Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

De normalidad a incompresibilidad vía codificación aritmética

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Facundo López Bristot

Director: Pablo Ariel Heiber

Buenos Aires, 2012

DE NORMALIDAD A INCOMPRESIBILIDAD VÍA CODIFICACIÓN ARITMÉTICA

En este trabajo damos una prueba completa de la caracterización de las secuencias normales como aquellas incompresibles mediante compresores de estados finitos sin pérdida de información. Para esto definimos una familia de codificadores que utilizan la técnica de codificación aritmética y son producidos por autómatas finitos, mostramos que la incompresibilidad por compresores de estados finitos sin pérdida de información equivale a la incompresibilidad por codificadores aritméticos de estados finitos y que esta última a su vez equivale a la normalidad. Usando estos resultados obtenemos una prueba sencilla del teorema de Agafonov sobre la preservación de la normalidad en la selección de subsecuencias vía autómatas finitos.

Palabras claves: Números normales, Aleatoriedad, Autómatas finitos, Compresores de estados finitos, Codificación aritmética, Selectores de estados finitos.

FROM NORMALITY TO INCOMPRESSIBILITY VIA ARITHMETIC CODING

We give a complete proof of the characterization of normal sequences as those incompressible by lossless finite-state compressors. In order to do this we define a family of coders based on arithmetic coding which are produced by finite automata, then we show that incompressibility by lossless finite-state compressors is equivalent to incompressibility by finite-state arithmetic coders, which in turn is equivalent to normality. Using these results we obtain a simple proof of Agafonov's theorem on the preservation of normality when choosing subsequences by finite-state selectors.

Keywords: Normal numbers, Randomness, Finite automata, Finite-state compressors, Arithmetic Coding, Finite-state selectors.

AGRADECIMIENTOS

En primer lugar, deseo agradecer profundamente a Pablo Heiber por haberme dado la oportunidad de hacer la tesis bajo su dirección y por su dedicación en ese rol.

Agradezco a Verónica Becher por su alegría y entusiasmo contagiosos y por su labor como directora en el marco de la Beca de Estímulo a las Vocaciones Científicas.

Al CIN por haberme otorgado dicha beca.

A Joos Heintz y Santiago Figueira por haberse tomado el trabajo de leer y corregir esta tesis.

A Nicolás Rosner por su pronta colaboración en la traducción de [Sch73].

A mis compañeros de la facultad. No podría expresar todo mi agradecimiento hacia ellos en un simple párrafo. Agradezco especialmente a Luisina y Manuel, cuyo apoyo durante la tesis fue fundamental.

A mi familia por acompañarme incondicionalmente durante la carrera.

Por último, agradezco a Pedro, no sólo porque gracias a él conocí a Pablo y surgió la oportunidad de realizar este trabajo, sino también por su paciencia admirable y su aliento durante la tesis.

Índice general

INTRODUCCIÓN

Durante el siglo pasado se intentó capturar matemáticamente la noción intuitiva de aleatoriedad, descripta como una propiedad de las secuencias infinitas de símbolos [Eag12]. Borel definió en 1909 [Bor09] las secuencias normales como las que cumplen la ley de los grandes números o, equivalentemente, aquellas estadísticamente balanceadas, una propiedad esperada en una secuencia aleatoria. En ellas todo bloque de símbolos tiene la misma frecuencia que los demás de la misma longitud.

Sin embargo, la normalidad es una condición necesaria pero no suficiente para garantizar aleatoriedad. Champernowne exhibió en 1933 [Cha33] una secuencia normal computable, compuesta por la concatenación de las representaciones en base 10 de los números naturales. Al estar regida por un patrón, esta secuencia no debería ser clasificada como aleatoria.

Las propuestas de Martin-Löf [ML66] y Kolmogorov [LV08, Nie09] formuladas en las décadas de 1960 y 1970 no sólo satisfacen propiedades deseables en una definición de aleatoriedad, entre las que figura implicar normalidad, sino que Schnorr probó en 1973 [Sch73] que son equivalentes. Esto motivó a Delahaye a enunciar en 1993 [Del93], en una analogía con la Tesis de Church, la Tesis de Martin-Löf–Chaitin, según la cual estas definiciones son la formulación correcta del concepto intuitivo de aleatoriedad.

Los enfoques utilizados por quienes intentaron caracterizar la aleatoriedad fueron diversos. En su definición, Kolmogorov estableció que una secuencia es aleatoria cuando sus segmentos iniciales son incompresibles por compresores computables. Borel, en cambio, definió la normalidad desde un punto de vista combinatorio, observando la frecuencia de subcadenas en una secuencia.

Una segunda analogía puede trazarse al comprobar que la normalidad es definible en términos de incompresibilidad, restringiendo el universo de compresores a aquellos producidos por autómatas finitos. La equivalencia entre normalidad e incompresibilidad mediante compresores de estados finitos sin pérdida de información es un hecho conocido, pero su prueba se encuentra fragmentada en varias publicaciones. Por un lado, Dai, Lathrop, Lutz y Mayordomo introdujeron en 2004 [DLLM04] la noción de *dimensión de estados finitos* de secuencias y mostraron que coincide con la tasa de compresión de compresores de estados finitos, de modo que una secuencia es incompresible si y sólo si su dimensión es 1. Usando un teorema de Schnorr y Stimm de 1972 [SS72] probaron que las secuencias normales tienen dimensión 1, mientras que Bourke, Hitchcock y Vinodchandran probaron la recíproca en 2005 [BHV05]. Becher y Heiber [BH12] formularon en 2012 una prueba elemental y directa de la relación entre normalidad e incompresibilidad, evitando un paso intermedio como es el uso de dimensión de estados finitos. Como corolario de este resultado obtuvieron, a su vez, una demostración del teorema de Agafonov sobre la preservación de la normalidad en subsecuencias elegidas por medio de autómatas finitos, cuya exposición original en 1968 [Aga68] no está acompañada por una prueba completa.

En este trabajo damos otra prueba de la caracterización de la secuencias normales como aquellas incompresibles. En primer lugar presentamos una familia de compresores producidos por autómatas finitos que usan la técnica de codificación aritmética, similares en su construcción a las cadenas de Markov estacionarias y a las martingalas basadas en autómatas finitos. Tras una breve exposición de diferencias y similitudes entre estos codi-

ficadores y los compresores de estados finitos sin pérdida de información, deducimos que el conjunto de secuencias incompresibles es el mismo para ambos. Finalmente, probamos que las secuencias normales son exactamente aquellas incompresibles por nuestros codificadores aritméticos y usando este resultado construimos una prueba sencilla del teorema de Agafonov, en la cual se puede apreciar la conveniencia de los codificadores aquí presentados por sobre los compresores de estados finitos.

El resto de la presentación se organiza de la siguiente manera. En el capítulo 1 exponemos las nociones previas a este estudio con las que trabajaremos, que son la normalidad, la compresión y la selección con estados finitos y la codificación aritmética. Una vez que contamos con estos elementos, en el capítulo 2 definimos los codificadores aritméticos de estados finitos y estudiamos algunas de sus propiedades, comparándolos con los compresores de estados finitos. Finalmente, en el capítulo 3 probamos vía codificadores aritméticos de estados finitos la equivalencia entre normalidad e incompresibilidad mediante compresores de estados finitos sin pérdida de información y luego damos una demostración del teorema de Agafonov.

1. PRELIMINARES

Notación. Denotamos con $\mathcal{A}$ a un conjunto finito de al menos dos elementos que usamos como alfabeto. Los conjuntos de cadenas finitas e infinitas construidas con símbolos de $\mathcal{A}$ son $\mathcal{A}^*$ y $\mathcal{A}^\omega$, respectivamente. En general llamamos "cadenas" a los elementos del primer conjunto y "secuencias" a los del segundo. Escribimos λ para denotar la cadena vacía. El conjunto $\mathcal{A}^+$ contiene todas las cadenas excluyendo la vacía, es decir $\mathcal{A}^+ = \mathcal{A}^* \setminus \lambda$. Si k es un entero no negativo, $\mathcal{A}^{<k}$ y $\mathcal{A}^{\leq k}$ representan los conjuntos de cadenas en $\mathcal{A}^*$ que tienen menos de k símbolos, en el primer caso, y a lo sumo k símbolos, en el segundo. Para nombrar elementos particulares de estos conjuntos usamos letras del alfabeto latino comenzando en c en el caso de símbolos de $\mathcal{A}$ y en v si se trata de cadenas. Para nombrar secuencias usamos letras griegas comenzando en α. Si $v, w \in \mathcal{A}^*$ y $\alpha \in \mathcal{A}^\omega$, representamos con $vw \in \mathcal{A}^*$ y $v\alpha \in \mathcal{A}^\omega$ a la cadena y la secuencia obtenidas por concatenación de v con w y con α, respectivamente. La concatenación de una cadena consigo misma $n \in \mathbb{N}_0$ veces es v^n, con $v^{n+1} = vv^n$ y $v^0 = \lambda$, y v^ω es la secuencia que se obtiene concatenando infinitas veces v, $v^\omega = vv^\omega$. La longitud de una cadena w es $|w|$ e $|I|$ es la medida del intervalo real I. La expresión $w[i..j]$ representa la subcadena de w que comienza con el elemento i-ésimo y termina con el j-ésimo, inclusive, si $1 \leq i \leq j \leq |w|$, y si no denota la cadena vacía. De la misma manera $\alpha[i..j]$ es una subcadena de α que es igual a λ si $i > j$. Cuando escribimos $w[i]$ y $\alpha[i]$ nos referimos al i-ésimo elemento de aquella cadena o secuencia. Para contar la cantidad de apariciones de v en w usamos la función $\mathrm{occ} : \mathcal{A}^* \times \mathcal{A}^* \to \mathbb{N}$, con $\mathrm{occ}(v, w) = |\{i : v = w[i..i + |v| - 1]\}|$.

