma,n)} \mu(\mathcal{S}^{-1}([w])) = |P(\varepsilon, \gamma, n)|\ |A|^{-n}.$$

Finalmente, el Lema 6.2 nos brinda la cota superior requerida para $|\overline{P}(\varepsilon, \gamma, n)|$. □

Para cualquier conjunto B, notamos con $\overline{B}$ su complemento en $A^\omega \times A^\omega$.

Proposición 6.4. Para todo ε, t, ℓ y n, tales que $6/\lfloor n/\ell \rfloor \leq \varepsilon \leq 1/|A|^\ell$,

$$\mu(F(\varepsilon, t, \ell, n)) > 1 - 2t|A|^{3\ell-1} e^{-\varepsilon^2 n/(3\ell)}.$$

Demostración. Por la definición 6.2,

$$\mu(\overline{F}(\varepsilon,t,\ell,n)) \leq \sum_{i=1}^{t}\sum_{r=1}^{\ell}\sum_{\gamma\in A^r}\mu(\overline{E_{\mathcal{S}_i}}(\varepsilon,\gamma,n)).$$

El número de términos de esta suma triple está acotado por

$$\sum_{i=1}^{t}\sum_{r=1}^{\ell}\sum_{\gamma\in A^r}1 = \sum_{i=1}^{t}\sum_{r=1}^{\ell}|A|^r < \sum_{i=1}^{t}\frac{|A|^{\ell+1}-1}{|A|-1} < \sum_{i=1}^{t}|A|^{\ell+1} = t|A|^{\ell+1}.$$

A partir de la cota inferior dada en la Proposición 6.3 obtenemos que para todo mezclador $\mathcal{S}$ y toda palabra $\gamma \in A^{\leq\ell}$,

$$\mu(\overline{E_{\mathcal{S}}}(\varepsilon,\gamma,n)) < 2|A|^{2\ell-2}\ell e^{-\varepsilon^2 n/(3\ell)}.$$

Entonces,

$$\mu(\overline{F}(\varepsilon,t,\ell,n)) < 2t|A|^{3\ell-1}e^{-\varepsilon^2 n/(3\ell)}. \qquad \square$$

Recordemos los valores dados en la Definición 6.2, $\ell_n = (\log_{|A|} n)/3$, $t_n = n$, $\varepsilon_n = 2\sqrt{(\log n \log_{|A|} n)/n}$ y $F_n = F(\varepsilon_n, t_n, \ell_n, n)$.

Proposición 6.5. Sea $n_{start} = \min\{n : \varepsilon_n \geq 6/\lfloor n/\ell_n\rfloor\}$. Entonces para todo $n \geq n_{start}$, $\ell_n, t_n \geq 1$,

$$\mu(F_n) \geq 1 - 1/n^2.$$

Demostración.
Para poder aplicar la Proposición 6.4 es necesario que $6/\lfloor n/\ell_n\rfloor \leq \varepsilon_n \leq 1/|A|^{\ell_n}$. Entonces para todo $n \geq n_{start}$ se cumple la desigualdad requerida y aplicando la Proposición 6.4 obtenemos

$$\begin{aligned}
\mu(\overline{F_n}) &\leq 2t_n|A|^{3\ell_n-1}e^{-\varepsilon^2 n/(3\ell_n)} \\
&\leq t_n\ |A|^{3\ell_n}e^{-\varepsilon^2 n/(3\ell_n)} \\
&= n|A|^{(\log_{|A|} n)}e^{-4n(\log n)(\log_{|A|} n)/(n\log_{|A|} n)} \\
&= n^2\ e^{-4\log n} \\
&= \frac{1}{n^2}. \qquad \square
\end{aligned}$$

Si tomamos n_{start} como fue definido en la Proposición 6.5, entonces $\bigcap_{n\geq n_{start}} F_n$ es no vacío y contiene únicamente pares de palabras normales independientes.