1.1. Normalidad

Aunque en este trabajo trataremos a la normalidad como una propiedad de las secuencias infinitas de símbolos de un alfabeto finito, es posible abordar la noción en términos de números reales. En efecto, cada número real r tiene una única secuencia infinita correspondiente a su desarrollo en base b luego de descartar las representaciones con período $b - 1$, es decir,

$$r = \lfloor r \rfloor + \sum_{j \geq 1} \frac{a_j}{b^j} = \lfloor r \rfloor + 0.a_1 a_2 \ldots,$$

donde $a_i \in \{0, 1, \ldots, b-1\}$ para todo i y existen infinitos índices j tales que a_j es distinto de $b - 1$. Las expresiones que definiremos a continuación, entonces, serán aplicables a un número real r si lo son para la secuencia $a_1 a_2 a_3 \ldots$.

La definición de normalidad que usaremos no es la original, formulada en 1909 por Émile Borel [Bor09], sino una equivalente utilizada por varios autores, entre quienes se encuentran Champernowne [Cha33] y Copeland y Erdös [CE46]. Puede consultarse la prueba de esta equivalencia en [Bug12] (Definición 4.1, páginas 87–88, y Teorema 4.5, páginas 91–93).

Definición. *Sea $\alpha \in \mathcal{A}^\omega$. La secuencia α es* simplemente normal *si y sólo si las frecuencias de los símbolos en α respetan la ley de los grandes números, es decir, tales frecuencias existen y coinciden. Para todo $c \in \mathcal{A}$ debe cumplirse*

$$\lim_{n \to \infty} \frac{\mathrm{occ}(c, \alpha[1..n])}{n} = |\mathcal{A}|^{-1}.$$

Una secuencia es normal *si y sólo si todos los bloques de símbolos de la misma longitud, cualquiera sea ésta, son igual de frecuentes en α en el límite. Formalmente, para todo $w \in \mathcal{A}^*$*

$$\lim_{n\to\infty} \frac{\mathrm{occ}(w, \alpha[1..n])}{n} = |\mathcal{A}|^{-|w|}.$$

Es esperable que sea normal una secuencia generada mediante la repetición infinita de un experimento cuyos resultados son equiprobables. Por el contrario, es posible que una secuencia normal no tenga el aspecto de una aleatoria. En 1933 Champernowne [Cha33] exhibió el primer ejemplo de secuencia normal, la que se construye escribiendo los números naturales en base 10, uno tras otro,

$$12345678910111213\ldots.$$

Esta secuencia no está libre de patrones, por lo que no debería ser considerada aleatoria. Efectivamente se trata de una secuencia computable, no aleatoria según la definición de Martin-Löf [ML66].

No se conocen ejemplos de secuencias normales que no hayan sido construidas específicamente, aunque se conjetura que constantes como e, π y $\sqrt{2}$ son *absolutamente normales* [BC01], es decir, sus desarrollos son normales para toda base de numeración. A pesar de esta elusividad, la mayoría de los números reales son absolutamente normales.

Teorema 1.1 (Teorema 4.8, [Bug12], páginas 94–95)**.** *La medida de Lebesgue del conjunto de los números reales del intervalo $[0, 1)$ que son absolutamente normales es 1.*

Para un natural arbitrario k, es posible vincular cada secuencia $\alpha \in \mathcal{A}^\omega$ con una secuencia $\beta \in (\mathcal{A}^k)^\omega$ tal que para todo i se verifique $\alpha[(i-1)k + 1..ik] = \beta[i]$. A lo largo de este trabajo llamaremos *cambio de alfabeto* a esta relación. Si $\mathcal{A} = \{0, 1, \ldots, n\}$ para algún n, un cambio de alfabeto es un cambio de base vía potenciación. Por ejemplo, si $\mathcal{A} = \{0, 1\}$, $k = 2$ y α es la secuencia que comienza con cero y siempre alterna dígitos, β es simplemente la repetición infinita del símbolo 01.

$$\alpha = 0\ 1\ 0\ 1\ 0\ 1\ldots \qquad \beta = 01\ 01\ 01\ldots$$

La normalidad es invariante bajo cambios de alfabeto.

Teorema 1.2 (Teorema 4.4, [Bug12], páginas 90–91)**.** *Sea k un número natural, $\alpha \in \mathcal{A}^\omega$ y $\beta \in (\mathcal{A}^k)^\omega$ obtenida a partir de α mediante un cambio de alfabeto. α es normal si y sólo si β es normal.*

Los cambios de alfabeto también son utilizados en otras caracterizaciones de la normalidad.

Teorema 1.3 (Teorema 4.2, [Bug12], páginas 88–89)**.** *Una secuencia $\alpha \in \mathcal{A}^\omega$ es normal si y sólo si para toda $k \in \mathbb{N}$ la secuencia $\alpha_k \in (\mathcal{A}^k)^\omega$, vinculada con α por un cambio de alfabeto, es simplemente normal.*

1.2. Compresores de estados finitos

Los compresores de estados finitos, definidos originalmente por Huffman en 1959 [Huf59], son autómatas finitos cuyas transiciones tienen asociada una cadena de salida además de un símbolo de entrada. Cada compresor denota una función que a cada cadena de entrada le asigna una cadena de salida.

Definición. *Un* compresor de estados finitos *es una tupla* $\langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0 \rangle$ *donde*

- Q *es un conjunto no vacío de estados,*
- $\mathcal{A}_I$ *y* $\mathcal{A}_O$ *son los alfabetos de entrada y salida, respectivamente,*
- $\delta : Q \times \mathcal{A}_I \to Q$ *es la función de transición,*
- $\nu : Q \times \mathcal{A}_I \to {\mathcal{A}_O}^*$ *es la función de salida,*
- $q_0 \in Q$ *es el estado inicial.*

Nos referimos a ellos como FSC *por su nombre en inglés,* finite-state compressors.

Cuando un compresor C lee una cadena se traza un recorrido sobre C del mismo modo que en un autómata finito sin salida [HMU07], acumulando además las cadenas de salida de las transiciones usadas. Extendemos las funciones de transición y de salida para, dado un estado inicial, poder asociar cada cadena finita de símbolos del alfabeto de entrada con el estado final y la cadena producida tras alimentar al compresor con aquella. Se definen $\delta^* : Q \times {\mathcal{A}_I}^* \to Q$ y $\nu^* : Q \times {\mathcal{A}_I}^* \to {\mathcal{A}_O}^*$ como

$$\begin{aligned} \delta^*(q, \lambda) &= q & \nu^*(q, \lambda) &= \lambda \\ \delta^*(q, wc) &= \delta(\delta^*(q, w), c) & \nu^*(q, wc) &= \nu^*(q, w)\nu(\delta^*(q, w), c). \end{aligned}$$

Con frecuencia usamos estas funciones considerando el estado inicial q_0, por lo que definimos la notación más abreviada $\delta^*(w) = \delta^*(q_0, w)$ y $\nu^*(w) = \nu^*(q_0, w)$. La función de compresión denotada por un FSC C es entonces $C(w) = \nu^*(w)$.

No todas las funciones de compresión definibles con FSC son igual de interesantes. Por ejemplo, consideremos un compresor E, esquematizado en la figura 1.1, que a todas las secuencias de entrada, binarias, les asigna la cadena más breve, la vacía.

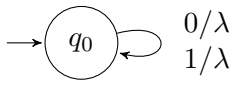

Fig. 1.1: Esquema del FSC E

Aunque la compresión es máxima, no es posible la descompresión, es decir, recuperar w a partir de $E(w) = \lambda$, para w arbitraria. La información contenida en $E(w)$ no es suficiente para esto. Restringiremos nuestro estudio a compresores que no presentan este problema.

Definición. *Decimos que FSC* C no tiene pérdida de información *(o que* C *es un* ILFSC*, por* information-lossless FSC*) si la función* $w \mapsto \langle \delta^*(w), C(w) \rangle$ *es inyectiva.*

Los ILFSC permiten recuperar la cadena de entrada a partir de la de salida y el estado final luego de la lectura.

Para cada compresor hay cadenas que efectivamente resultan compresibles y otras que no lo son. Para un compresor C con alfabetos de entrada y salida con igual cardinalidad, decimos que una cadena w es compresible por C si $|C(w)| < |w|$. En general, para alfabetos arbitrarios es necesario considerar un factor por el cambio de alfabeto. Dado un FSC C cualquiera, una cadena w se dice compresible por C si $|C(w)| < |w| \log_{|\mathcal{A}_O|} |\mathcal{A}_I|$.

Por ejemplo, consideremos el ILFSC F (figura 1.2), extraído de [Sch94], que tiene como alfabetos de entrada y de salida a $\{0,1\}$. La secuencia 111 no es compresible porque $F(111) = 111111$, pero sí lo es 0000, con $F(0000) = 0$. El ILFSC G (figura 1.3), por otro lado, tiene como alfabetos de entrada y de salida a $\{0,1,2,3\}$ y $\{0,1\}$, respectivamente. Aunque el primer dígito leído no genera ninguna salida, es posible recuperarlo conociendo el estado final tras la lectura. Sobre el resto de la cadena de entrada se efectúa un cambio de alfabeto. Si bien $|G(w)| > |w|$ para w de longitud mayor que 1, el análisis de compresibilidad contempla que el alfabeto de salida cuenta con la mitad de los símbolos que el alfabeto de entrada. Toda w es compresible por G debido a que $2(|w| - 1) = |G(w)| < |w| \log_2 4 = 2|w|$.