De hecho, podemos mostrar que la intersección de una subsecuencia de F_n's con n creciendo a lo sumo exponencialmente, también contiene únicamente pares de palabras normales independientes. La siguiente definición fija un n_0 como $\log n_{start}$ y define los conjuntos G_n que se usarán en la prueba del Teorema 6.1.

Definición. Sea $n_0 = \log_{|A|} \min\{n : \varepsilon_n \geq 6/\lfloor n/\ell_n \rfloor\}$. Definimos una secuencia $(G_n)_{n\geq 0}$ de conjuntos finitos de pares de cilindros en $A^\omega \times A^\omega$, tales que para todo n, $G_{n+1} \subseteq G_n$ como

$$G_n = \bigcap_{j=0}^{n} F_{|A|^{n_0+j}}$$

Lema 6.6. *El conjunto $\bigcap_{n\geq 0} G_n$ contiene exclusivamente pares de palabras independientes.*

Demostración. Tomemos el n_0 de la Definición 6.2. Supongamos que $(x, y) \in \bigcap_{n\geq 0} G_n$. Para demostrar que x e y son independientes, mostramos que para todo mezclador $\mathcal{S}$, $\mathcal{S}(x, y)$ es una secuencia normal. Fijemos una palabra finita $w \in A^*$. Elegimos m_0 tal que si i es el índice de $\mathcal{S}$ en la enumeración de mezcladores, $t_{m_0} \geq i$, $\ell_{m_0} \geq |w|$, $m_0 \geq n_0$ y $\varepsilon_{m_0} < 1/|A|^{|w|}$.

Veamos que para cualquier m mayor que m_0 se cumple lo siguiente. Sea k el entero tal que $|A|^k \leq m < |A|^{k+1}$. Luego, sabiendo que $(x, y) \in F_{|A|^{k+1}}$,

$$\begin{aligned}
\frac{|\mathcal{S}(x,y)[1..m]|_w}{m} &< \frac{|\mathcal{S}(x,y)[1..|A|^{k+1}]|_w}{m} \\
&< \frac{1}{m}|A|^{k+1}\left(\frac{1}{|A|^{|w|}} + \varepsilon_{m_0}\right) \\
&\leq \frac{|A|^{k+1}}{|A|^k}\frac{2}{|A|^{|w|}} \\
&= \frac{2|A|}{|A|^{|w|}}.
\end{aligned}$$

Esto implica que

$$\limsup_{m\to\infty} \frac{|\mathcal{S}_i(x,y)[1..m]|_w}{m} < \frac{2|A|}{|A|^{|w|}}.$$

Concluimos que $\mathcal{S}(x, y)$ es normal al aplicar el Teorema 4.6 de [20] que establece que una palabra x es normal si y solo si existe un número positivo C tal que para toda palabra finita w,

$$\limsup_{m\to\infty} \frac{|x[1..m]|_w}{m} \leq \frac{C}{|A|^{|w|}}.$$

Por lo tanto, tomando C igual a $2|A|$ mostramos que $S(x, y)$ es normal.

Ahora demostramos que ambas palabras x e y son también normales. Considere el selector $\mathcal{S}'$ definido como el separador que invierte $\mathcal{S}$ y luego ignora la segunda cinta de salida. Es decir, si $\mathcal{S}(x, y) = z$ entonces $\mathcal{S}'(z) = x$. Usando el Teorema de Agafonov (ver [15]), que establece que toda selección mediante autómatas finitos preserva normalidad, y dado que x se obtiene a partir de la palabra normal $S(x, y)$ mediante el selector de estado finito S', concluimos que x es normal. Un argumento similar demuestra que y también es normal. Demostramos que todo par $(x, y) \in \bigcap_{n\geq 0} G_n$ es un par de palabras normales que satisfacen la afirmación (4) del Teorema 5.1 y por lo tanto son independientes. □

6.3. Prueba de correctitud del algoritmo

Prueba del Teorema 6.1. Para mayor claridad, presentamos la prueba usando el alfabeto $A = \{0, 1\}$, por lo tanto $|A| = 2$. Es fácil generalizar la prueba a cualquier alfabeto de tamaño arbitrario. Probamos que el Algoritmo 7 construye un par de palabras normales independientes.

Del algoritmo es inmediato que la secuencia $(I_n)_{n\geq 0}$ tiene la propiedad que para todo n, $I_{n+1} \subset I_n$ y $\mu(I_{n+1}) = \mu(I_n)/2$.

Mostramos que para todo n, $\mu(I_n \cap G_n) > 0$. Probamos por inducción en n que,

$$\mu(G_n \cap I_n) > 2^{-2n-1}.$$

Para el caso base, $n = 0$, $\mu(G_0 \cap I_0) = 1 > 2^{-1}$.

Para el paso inductivo, dado que

$$\mu(\overline{F_{2^{n_0+n+1}}}) < \frac{1}{(2^{n_0+n+1})^2} = 2^{-2(n_0+n+1)} < 2^{-2(n+1)},$$

tenemos que

$$\begin{aligned}\mu(G_{n+1} \cap I_n) &= \mu(G_n \cap I_n \cap F_{2^{n_0+n+1}}) \\ &> 2^{-2n-1} - 2^{-2(n+1)} \\ &= 2^{-2(n+1)}.\end{aligned}$$

Entonces, al menos uno de los dos intervalos $G_{n+1} \cap I_n^0$ o $G_{n+1} \cap I_n^1$ debe tener medida mayor a $2^{-2(n+1)-1}$.

Prueba del Teorema 6.1

Algoritmo: Construcción de un par de palabras normales independientes
Entrada: Nada
Salida: Una secuencia $I_n = [u_n] \times [v_n]_{n\geq 0}$, tal que $u_n, v_n \in \{0,1\}^*$, $|u_n| + |v_n| = n$ y $\bigcap_{i\geq 0} I_n$ contiene un único par de palabras normales independientes (x, y).

Sea $\mathcal{S}_1, \mathcal{S}_2, \ldots$ una enumeración de los autómatas mezcladores.
Para cada $n \geq 1$, sea $\ell_n = (\log n)/3$, $\varepsilon_n = 2\sqrt{(\log n \log_2 n)/n}$ y

$$F_n = \bigcap_{i=1}^{n} \bigcap_{\gamma \in 2^{\leq \ell_n}} E_{\mathcal{S}_i}(\varepsilon_n, \gamma, n), \text{ donde}$$

$$E_{\mathcal{S}_i}(\varepsilon_n, \gamma, n) = \{(x,y) \in \{0,1\}^\omega \times \{0,1\}^\omega : \left| \, |\mathcal{S}_i(x,y)[1..n]|_\gamma - n/2^{|\gamma|} \right| < n\varepsilon_n\}.$$

Sea $n_0 = \log_2 \text{mín}\{n : \varepsilon_n \geq 6/\lfloor n/\ell_n \rfloor\}$. Notamos con λ a la palabra vacía.

begin
  $n \leftarrow 0$
  $I_0 \leftarrow ([\lambda], [\lambda])$
  $G_0 \leftarrow ([\lambda], [\lambda])$
  repetir para siempre
    $([u_n], [v_n]) \leftarrow I_n$
    si *n es par* **entonces**
      $I_n^0 \leftarrow ([u_n 0], [v_n])$
      $I_n^1 \leftarrow ([u_n 1], [v_n])$
    si no
      $I_n^0 \leftarrow ([u_n], [v_n 0])$
      $I_n^1 \leftarrow ([u_n], [v_n 1])$
    $G_{n+1} \leftarrow G_n \cap F_{2^{n_0+n+1}}$;
    si $\mu(I_n^0 \cap G_{n+1}) > 2^{-2n+1}$ **entonces**
      $I_{n+1} \leftarrow I_n^0$
    si no
      $I_{n+1} \leftarrow I_n^1$
    imprimir I_{n+1}
    $n \leftarrow n + 1$
end

Algoritmo 7: Construcción de un par de palabras normales independientes usando autómatas mezcladores

Dado que $(I_n)_{n\geq 0}$ es una secuencia anidada de intervalos de medida positiva pero estrictamente decreciente, concluimos que

$$\bigcap_{n\geq 0} I_n = \bigcap_{n\geq 0} G_n \cap I_n$$

contiene un único par (x, y).