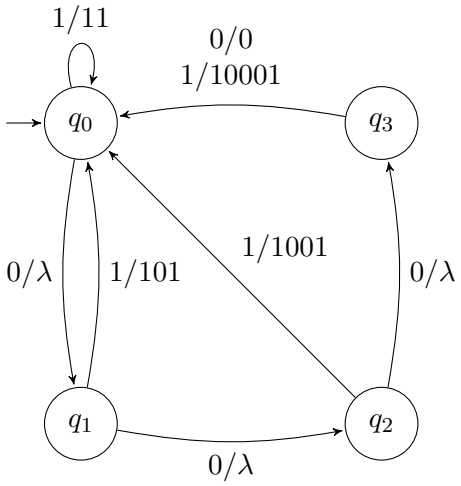

Fig. 1.2: Esquema del ILFSC F

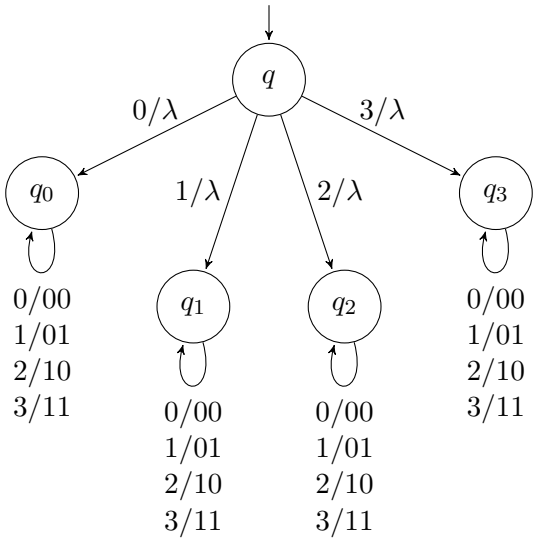

Fig. 1.3: Esquema del ILFSC G

Extendemos la noción de compresibilidad a secuencias infinitas y definimos para ellas

una medida de la compresibilidad alcanzable con esta familia de autómatas, generalizando las definiciones de [DLLM04] a alfabetos arbitrarios.

Definición. *Si C es un FSC con alfabetos de entrada y salida $\mathcal{A}_I$ y $\mathcal{A}_O$, respectivamente, y $\alpha \in {\mathcal{A}_I}^\omega$, decimos que la* tasa de compresión *de C sobre α es*

$$\rho_C(\alpha) = \liminf_{n\to\infty} \frac{|C(\alpha[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|}.$$

La tasa de compresión con estados finitos *de $\alpha \in {\mathcal{A}_I}^\omega$ es*

$$\rho_{FS}(\alpha) = \inf\{\rho_C(\alpha) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\}.$$

Una secuencia infinita es compresible por C si y sólo si la tasa de compresión de C sobre ella es estrictamente menor que 1. En el marco de este trabajo decimos además que una secuencia infinita es compresible si su tasa de compresión con estados finitos es estrictamente menor que 1, e incompresible si no.

Notemos que la tasa de compresión con estados finitos de cualquier secuencia infinita α es menor o igual que 1, porque un FSC C con el alfabeto de entrada adecuado que implementa la identidad como función de codificación, es decir, $C(w) = w$ para toda cadena w, alcanza una tasa de compresión igual a 1 sobre α. Esto quiere decir que las secuencias incompresibles son aquellas cuya tasa de compresión con estados finitos es exactamente 1.

Es fácil ver que $\rho_F(111\ldots) = 2$, $\rho_F(000\ldots) = 1/4$ y $\rho_G(\alpha) = 1$ para toda secuencia $\alpha \in \{0,1,2,3\}^\omega$.

1.3. Selección

Un selector de estados finitos puede ser visto como un compresor de estados finitos muy sencillo. Si tras la lectura de un prefijo de la cadena de entrada se llega a un estado selector, el próximo símbolo se agregará a la cadena de salida. De lo contrario, la cadena de salida no será afectada al atravesar la próxima transición.

Definición. *Un* selector de estados finitos *es una tupla $S = \langle Q, \mathcal{A}, \delta, Q_S, q_0\rangle$ donde*

- *Q es un conjunto no vacío de estados,*
- *$\mathcal{A}$ es el alfabeto de entrada,*
- *$\delta : Q \times \mathcal{A} \to Q$ es la función de transición,*
- *$Q_S \subset Q$ es el conjunto de estados selectores,*
- *$q_0 \in Q$ es el estado inicial,*
- *no existen en S ciclos sin estados selectores.*

La función de transición puede ser extendida a $\delta^* : Q \times \mathcal{A}^* \to Q$ como se hizo con la de los compresores de estados finitos. Definimos la función de selección $\sigma_S : Q \times \mathcal{A}^* \to \mathcal{A}^*$ producida por el selector de estados finitos S como

$$\sigma_S(q, \lambda) = \lambda$$
$$\sigma_S(q, wc) = \begin{cases} \sigma_S(q,w)c & \text{si } \delta^*(q,w) \in Q_S \\ \sigma_S(q,w) & \text{si no} \end{cases}$$

Para la función de selección que comienza desde el estado inicial de S usamos la notación $S(w) = \sigma_S(q_0, w)$.

1.4. Codificación aritmética

Muchas técnicas de codificación de datos sin pérdida de información, entre las que se encuentran los compresores de estados finitos, se basan en la asignación de un código a cada símbolo de la fuente, de modo que para construir el mensaje codificado simplemente se reemplaza cada símbolo del mensaje original por su respectivo código. El miembro más renombrado de esa familia de codificaciones es la de Huffman [Huf52], que produce códigos de longitud óptima dentro de lo que permite un esquema de traducción de símbolo por símbolo.

Si se quiere construir un código binario para una fuente tal que las probabilidades de los símbolos son potencias negativas de dos, la longitud del código asignado por Huffman a cada símbolo será igual a la cantidad de información en bits de dicho símbolo. Como el teorema de la codificación de Shannon [Sha48] indica que la longitud media del código de un símbolo no puede ser menor que la entropía de la fuente, el código obtenido con Huffman es inmejorable. En otros casos, sin embargo, Huffman resulta ineficiente, con un exceso de hasta un bit en el código de cada símbolo en comparación con la cantidad de información representada. Por ejemplo, un símbolo con una probabilidad cercana a 1 transmite una cantidad de información casi nula, pero su código tendrá al menos un bit.

Es posible codificar un mensaje sin pérdida de información con aún menos caracteres que Huffman si consideramos estrategias que no se limitan a la codificación símbolo por símbolo. Ese es el caso de la codificación aritmética [WNC87, Sai04, Mac03], que garantiza códigos de longitud óptima con respecto al conjunto de todas las codificaciones unívocamente decodificables posibles. En esta técnica la compresión consiste en asignarle al mensaje original un intervalo de números reales usando un modelo probabilístico de la fuente. El mensaje codificado se obtiene eligiendo un número real de aquel intervalo y representándolo en algún sistema de numeración.

Para poder implementar un codificador aritmético se necesita estimar, para todo símbolo c, la probabilidad $P(x_j = c|x_1x_2 \dots x_{j-1})$ de que el próximo carácter emitido por la fuente x_j sea c sabiendo que los últimos vistos fueron $x_1, x_2, \dots, x_{j-1}$. Dado un modelo de esas características y fijando una enumeración $c_1, c_2, \dots, c_n$ de los símbolos de la fuente, definimos las probabilidades condicionales acumuladas

$$Q(c_i|x_1 \dots x_{j-1}) = \sum_{k=1}^{i-1} P(x_j = c_k|x_1 \dots x_{j-1}),$$

$$R(c_i|x_1 \dots x_{j-1}) = \sum_{k=1}^{i} P(x_j = c_k|x_1 \dots x_{j-1}).$$

Este método, a diferencia de lo que ocurre con el de Huffman, permite la implementación de una codificación adaptativa, es decir, una que varíe según los símbolos que ya fueron leídos.

El tamaño del intervalo asignado a un mensaje cualquiera w es igual a la probabilidad de w según el modelo probabilístico de la fuente. Además, los intervalos asociados a mensajes de igual longitud son mutuamente disjuntos. A grandes rasgos, el proceso de codificación comienza con el intervalo $[0, 1)$ y tras procesar cada símbolo del mensaje se

toma un intervalo cada vez más pequeño, contenido en el anterior, de manera tal que el tamaño del nuevo subintervalo es proporcional a la probabilidad del símbolo leído.

Con estos elementos ya podemos dar el algoritmo de codificación aritmética (algoritmo 1.1) para el cómputo del intervalo $[a, b)$, notado $\Phi(x_1 x_2 \dots x_n)$, para el mensaje $x_1 x_2 \dots x_n$.

Algoritmo 1.1 Algoritmo de codificación aritmética

$a \leftarrow 0{,}0$
$b \leftarrow 1{,}0$
$s \leftarrow b - a$
for all $i \in \{1 \dots n\}$ **do**
  $b \leftarrow a + s\ R(x_i | x_1 \dots x_{i-1})$
  $a \leftarrow a + s\ Q(x_i | x_1 \dots x_{i-1})$
  $s \leftarrow b - a$
end for

Es posible dar una formulación recursiva cuya equivalencia con la anterior puede comprobarse fácilmente por inducción. Para todos $w \in \mathcal{A}^*$ y $c \in \mathcal{A}$

$$\Phi(\lambda) = [0, 1)$$
$$\Phi(wc) = [\text{mín}(\Phi(w)) + |\Phi(w)|\ Q(c|w), \text{mín}(\Phi(w)) + |\Phi(w)|\ R(c|w)).$$

Estudiemos el funcionamiento del algoritmo cuando se quiere codificar el mensaje *eai*. Las probabilidades relevantes para el proceso se exhiben en la tabla 1.1 y se considera la enumeración de los símbolos correspondiente al orden alfabético.

Símbolo x	$P(x\|\lambda)$	$P(x\|e)$	$P(x\|ea)$
a	0,2	0,2	0,3
e	0,1	0,2	0,2
i	0,3	0,1	0,3
o	0,3	0,4	0,1
u	0,1	0,1	0,1

Tab. 1.1: Modelo probabilístico de la fuente

En la tabla 1.2 se muestran los intervalos que conforman las sucesivas particiones consideradas a lo largo de la evaluación. Se comienza con el intervalo unitario, que se particiona en cinco subintervalos, uno por símbolo. Como el largo del intervalo original es 1, el tamaño de cada subintervalo coincide con la probabilidad de su símbolo asociado. Se selecciona el intervalo $[0,2; 0,3)$, correspondiente al primer símbolo del mensaje, e, y se lo particiona en cinco subintervalos. Como $[0,2; 0,3)$ tiene tamaño $0,1$, el tamaño del subintervalo asociado a x es un décimo de la probabilidad de ocurrencia de x después de haber leído e, $P(x|e)$. El intervalo asociado a ea, entonces, tiene tamaño 0,02 y está al comienzo de $[0,2; 0{,}3)$ según la enumeración de los símbolos. Se trata de $[0,2; 0,22)$. Finalmente consideramos una partición de este intervalo y tomamos el tercer subintervalo, $[0,21; 0,216)$, de longitud igual a $0,02\ P(i|ea) = 0,006$.

Símbolo x	Intervalo de x	Intervalo de ex	Intervalo de eax
a	[0 ; 0,2)	[0,2 ; 0,22)	[0,2 ; 0,206)
e	[0,2; 0,3)	[0,22; 0,24)	[0,206; 0,21)
i	[0,3; 0,6)	[0,24; 0,25)	[0,21 ; 0,216)
o	[0,6; 0,9)	[0,25; 0,29)	[0,216; 0,218)
u	[0,9; 1)	[0,29; 0,3)	[0,218; 0,22)

Tab. 1.2: Intervalos asociados a extensiones en un símbolo de prefijos propios de *eai*

Para codificar *eai* sólo es necesario seleccionar un real en $[0,21; 0,216)$ que tenga una representación breve en el sistema numérico elegido. Por ejemplo, si se quiere usar códigos en base decimal puede elegirse 0,21, que resulta en el código 21. Si la base fuera binaria una posible elección sería $(0,0011011)_2 = (0{,}2109375)_{10}$, que arrojaría el código 0011011.

Para todo $m \in \mathbb{N}$, los intervalos asociados a mensajes de longitud m conforman una partición del intervalo unitario. Por lo tanto, para descomprimir un código sólo hace falta conocer la longitud del mensaje original además del modelo probabilístico usado en la compresión. Este proceso consiste en interpretar el código como un número real y simular el proceso de compresión, particionando intervalos y elegiendo el único subintervalo que contiene el real codificado. Cada elección de un subintervalo determina un símbolo del mensaje original.

A modo de ejemplo, descomprimamos el código 21 según el modelo probabilístico del ejemplo y sabiendo que corresponde a un mensaje de tres símbolos. Dentro de los intervalos que constituyen la partición de $[0, 1)$, el único que contiene al real 0,21 es el correspondiente al símbolo e, por lo que deducimos que el mensaje comienza con ese símbolo. De los intervalos en los que se subdivide el anterior es el primero el que contiene a 0,21, por lo que obtenemos que el segundo símbolo del mensaje es a. Análogamente llegamos a que el tercer símbolo es i y como sabemos que el mensaje tenía tres símbolos termina la descompresión.

Es posible evitar la necesidad de conocer la longitud del mensaje en la descompresión agregando un símbolo de fin de mensaje al modelo probabilístico. Todo mensaje emitido finalizaría con ese símbolo y el criterio de terminación de la descompresión sería encontrar el carácter de fin de mensaje.

La optimalidad de la codificación reside en el hecho de que en cualquiera de aquellos intervalos reales podemos tomar un número cuya representación es suficientemente breve. Si la longitud del intervalo, equivalente a la probabilidad del mensaje asociado M, es p, existe un real contenido en ese rango cuya representación en base b ocupa a lo sumo $\lceil -\log_b p \rceil$ dígitos, lo más cerca posible de la cantidad de información transmitida en M desde la perspectiva de la teoría de la información. Mientras más probable es un mensaje, más grande es su correspondiente intervalo y luego puede acceder a un código más corto.

A pesar de que se comenzó por diferenciar estas dos técnicas, la codificación aritmética es una generalización de la de Huffman. Si en lugar de obtener un código para la totalidad del mensaje se calculara un código aritmético para cada símbolo, se obtendría un código de Huffman.

2. CODIFICADORES ARITMÉTICOS DE ESTADOS FINITOS

En el capítulo anterior exhibimos una familia de compresores producidos por autómatas finitos que codifican un mensaje símbolo por símbolo y además presentamos una técnica de compresión alternativa, la codificación aritmética, que considera la totalidad del mensaje como unidad de codificación. En este capítulo definimos una familia de codificadores producidos por autómatas finitos que implementan la codificación aritmética y que tanto la compresibilidad de estos nuevos codificadores como la de los compresores de estados finitos son invariantes bajo cambios de alfabeto.

2.1. Definición

El rol del autómata en la codificación aritmética de estados finitos es la provisión del modelo probabilístico usado para la asignación de un intervalo a cada mensaje. Esto implica estimar, para todo símbolo c, la probabilidad $P(x_j = c|x_1x_2\ldots x_{j-1})$ de que el próximo símbolo emitido por la fuente x_j sea c sabiendo que los últimos símbolos vistos fueron $x_1, x_2, \ldots, x_{j-1}$. El poder de cómputo limitado de esta construcción no permite almacenar cadenas de símbolos arbitrariariamente largas, provocando que a la hora de formular una distribución de probabilidades para la emisión del próximo símbolo no se cuente con toda la información contenida en el prefijo del mensaje conocido. En efecto, para cada estado se define una distribución de probabilidades para el alfabeto de la fuente y la cadena de símbolos ya emitidos se utiliza para determinar un estado y por lo tanto una función de probabilidad para el próximo símbolo.

Salvo por la presencia de un alfabeto de salida en su definición, esta construcción es similar a una cadena de Markov estacionaria [Chu67] o a una martingala producida por un autómata finito [DLLM04].

En adelante, sea $\mathbb{P} = \mathbb{Q} \cap [0, 1]$.

Definición. *Un* codificador aritmético de estados finitos (FSAC, finite state arithmetic coder) *es una tupla* $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ *donde*

- Q *es un conjunto no vacío de estados,*
- $\mathcal{A}_I$ *y* $\mathcal{A}_O$ *son los alfabetos de entrada y de salida, respectivamente,*
- $\delta : Q \times \mathcal{A}_I \to Q$ *es la función de transición,*
- $p : Q \times \mathcal{A}_I \to \mathbb{P}$ *es una medida de probabilidad positiva para cada estado* q*, es decir,* $\sum_{d\in\mathcal{A}_I} p(q, d) = 1$ *y* $p(q, c) > 0$ *para todos* $q \in Q$ *y* $c \in \mathcal{A}_I$*.*
- $q_0 \in Q$ *es el estado inicial.*

La función de transición puede ser extendida a $\delta^* : Q \times {\mathcal{A}_I}^* \to Q$ como se hizo para las construcciones previas. De forma análoga realizamos la extensión de p a $p^* : Q \times {\mathcal{A}_I}^* \to \mathbb{P}$,

$$\begin{aligned} p^*(q, \lambda) &= 1 \\ p^*(q, wc) &= p^*(q, w)\ p(\delta^*(q, w), c) \end{aligned}$$

o equivalentemente

$$p^*(q,w) = \prod_{i=1}^{|w|} p(\delta^*(q, w[1..i-1]), w[i]). \tag{1}$$

Escribimos $p^*(w)$ y $\delta^*(w)$ para abreviar $p^*(q_0, w)$ y $\delta^*(q_0, w)$, respectivamente.

Cuando usamos un FSAC como modelo probabilístico simplemente tomamos

$$P(x_j = c | x_1, x_2, \ldots, x_{j-1}) = p(\delta^*(x_1, x_2, \ldots, x_{j-1}), c)$$

para c y $x_1, x_2, \ldots, x_{j-1}$ símbolos cualesquiera. Llamamos $\Phi_A(w)$ al intervalo obtenido usando el algoritmo 1.1 con las probabilidades determinadas de esta manera a partir del FSAC A.