Por el Lema 6.6, todos los elementos de $\bigcap_{n\geq 0} G_n$ son pares de palabras normales e independientes. Lo cual concluye la prueba. □

6.4. Complejidad computacional

El Algoritmo 7 calcula una secuencia $(I_n)_{n\geq 0}$ de pares de cilindros en $\{0,1\}^\omega \times \{0,1\}^\omega$ tal que $\bigcap_{i\geq 0} I_n$ contiene un único par (x, y) de palabras infinitas normales e independientes. Ahora estableceremos su complejidad computacional.

Proposición 6.8. El Algoritmo 7 tiene una complejidad doblemente exponencial: para computar los primeros n símbolos de salida realiza una cantidad de operaciones matemáticas que es doblemente exponencial en n.

Demostración. Como en la construcción original de Turing, la complejidad de cada paso de nuestro algoritmo está dominada por el cálculo del conjunto $F_{n_0+2^{2^n+1}}$, que es doblemente exponencial. Tener en cuenta que las medidas de los conjuntos inspeccionados se pueden calcular en tiempo simplemente exponencial, y el resto del cálculo lleva un tiempo constante.

La construcción toma una secuencia de "conjuntos buenos' $(G_n)_{n\geq 0}$ y una secuencia $(I_n)_{n\geq 0}$ de pares de cilindros en $\{0,1\}^\omega \times \{0,1\}^\omega$. Para el paso inicial, $n = 0$, $\mu(G_0) = 1, \mu(I_0) = 1$ y $\mu(G_0 \cap I_0) = 1$. Para los pasos siguientes, refinamos G_n en G_{n+1} y elegimos una mitad adecuada de I_n para que sea I_{n+1}. Ahora veamos la longitud del prefijo de las mezclas s_n que es necesario inspeccionar en el n-ésimo paso del algoritmo.

En el paso n, $G_{n+1} = G_n \cap F_{s_n}$ y $\mu(G_{n+1}) \geq \mu(G_n) - \mu(\overline{F_{s_n}})$. El algoritmo elige la mitad de I_n cuya intersección con G_{n+1} tiene una medida de al menos $(\mu(G_n) - \mu(\overline{F_{s_n}}))/2$. Necesitamos que esta medida sea positiva para todo n:

$$\big((((\mu(G_0) - \mu(\overline{F_{s_0}}))/2 - \mu(\overline{F_{s_1}}))/2 - \mu(\overline{F_{s_2}}))/2 \ldots - \mu(\overline{F_{s_{n-1}}})\big)/2 > 0$$

$$2^{-n} - 2^{-(n-1)}\mu(\overline{F_{s_0}}) - \ldots - 2^{-1}\mu(\overline{F_{s_{n-1}}}) > 0$$

Multiplicando por 2^n

$$1 - 2\mu(\overline{F_{s_0}}) - \ldots - 2^{n-1}\mu(\overline{F_{s_{n-1}}}) > 0$$

$$\sum_{n=1}^{\infty} 2^n \mu(\overline{F_{s_{n-1}}}) < 1.$$

Por lo tanto, necesitamos $\sum_{n=1}^{\infty} 2^n \mu(\overline{F_{s_{n-1}}}) < 1$ mientras que la Proposición 6.5 establece que $\mu(\overline{F_{s_{n-1}}}) < 1/s_{n-1}^2$. Luego, necesitamos que $s_{n-1} \geq 2^n$, esto muestra la necesidad del crecimiento exponencial de los índices s_n utilizados. Notar que el algoritmo fija $s_n = 2^{n+1}$ y el cálculo del conjunto F_{s_n} requiere la inspección de 2^{s_n} palabras de longitud s_n. Entonces en el paso n el algoritmo realiza una cantidad de operaciones que es doblemente exponencial en n. Finalmente, observar que en el paso n el algoritmo genera n símbolos en la forma de dos palabras u_n, v_n, tales que $|u_n| + |v_n| = n$. □

Bibliografía

[1] V. N. Agafonov. Normal sequences and finite automata. *Soviet Mathematics Doklady*, 9:324–325, 1968.