Se puede probar fácilmente por inducción que $p^*(q_0, w)$ es la probabilidad del mensaje w según el modelo de la fuente. La función $w \mapsto p^*(q, w)$ es por lo tanto una medida de probabilidad positiva en ${\mathcal{A}_I}^n$ para todo $n \geq 0$ y para todo $q \in Q$. Además, en la codificación aritmética con estados finitos se verifican las propiedades enunciadas en el capítulo anterior, vigentes para modelos probabilísticos de complejidad arbitraria. Una de ellas, que usaremos varias veces en el resto del trabajo, es que para todo codificador aritmético de estados finitos $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0 \rangle$ y para toda cadena $w \in {\mathcal{A}_I}^*$ la longitud del intervalo $\Phi_A(w)$ es igual a $p^*(w)$, la probabilidad de w según A.

2.2. Codificación

Un FSAC $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0 \rangle$ determina una asignación Φ_A de intervalos reales a cada cadena de símbolos del alfabeto de entrada. Como en la codificación aritmética general, el código correspondiente a un mensaje w es la representación en base $|\mathcal{A}_O|$ de un número real contenido en $\Phi_A(w)$. En el caso de que $\mathcal{A}_O$ no sea exactamente $\{0, 1, 2, \ldots, |\mathcal{A}_O| - 1\}$ necesitamos una biyección entre $\mathcal{A}_O$ y ese conjunto de dígitos para poder asociar cada secuencias con un número real. En adelante supondremos que $\mathcal{A}_O = \{0, 1, 2, \ldots, |\mathcal{A}_O| - 1\}$ para simplificar la notación.

Definimos dos funciones de codificación que responden a dos criterios diferentes de selección del número real que se toma como representante. En el primer caso, dentro de los códigos posibles se elige el lexicográficamente menor. En el segundo se elige el código de menor longitud y, en caso de que haya varios, se elige el lexicográficamente menor.

Definición. *Sea $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0 \rangle$ un codificador aritmético de estados finitos. Las funciones de codificación $A : {\mathcal{A}_I}^* \to {\mathcal{A}_O}^*$ y $A^\star : {\mathcal{A}_I}^* \to {\mathcal{A}_O}^*$ están definidas por las ecuaciones*

$$A(w) = \min_{lex} \left\{ v \in {\mathcal{A}_O}^+ : \sum_{i=1}^{|v|} v[i] |\mathcal{A}_O|^{-i} \in \Phi_A(w) \ \wedge \ |v| = \lceil -\log_{|\mathcal{A}_O|} |\Phi_A(w)| \rceil \right\}$$

$$A^\star(w) = \min_{length\text{-}lex} \left\{ v \in {\mathcal{A}_O}^+ : \sum_{i=1}^{|v|} v[i] |\mathcal{A}_O|^{-i} \in \Phi_A(w) \right\}.$$

Con la función de codificación $A^\star$ se genera el código más breve posible dado el intervalo, hasta tal punto que el tamaño del código obtenido puede no guardar relación con la longitud del mensaje original. Para cada $\varepsilon > 0$, existe w tal que $|A^\star(w)|/|w| < \varepsilon$. Si c es el

primer símbolo en la enumeración utilizada para particionar intervalos, basta tomar c^n con n suficientemente grande. El intervalo $\Phi_A(c^n)$ contiene al cero para todo n, de modo que $|A^\star(c^n)|/|c^n| = n^{-1}$. Esta brecha creciente se presenta en toda sucesión de intervalos tal que existe un real con representación finita en base $|\mathcal{A}_O|$ contenido en todos los miembros de la sucesión.

Por el contrario, la longitud de un código obtenido con la función A es predecible. La existencia de un código de tal longitud se sustenta en dos hechos. A partir de una representación de un número real en base b de longitud $n \in \mathbb{N}$ se puede, para $m \in \mathbb{N}$, obtener otra de longitud $n+m$ que denote el mismo real simplemente agregando una cola de m ceros a la derecha de la secuencia original. Como en un intervalo de medida $l > 0$ existe un real r cuya representación en base b tiene tamaño a lo sumo $\lceil -\log_b l \rceil$, es posible extender esa representación de r para que tenga exactamente $\lceil -\log_b l \rceil$ dígitos.

Si $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0 \rangle$ es un FSAC, como $|\Phi_A(w)| = p^*(w)$ para toda $w \in {\mathcal{A}_I}^*$, tenemos que $|A(w)| = \lceil -\log_{|\mathcal{A}_O|} p^*(w) \rceil$ para toda $w \in {\mathcal{A}_I}^*$.

En el resto de este trabajo usamos solamente la primera función de codificación. Se definen para estos compresores las tasas análogas a las vistas para FSC.

Definición. *Si A es un FSAC con alfabetos de entrada y salida $\mathcal{A}_I$ y $\mathcal{A}_O$, respectivamente, y $\alpha \in {\mathcal{A}_I}^\omega$, decimos que la* tasa de compresión *de A sobre α es*

$$\rho_A(\alpha) = \liminf_{n\to\infty} \frac{|A(\alpha[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|}.$$

La tasa de compresión aritmética con estados finitos *de $\alpha \in {\mathcal{A}_I}^\omega$ es*

$$\rho_{FSA}(\alpha) = \text{ínf}\{\rho_A(\alpha) : A \text{ es un FSAC con alfabeto de entrada } \mathcal{A}_I\}.$$

Estas tasas se vinculan con la noción de compresibilidad de la misma manera que las de los FSC. Un FSAC A con alfabetos de entrada y salida $\mathcal{A}_I$ y $\mathcal{A}_O$, respectivamente, comprime una cadena w si y sólo si $|A(w)| < |w| \log_{|\mathcal{A}_O|} |\mathcal{A}_I|$. Una secuencia infinita α es compresible por A si y sólo si la tasa de compresión de A sobre α es menor que 1, y decimos que α es compresible mediante codificadores aritméticos de estados finitos si y sólo si $\rho_{FSA}(\alpha) < 1$.

Observemos que la tasa de compresión aritmética de estados finitos de cualquier secuencia infinita es menor o igual que 1, como su análoga para los FSC, porque el FSAC que describe un modelo probabilístico en el que todos los símbolos son equiprobables en todo contexto implementa como función de codificación la identidad y luego alcanza una tasa de compresión igual a 1 para toda secuencia infita en su dominio. Para esta familia de compresores las secuencias incompresibles son también aquellas cuya tasa de compresión arimética con estados finitos es exactamente 1.

Veamos dos ejemplos de codificadores de estados finitos con esquemas de compresión similares a los de los compresores de estados finitos vistos en el capítulo anterior. En la figura 2.1 presentamos un esquema del codificador aritmético de estados finitos H con alfabeto de salida $\{0, 1\}$, similar al FSC F (figura 1.2). A las cadenas 0000 y 111 les son asignados los intervalos $\Phi_H(0000) \approx [0, 63)$ y $\Phi_H(111) = [124/125, 1)$ y los códigos $H(0000) = 0$ y $H(111) = 1111111$, en este último caso un símbolo más largo que $F(111)$. Por otro lado, en la figura 2.2 está representado el FSAC $I = \langle \{q_0\}, \{0, 1\}, \{0, 1, 2, 3\}, \delta, p, q_0 \rangle$, con $\delta(q_0, c) = q_0$ y $p(q_0, c) = 1/2$ para $c \in \{0, 1\}$. La función de codificación de I computa un cambio de alfabeto como el compresor de estados finitos G (figura 1.3) pero sin descartar el primer carácter. Podemos ver, por ejemplo, que $\Phi_I(1011) = [11/16, 3/4)$ y $I(1011) = 23$.

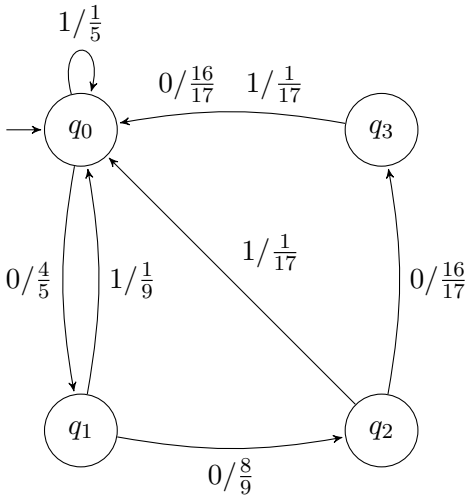

Fig. 2.1: Esquema del FSAC H

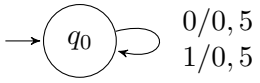

Fig. 2.2: Esquema del FSAC I

2.3. Codificadores aritméticos de estados finitos y compresores de estados finitos

Comenzamos esta sección estudiando una diferencia en el comportamiento de ambos tipos de codificadores con el fin de comprender mejor la codificación aritmética de estados finitos, para luego exponer un resultado que será útil en capítulos posteriores y que constituye un primer vínculo entre los dos conceptos centrales en este trabajo: la compresibilidad, al igual que la normalidad (teorema 1.2), es invariante bajo cambios de alfabeto.

2.3.1. Códigos de prefijos iniciales de una cadena

Llamamos *stream* a una cadena de longitud desconocida pero finita. Si necesitáramos codificar un stream podría ser deseable que el codificador emita símbolos del código final aún cuando todavía no ha recibido la totalidad de la entrada, contando en cada momento con un prefijo del código final lo más largo posible.

En el caso de que la codificación se realizara con un compresor de estados finitos sin pérdida de información C se tiene la certeza de que existe una constante $k_C \in \mathbb{N}$ tal que como mucho se necesitará recibir k_C símbolos del stream, cualquiera sea éste, para aumentar en al menos una unidad la longitud del prefijo conocido del código final. El hecho de que C no tenga pérdida de información asegura la inexistencia de ciclos en los que la función de salida emite λ en todas sus transiciones. Sumando a eso la finitud del conjunto de estados de C se obtiene que la longitud de un camino en el que la función de salida no emite ningún símbolo está acotada por la cantidad de estados de C.

Los codificadores aritméticos de estados finitos no verifican esa propiedad. Dado un FSAC A, no existe una constante k_A tal que la lectura de k_A símbolos de un stream arbitrario permita necesariamente deducir nuevos símbolos del código final. Esto ocurre porque al tomar los códigos correspondientes a dos prefijos de α arbitrarios no necesariamente obtenemos que uno es prefijo del otro, a diferencia de lo que sucede con los FSC.

A pesar de aquello, el carácter finito de A nos permite determinar una cantidad n tal que, si se está procesando una secuencia, tras la lectura de n símbolos más el código del prefijo leído habrá crecido en al menos un carácter. Formalmente, para todo A existe una constante $n \in \mathbb{N}$ para la cual $|A(\alpha[1..m+n])| \geq |A(\alpha[1..m])| + 1$ para α y m cualesquiera. Para probar esto simplemente acotamos inferiormente el aporte de un símbolo a la longitud del código. Si $r = \text{máx}\{p(q,c) : c \in \mathcal{A}_I \ \wedge \ q \in \mathbb{Q}\}$ y tomamos $n = \lceil -2/\log_{|\mathcal{A}_O|}(r) \rceil$, comprobamos que

$$\begin{aligned} |A(\alpha[1..m+n])| &= \lceil -\log_{|\mathcal{A}_O|}(p^*(\alpha[1..m+n])) \rceil \\ &\geq -\log_{|\mathcal{A}_O|}(p^*(\alpha[1..m])) - n \log_{|\mathcal{A}_O|} r \\ &\geq -\log_{|\mathcal{A}_O|}(p^*(\alpha[1..m])) + 2 \\ &\geq \lceil -\log_{|\mathcal{A}_O|}(p^*(\alpha[1..m])) \rceil + 1 \\ &\geq |A(\alpha[1..m])| + 1. \end{aligned}$$

Sean $r \in [0,1)$ con representación finita v_r en base $|\mathcal{A}_O|$ y $w \in {\mathcal{A}_I}^*$. Si $\Phi_A(w)$ está contenido en $[r, r + {|\mathcal{A}_O|}^{-|v_r|})$ es evidente que v_r es un prefijo de $A(w)$. Más aún, para toda $w' \in {\mathcal{A}_I}^*$ se cumple $\Phi_A(ww') \subseteq [r, r + {|\mathcal{A}_O|}^{-|v_r|})$ y entonces v_r es prefijo de $A(ww')$, de modo que si los primeros símbolos de un stream conforman la cadena w podemos asegurar que el código del stream comienza con v_r.

Sea $\alpha \in {\mathcal{A}_I}^\omega$ una secuencia tal que la intersección $\bigcap_{n\in\mathbb{N}} \Phi_A(\alpha[1..n])$ es no vacía, es decir, define un número real r. Si r tiene una representación finita v_r en base $|\mathcal{A}_O|$, puede ocurrir que no exista n tal que $\Phi_A(\alpha[1..n]) \subseteq [r, r + {|\mathcal{A}_O|}^{-|v_r|})$. En ese caso, mientras al procesar un stream los símbolos leídos formen un prefijo de α será imposible predecir una cantidad de caracteres del código asociado al stream completo mayor o igual que $|v_r|$. Veamos un ejemplo de esta situación en la prueba de la siguiente proposición.

Proposición 2.1. *Para cada alfabeto $\mathcal{A}$ existen una secuencia infinita $\alpha \in \mathcal{A}^\omega$ y un FSAC A tales que, para todo n, el prefijo más largo común a los códigos asignados por A a todas las extensiones finitas de $\alpha[1..n]$ es λ.*

Demostración. Sean $\alpha = (0(|\mathcal{A}|-1))^\omega \in \{0,1,\ldots,|\mathcal{A}|-1\}^\omega$ y $A = \langle \{q_0, q_1\}, \mathcal{A}, \mathcal{A}, \delta, p, q_0 \rangle$ un FSAC con

$$\delta(q,c) = q_1$$

$$p(q,c) = \begin{cases} \dfrac{|\mathcal{A}|+1}{|\mathcal{A}|^2} & \text{si } q = q_0 \text{ y } c = 0 \\ \left(1 - \dfrac{|\mathcal{A}|+1}{|\mathcal{A}|^2}\right) \dfrac{1}{|\mathcal{A}|-1} & \text{si } q = q_0 \text{ y } c \neq 0 \\ |\mathcal{A}|^{-1} & \text{si } q = q_1 \end{cases}$$

para $q \in \{q_0, q_1\}$ y $a \in \{0,1,\ldots,|\mathcal{A}|-1\}$. El orden de los elementos de $\mathcal{A}$ que usaremos en la codificación aritmética para asignar intervalos es el usual.

Comprobemos por inducción en $i \in \mathbb{N}_0$ que

$$\Phi_A(\alpha[1..2i+1]) = \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}}, \frac{|\mathcal{A}|^{2i+1}+1}{|\mathcal{A}|^{2i+2}}\right) \quad \text{con medida} \quad |\Phi_A(\alpha[1..2i+1])| = \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+2}}$$

de donde se deduce que $\bigcap_{n\in\mathbb{N}} \Phi_A(\alpha[1..n]) = |\mathcal{A}|^{-1}$. Es claro que $\Phi_A(0) = [0, (|\mathcal{A}|+1)|\mathcal{A}|^{-2})$ y $|\Phi_A(0)| = (|\mathcal{A}|+1)|\mathcal{A}|^{-2}$. Por otro lado, si suponemos que

$$\Phi_A(\alpha[1..2i-1]) = \left[\frac{|\mathcal{A}|^{2i-2}-1}{|\mathcal{A}|^{2i-1}}, \frac{|\mathcal{A}|^{2i-1}+1}{|\mathcal{A}|^{2i}}\right) \quad \text{con medida} \quad |\Phi_A(\alpha[1..2i-1])| = \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i}}$$

obtenemos con el algoritmo de codificación aritmética que

$$\begin{aligned}
\Phi_A(\alpha[1..2i]) =& [\text{mín}(\Phi_A(\alpha[1..2i-1])) + |\Phi_A(\alpha[1..2i-1])|\ Q(|\mathcal{A}|-1 \mid \alpha[1..2i-1]),\\
&\quad \text{mín}(\Phi_A(\alpha[1..2i-1])) + |\Phi_A(\alpha[1..2i-1])|\ R(|\mathcal{A}|-1 \mid \alpha[1..2i-1]))\\
=& \left[\frac{|\mathcal{A}|^{2i-2}-1}{|\mathcal{A}|^{2i-1}} + \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i}}\frac{|\mathcal{A}|-1}{|\mathcal{A}|}, \frac{|\mathcal{A}|^{2i-2}-1}{|\mathcal{A}|^{2i-1}} + \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i}}\right)\\
=& \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}}, \frac{|\mathcal{A}|^{2i-1}+1}{|\mathcal{A}|^{2i}}\right)\\
|\Phi_A(\alpha[1..2i])| =& \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+1}}
\end{aligned}$$

y

$$\begin{aligned}
\Phi_A(\alpha[1..2i+1]) =& [\text{mín}(\Phi_A(\alpha[1..2i])) + |\Phi_A(\alpha[1..2i])|\ Q(0 \mid \alpha[1..2i]),\\
&\quad \text{mín}(\Phi_A(\alpha[1..2i])) + |\Phi_A(\alpha[1..2i])|\ R(0 \mid \alpha[1..2i]))\\
=& \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}}, \frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}} + \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+1}}\frac{1}{|\mathcal{A}|}\right)\\
=& \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}}, \frac{|\mathcal{A}|^{2i+1}+1}{|\mathcal{A}|^{2i+2}}\right)\\
|\Phi_A(\alpha[1..2i+1])| =& \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+2}}.
\end{aligned}$$

De forma análoga puede probarse para $i, m \in \mathbb{N}_0$ que

$$\begin{aligned}
\Phi_A(\alpha[1..2i+1]0^m) &= \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}}, \frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}} + \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+m+2}}\right)\\
|\Phi_A(\alpha[1..2i+1]0^m)| &= \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+m+2}}\\
\Phi_A(\alpha[1..2i+1](|\mathcal{A}|-1)^m) &= \left[\frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}} + \frac{(|\mathcal{A}|^2-1)(|\mathcal{A}|^m-1)}{|\mathcal{A}|^{2i+3}(|\mathcal{A}|^m-|\mathcal{A}|^{m-1})}, \frac{|\mathcal{A}|^{2i+1}+1}{|\mathcal{A}|^{2i+2}}\right)\\
|\Phi_A(\alpha[1..2i+1](|\mathcal{A}|-1)^m)| &= \frac{|\mathcal{A}|^2-1}{|\mathcal{A}|^{2i+3}(|\mathcal{A}|^m-|\mathcal{A}|^{m-1})}.
\end{aligned}$$