[2] C. Aistleitner, V. Becher, A.-M. Scheerer, and T. Slaman. On the construction of absolutely normal numbers. *Acta Arithmetica*, 180(4):333–346, 2017.

[3] J. Allouche and J. Shallit. *Automatic sequences*. Cambridge University Press, Cambridge, 2003. Theory, applications, generalizations.

[4] J. Almarza and S. Figueira. Normality in non-integer bases and polynomial time randomness. *Journal of Computer and System Sciences*, 81(7):1059–1087, 2015.

[5] N. Alvarez and V. Becher. M. Levin's construction of absolutely normal numbers with very low discrepancy. *Mathematics of Computation*, 86(308):2927–2946, 2017.

[6] N. Alvarez, V. Becher, P. Ferrari, and S. Yuhjtman. Perfect necklaces. *Advances in Applied Mathematics*, 80:48–61, 2016.

[7] É. Barbier. On suppose écrite la suite naturelle des nombres; quel est le $(10^{1000})^{\text{ième}}$ chiffre écrit? *Comptes Rendus des Séances de l'Académie des Sciences Paris*, 105:795–798, 1887.

[8] Ém. Barbier. On suppose écrite la suite naturelle des nombres; quel est le $(10^{10000})^{\text{ième}}$ chiffre écrit? *Comptes Rendus des Séances de l'Académie des Sciences Paris*, 105:1238–1239, 1887.

[9] V. Becher and O. Carton. Normal numbers and computer science. In Valérie Berthé and Michel Rigó, editors, *Sequences, Groups, and Number Theory*, Trends in Mathematics Series. Birkhauser/Springer, 2017.

[10] V. Becher, O. Carton, and P. A. Heiber. Normality and automata. *Journal of Computer and System Sciences*, 81(8):1592–1613, 2015.

[11] V. Becher, O. Carton, and P. A. Heiber. Finite-state independence. *Theory of Computing Systems*, page in press, 2017.

[12] V. Becher and S. Figueira. An example of a computable absolutely normal number. *Theoretical Computer Science*, 270:947–958, 2002.

[13] V. Becher, S. Figueira, and R. Picchi. Turing's unpublished algorithm for normal numbers. *Theoretical Computer Science*, 377(1-3):126–138, 2007.

[14] V. Becher and P. A. Heiber. On extending de Bruijn sequences. *Information Processing Letters*, 111(18):930–932, 2011.

[15] V. Becher and P. A. Heiber. Normal numbers and finite automata. *Theoretical Computer Science*, 477:109–116, 2013.

[16] V. Becher, P.A. Heiber, and T. Slaman. A polynomial-time algorithm for computing absolutely normal numbers. *Information and Computation*, 232:1–9, 2013.

[17] J. Berstel and D. Perrin. The origins of combinatorics on words. *European J. Combin.*, 28(3):996–1022, 2007.

[18] É. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27:247–271, 1909.

[19] J.M. Borwein and D.H. Bailey. *Mathematics by Experiment, 2nd Edition: Plausible Reasoning in the 21st Century*. Ak Peters Series. Taylor & Francis, 2004.

[20] Y. Bugeaud. *Distribution modulo one and Diophantine approximation*, volume 193 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2012.

[21] O. Carton and P. A. Heiber. Normality and two-way automata. *Information and Computation*, 241:264–276, 2015.

[22] J. W. S. Cassels. On a paper of niven and zuckerman. *Pacific Journal of Mathematics*, 2(4):555–557, 1952.

[23] D. Champernowne. The construction of decimals normal in the scale of ten. *J. London Math. Soc.*, s1-8(4):254–260, 1933.

[24] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of graphs*, volume 87 of *Pure and Applied Mathematics*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1980. Theory and application.

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

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

[27] N. G. de Bruijn. A combinatorial problem. *Nederl. Akad. Wetensch., Proc.*, 49:758–764 = Indagationes Math. 8, 461–467 (1946), 1946.

[28] R. G. Downey and D. R. Hirschfeldt. *Algorithmic randomness and complexity.* Theory and Applications of Computability. Springer, New York, 2010.

[29] S. Figueira and A. Nies. Feasible analysis, randomness, and base invariance. *Theory of Computing Systems*, 56:439–464, 2015.

[30] F. Harary. *Graph theory.* Addison-Wesley Publishing Co., Reading, Mass.-Menlo Park, Calif.-London, 1969.

[31] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers, 1st edition.* Oxford University Press, 1938.

[32] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* Oxford University Press, Oxford, sixth edition, 2008.

[33] D. Huffman. A method for the construction of minimum-redundancy codes. In *Institute of Radio Engineers*, pages 1098–1102, 1952.

[34] T. Kamae and B. Weiss. Normal numbers and selection rules. *Israel Journal of Mathematics*, 21(2):101–110, 1975.

[35] D. E. Knuth. *The Art of Computer Programming. Vol. 2.* Addison-Wesley, 1998. Seminumerical Algorithms. Third edition.

[36] P. L'Ecuyer and R. Simard. TestU01: a C library for empirical testing of random number generators. *ACM Trans. Math. Software*, 33(4):Art. 22, 40, 2007.

[37] E. L. Lehmann. *Fisher, Neyman, and the creation of classical statistics.* Springer, New York, 2011.

[38] M. Levin. On absolutely normal numbers. *Vestnik Moscov. Univ. ser. I, Mat-Meh*, 1:31–37, 87, 1979. English translation in Moscow Univ. Math. Bull., 34 (1979), no. 1, 32-39.

[39] M. Lothaire. *Combinatorics on words.* Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1997.

[40] M. Lothaire. *Algebraic combinatorics on words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002.

[41] J. Lutz and E. Mayordomo. Computing absolutely normal numbers in nearly linear time. Arxive 1611.05911, 2016.

[42] M. Madritsch, A. Scheerer, and R. Tichy. Computable absolutely pisot normal numbers. *arXiv preprint arXiv:1610.06388*, 2016.

[43] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[44] J.R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[45] W. Parry. Intrinsic markov chains. *Transactions of the American Mathematical Society*, 112(1):55–66, 1964.

[46] D. Perrin and J.-É. Pin. *Infinite Words*. Elsevier, 2004.

[47] I.I. Piatetski-Shapiro. On the distribution of the fractional parts of the exponential function. *Moskov. Gos. Ped. Inst. Uč. Zap.*, 108:317–322, 1957.

[48] J.-E. Pin. *Relational morphisms, transductions and operations on languages*, pages 34–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.

[49] J. Sakarovitch. *Elements of automata theory*. Cambridge University Press, 2009.

[50] Wolfgang Schmidt. Über die Normalität von Zahlen zu verschiedenen Basen. *Acta Arithmetica*, 7:299–309, 1961/1962.

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

[52] E. Seneta. *Non-negative Matrices ans Markov Chains*. Springer, 2006.

[53] D. Sheinwald. On the ziv-lempel proof and related topics. *Proceedings of the IEEE*, 82(6):866–871, 1994.

[54] A. Turing. A note on normal numbers. In J.L.Britton, editor, *Collected Works of A.M. Turing: Pure Mathematics*, pages 117–119. North Holland, Amsterdam, 1992. with notes of the editor in 263–265.

[55] W. T. Tutte. *Graph theory*, volume 21 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company, Advanced Book Program, Reading, MA, 1984.

[56] D. D. Wall. *Normal Numbers*. PhD thesis, University of California, Berkeley, California, 1949.

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