Fijando i en un entero no negativo cualquiera, como

$$\lim_{m\to\infty} \frac{|\mathcal{A}|+1}{|\mathcal{A}|^{2i+m+2}} = 0$$

$$\lim_{m\to\infty} \frac{|\mathcal{A}|^{2i}-1}{|\mathcal{A}|^{2i+1}} + \frac{(|\mathcal{A}|^2-1)(|\mathcal{A}|^m-1)}{|\mathcal{A}|^{2i+3}(|\mathcal{A}|^m-|\mathcal{A}|^{m-1})} = |\mathcal{A}|^{-1} + |\mathcal{A}|^{-2i-2}$$

existe m_0 tal que para $m > m_0$ los intervalos $\Phi_A(\alpha[1..2i+1]0^m)$ están contenidos en $[0, |\mathcal{A}|^{-1})$, por un lado, y todo $\alpha[1..2i+1](|\mathcal{A}|-1)^m$ está incluido en $[|\mathcal{A}|^{-1}, 1)$, por otro. Esto indica que $A(\alpha[1..2i+1]0^m)$ comienza con cero y $A(\alpha[1..2i+1](|\mathcal{A}|-1)^m)$, con un símbolo distinto de cero, si $m > m_0$. Dado que $\alpha[1..2i+1]0^m$ y $\alpha[1..2i+1](|\mathcal{A}|-1)^m$ son extensiones de $\alpha[1..n]$ para $n \leq 2i+1$ e i es un entero no negativo arbitrario, concluimos que el prefijo común a los códigos de todas las extensiones de $\alpha[1..n]$ es λ. □

Es imposible predecir el primer símbolo del código de un stream cuyos símbolos corresponden a un prefijo de la secuencia α usada en esta prueba. Afortunadamente, el conjunto de secuencias tales que la intersección $\bigcap_{n\in\mathbb{N}} \Phi(\alpha[1..n])$ es no vacía y determina un real r con representación finita v_r en base $|\mathcal{A}_O|$ tiene medida cero, pues los números racionales son numerables.

2.3.2. Invariancia de la compresibilidad bajo cambios de alfabeto

La siguiente proposición nos será útil para comparar tasas de compresibilidad de secuencias vinculadas por un cambio de alfabeto. Aunque en general el límite de una sucesión no coincide con aquel de una subsucesión, podemos calcular correctamente la tasa de compresión de un FSC o de un FSAC sólo viendo términos a intervalos regulares.

Proposición 2.2. *Si D es un codificador de estados finitos con función de probabilidad p o un compresor de estados finitos con función de salida ν, los alfabetos de entrada y de salida son $\mathcal{A}_I$ y $\mathcal{A}_O$, respectivamente, $\alpha \in {\mathcal{A}_I}^\omega$ y $k \in \mathbb{N}$,*

$$\liminf_{n\to\infty} \frac{|D(\alpha[1..n])|}{n} = \liminf_{n\to\infty} \frac{|D(\alpha[1..nk])|}{nk}$$

Demostración. Sabemos que para todo n

$$\frac{|D(\alpha[1..\lfloor n/k\rfloor k])|}{n} \leq \frac{|D(\alpha[1..n])|}{n} \leq \frac{|D(\alpha[1..(\lfloor n/k\rfloor+1)k])|}{n}.$$

La finitud de los estados de D tiene como consecuencia que el conjunto de códigos asignados a las cadenas de longitud k también sea finito. Si D es un FSC tomemos $m = \max\{|\nu(q,w)| : q \in Q \ \wedge \ w \in ({\mathcal{A}_I}^k)^*\}$ y si es un codificador aritmético de estados finitos, $m = \max\{\lceil -\log_{|\mathcal{A}_O|} p^*(q,w)\rceil : q \in Q \ \wedge \ w \in ({\mathcal{A}_I}^k)^*\}$. En el primer caso es claro que $|D(\alpha[1..(\lfloor n/k\rfloor+1)k])| \leq |D(\alpha[1..\lfloor n/k\rfloor k])|+m$. Por otro lado, si D es un FSAC

$$\begin{aligned}
&|D(\alpha[1..(\lfloor n/k\rfloor+1)k])| \\
&= \lceil -\log_{|\mathcal{A}_O|} p^*(\alpha[1..(\lfloor n/k\rfloor+1)k])\rceil \\
&= \lceil -\log_{|\mathcal{A}_O|}(p^*(\alpha[1..\lfloor n/k\rfloor k])\ p^*(\delta^*(\alpha[1..\lfloor n/k\rfloor k]), \alpha[\lfloor n/k\rfloor k+1..(\lfloor n/k\rfloor+1)k]))\rceil \\
&\leq |D(\alpha[1..\lfloor n/k\rfloor k])| + m.
\end{aligned}$$

En ambos casos, entonces, afirmamos que para todo n

$$\frac{|D(\alpha[1..\lfloor n/k\rfloor k])|}{n} \leq \frac{|D(\alpha[1..n])|}{n} \leq \frac{|D(\alpha[1..\lfloor n/k\rfloor k])|}{n} + \frac{m}{n}$$

y como las expresiones de los extremos tienen igual límite inferior cuando n tiende a infinito, obtenemos que

$$\begin{aligned}
\liminf_{n\to\infty} \frac{|D(\alpha[1..n])|}{n} &= \liminf_{n\to\infty} \frac{|D(\alpha[1..\lfloor n/k\rfloor k])|}{n} = \liminf_{n\to\infty} \frac{|D(\alpha[1..\lfloor n/k\rfloor k])|}{\lfloor n/k\rfloor k} \\
&= \liminf_{n\to\infty} \frac{|D(\alpha[1..nk])|}{nk}.
\end{aligned}$$

□

Demostremos que, al igual que la normalidad, la compresibilidad mediante codificadores aritméticos de estados finitos y compresores de estados finitos son invariantes bajo cambios de alfabeto.

Teorema 2.3. *Sea k un natural cualquiera. Si la secuencia $\alpha \in {\mathcal{A}_I}^\omega$ está vinculada con $\beta \in ({\mathcal{A}_I}^k)^\omega$ a través de un cambio de alfabeto, $\rho_{FS}(\alpha) = \rho_{FS}(\beta)$ y $\rho_{FSA}(\alpha) = \rho_{FSA}(\beta)$.*

La idea que seguimos es, para cada compresor que lee secuencias de ${\mathcal{A}_I}^\omega$, construir otro que procese secuencias de $({\mathcal{A}_I}^k)^\omega$ logrando la misma tasa de compresión y viceversa.

Comencemos definiendo por cada codificador que procese secuencias de símbolos de $\mathcal{A}_I$ uno del mismo tipo pero que lee símbolos de ${\mathcal{A}_I}^k$ con un comportamiento similar al primero y que alcanza sus mismas tasas de compresión. Tanto para los codificadores aritméticos como para los compresores la estrategia consiste en definir al nuevo autómata usando las funciones de transición, salida y probabilidad para secuencias del autómata original.

Definición. *Sea $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ un FSC. Definimos $C^k = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta', \nu', q_0\rangle$ al FSC tal que*

$$\delta'(q, c) = \delta^*(q, c)$$
$$\nu'(q, c) = \nu^*(q, c)$$

Definición. *Si $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ es un FSAC, $A^k = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta', p', q_0\rangle$ es el FSAC tal que*

$$\delta'(q, c) = \delta^*(q, c)$$
$$p'(q, c) = p^*(q, c)$$

Como en estas construcciones $\delta'^* = \delta^*$, $\nu'^* = \nu^*$ y $p'^* = p^*$, aseguramos que para todo $w \in ({\mathcal{A}_I}^k)^*$ se cumple $|C^k(w)| = |C(w)|$ y $|A^k(w)| = |A(w)|$, además de que si C no tiene pérdida de información entonces C^k tampoco.

Lema 2.4. *Sean $k \in \mathbb{N}$ y D un FSC o FSAC con alfabeto de entrada $\mathcal{A}_I$. Si $\alpha \in {\mathcal{A}_I}^\omega$ está vinculada con $\beta \in ({\mathcal{A}_I}^k)^\omega$ a través de un cambio de alfabeto, $\rho_D(\alpha) = \rho_{D^k}(\beta)$.*

Demostración. Sea D un FSC o un FSAC con alfabeto de entrada $\mathcal{A}_I$. En virtud de la proposición 2.2

$$\rho_D(\alpha) = \liminf_{n\to\infty} \frac{|D(\alpha[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} = \liminf_{n\to\infty} \frac{|D(\alpha[1..nk])|}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} = \liminf_{n\to\infty} \frac{|D^k(\beta[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|^k} = \rho_{D^k}(\beta).$$

□

Definamos ahora un autómata que lea símbolos de $\mathcal{A}_I$ a partir de cada FSC o FSAC con alfabeto de entrada ${\mathcal{A}_I}^k$, cuidando que el comportamiento del primero coincida con el del segundo cuando se procese una cantidad de símbolos múltiplo de k. Si eso ocurre, la proposición 2.2 garantiza la igualdad de sus tasas de compresión.

En ambos casos, para poder manejar cadenas con longitudes que no son múltiplos de k, las construcciones definidas abajo se obtienen reemplazando el conjunto de aristas salientes de un estado del autómata original por un árbol $|\mathcal{A}_I|$-ario completo de altura $k-1$, agregando los nodos internos del árbol como estados nuevos. Esta estructura permite almacenar hasta $k-1$ símbolos leídos de la cadena de entrada.

Si el autómata original es un FSC, como sólo especifica para secuencias en $({\mathcal{A}_I}^k)^*$ qué cadenas de salida debe asociarles el nuevo autómata, el nuevo FSC generará output cada k símbolos leídos solamente.

En el caso de los FSAC el autómata original también especifica únicamente para secuencias en $({\mathcal{A}_I}^k)^*$ las probabilidades que debe asociarles el nuevo autómata. Para asignarle una probabilidad a una cadena w con $|w|$ no divisible por k, sumamos las probabilidades asignadas por el autómata original a todas las extensiones a derecha mínimas de w que convierten su longitud en múltiplo de k.

Definición. *Sean $k \in \mathbb{N}$, $C = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta, \nu, q_0\rangle$ un compresor de estados finitos. Definimos el FSC $C^{-k} = \langle Q \times {\mathcal{A}_I}^{<k}, \mathcal{A}_I, \mathcal{A}_O, \delta', \nu', \langle q_0, \lambda\rangle\rangle$ como aquel en el que*

$$\delta'(\langle q, w\rangle, c) = \begin{cases} \langle q, wc\rangle & si\ |w| < k-1 \\ \langle \delta(q, wc), \lambda\rangle & si\ |w| = k-1 \end{cases}$$

$$\nu'(\langle q, w\rangle, c) = \begin{cases} \lambda & si\ |w| < k-1 \\ \nu(q, wc) & si\ |w| = k-1 \end{cases}$$

Definición. *Si $k \in \mathbb{N}$ y $A = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta, p, q_0\rangle$ es un codificador aritmético de estados finitos, llamamos $A^{-k} = \langle Q \times {\mathcal{A}_I}^{<k}, \mathcal{A}_I, \mathcal{A}_O, \delta', p', \langle q_0, \lambda\rangle\rangle$ al FSAC tal que*

$$\delta'(\langle q, w\rangle, c) = \begin{cases} \langle q, wc\rangle & si\ |w| < k-1 \\ \langle \delta(q, wc), \lambda\rangle & si\ |w| = k-1 \end{cases}$$

$$p'(\langle q, w\rangle, c) = \frac{\sum_{v \in {\mathcal{A}_I}^{k-|w|-1}} p(q, wcv)}{\sum_{v \in {\mathcal{A}_I}^{k-|w|}} p(q, wv)}$$

Se puede probar fácilmente por inducción que si $v \in ({\mathcal{A}_I}^k)^*$ y $w \in {\mathcal{A}_I}^{<k}$ entonces

$$\delta'^*(\langle q, \lambda\rangle, vw) = \langle \delta^*(q, v), w\rangle \tag{II}$$

y $\nu'^*(\langle q, \lambda\rangle, vw) = \nu^*(q, v)$. En consecuencia, $|C(w)| = |C^{-k}(w)|$ para toda $w \in ({\mathcal{A}_I}^k)^*$. Además C^{-k} es un ILFSC si C no tiene pérdida de información.

La siguiente proposición nos permite afirmar que $|A(w)| = |A^{-k}(w)|$ para toda cadena $w \in ({\mathcal{A}_I}^k)^*$.

Proposición 2.5. *Para todo $w \in {\mathcal{A}_I}^{\leq k}$, $p'^*(\langle q, \lambda\rangle, w) = \sum_{v \in {\mathcal{A}_I}^{k-|w|}} p(q, wv)$. Por lo tanto, para todo $w \in ({\mathcal{A}_I}^k)^*$, $p'^*(\langle q, \lambda\rangle, w) = p^*(q, w)$.*

Demostración. Probemos que para $n \leq k$ y $w \in {\mathcal{A}_I}^n$, $p'^*(\langle q, \lambda\rangle, w) = \sum_{v \in {\mathcal{A}_I}^{k-|w|}} p(q, wv)$ por inducción en n.

Si $n = 0$,

$$p'^*(\langle q, \lambda\rangle, \lambda) = 1 = \sum_{v \in {\mathcal{A}_I}^k} p(q, v)$$

por ser p una medida de probabilidad positiva en ${\mathcal{A}_I}^k$ para el estado q.

Supongamos ahora que la proposición vale para $n < k$. Si $wd \in {\mathcal{A}_I}^{n+1}$, por (II), hipótesis inductiva y definición de p'

$$\begin{aligned} p'^*(\langle q,\lambda\rangle, wd) &= p'^*(\langle q,\lambda\rangle, w)\ p'(\delta'^*(\langle q,\lambda\rangle, w), d) = p'^*(\langle q,\lambda\rangle, w)\ p'(\langle \delta^*(q,\lambda), w\rangle, d) \\ &= p'^*(\langle q,\lambda\rangle, w)\ p'(\langle q, w\rangle, d) \\ &= \left(\sum_{v\in {\mathcal{A}_I}^{k-|w|}} p(q, wv)\right) \frac{\sum_{v\in {\mathcal{A}_I}^{k-|w|-1}} p(q, wdv)}{\sum_{v\in {\mathcal{A}_I}^{k-|w|}} p(q, wv)} \\ &= \sum_{v\in {\mathcal{A}_I}^{k-|wd|}} p(q, wdv). \end{aligned}$$

Tenemos entonces que si $v \in {\mathcal{A}_I}^k$,

$$p'^*(\langle q,\lambda\rangle, v) = p(q, v). \tag{III}$$

Luego, si tomamos $w_1 w_2 w_3 \dots w_n = w \in ({\mathcal{A}_I}^k)^*$, con $w_i \in {\mathcal{A}_I}^k$ para todo i,

$$\begin{aligned} p'^*(\langle q,\lambda\rangle, w) &= \prod_{i=1}^{n} p'^*(\delta'^*(\langle q,\lambda\rangle, w_1 \dots w_{i-1}), w_i) && \text{por (I)} \\ &= \prod_{i=1}^{n} p'^*(\langle \delta^*(q, w_1 \dots w_{i-1}),\lambda\rangle, w_i) && \text{por (II)} \\ &= \prod_{i=1}^{n} p(\delta^*(q, w_1 \dots w_{i-1}), w_i) && \text{por (III)} \\ &= p^*(q, w) && \text{por (I)}. \end{aligned}$$

□

Lema 2.6. *Sean $k \in \mathbb{N}$ y D un FSC o FSC con alfabeto de entrada ${\mathcal{A}_I}^k$. Si $\alpha \in {\mathcal{A}_I}^\omega$ está vinculada con $\beta \in ({\mathcal{A}_I}^k)^\omega$ a través de un cambio de alfabeto, $\rho_D(\beta) = \rho_{D^{-k}}(\alpha)$.*

Demostración. Sea D un FSC o FSAC con alfabeto de entrada ${\mathcal{A}_I}^k$. Sabemos que para todo $w \in ({\mathcal{A}_I}^k)^*$ $|D(w)| = |D^{-k}(w)|$. Esta propiedad y la proposición 2.2 nos permiten deducir que

$$\begin{aligned} \rho_D(\beta) &= \liminf_{n\to\infty} \frac{|D(\beta[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|^k} = \liminf_{n\to\infty} \frac{|D^{-k}(\beta[1..n])|}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} \\ &= \liminf_{n\to\infty} \frac{|D^{-k}(\alpha[1..nk])|}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} = \liminf_{n\to\infty} \frac{|D^{-k}(\alpha[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} = \rho_{D^{-k}}(\alpha). \end{aligned}$$

□

Con estos dos lemas podemos probar la invariancia de la compresibilidad bajo cambios de alfabeto.

Demostración del Teorema 2.3. Sean $\alpha \in {\mathcal{A}_I}^\omega$ y $\beta \in ({\mathcal{A}_I}^k)^\omega$. Por el lema 2.4

$$\begin{aligned} \rho_{FS}(\beta) &= \text{ínf}\{\rho_C(\beta) : C \text{ es un ILFSC con alfabeto de entrada } {\mathcal{A}_I}^k\} \\ &\le \text{ínf}\{\rho_{C^k}(\beta) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\} \\ &\le \text{ínf}\{\rho_C(\alpha) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\} \\ &\le \rho_{FS}(\alpha). \end{aligned}$$

Por otro lado, según el lema 2.6

$$\begin{aligned}\rho_{FS}(\alpha) &= \text{ínf}\{\rho_C(\alpha) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\}\\ &\leq \text{ínf}\{\rho_{C^{-k}}(\alpha) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\}\\ &\leq \text{ínf}\{\rho_C(\beta) : C \text{ es un ILFSC con alfabeto de entrada } {\mathcal{A}_I}^k\}\\ &\leq \rho_{FS}(\beta).\end{aligned}$$

En consecuencia, $\rho_{FS}(\alpha) = \rho_{FS}(\beta)$. Análogamente, $\rho_{FSA}(\alpha) = \rho_{FSA}(\beta)$. □

Aunque los FSAC o FSC D, $(D^k)^{-k}$ y $(D^{-k})^k$ tienen el mismo alfabeto de entrada e iguales tasas de compresión sobre toda secuencia, no son necesariamente el mismo autómata. Mientras que los conjuntos de estados de D y D^k son iguales, la transformación de D a D^{-k} agrega nuevos estados.

3. RESULTADOS PRINCIPALES

En este capítulo probamos que las secuencias normales son exactamente las incompresibles por autómatas finitos. Para esto primero vinculamos las tasas de compresión de ILFSC y FSAC y luego comprobamos que una secuencia es compresible mediante FSAC si y sólo si es normal. Usando este último resultado, concluimos el capítulo con una prueba del teorema Agafonov.

3.1. Igualdad de tasas de compresión de compresores de estados finitos sin pérdida de información y codificadores aritméticos de estados finitos

Probamos la igualdad de las tasas de compresión de los compresores de estados finitos sin pérdida de información y los codificadores aritméticos de estados finitos.

Teorema 3.1. *$\rho_{FS}(\alpha) = \rho_{FSA}(\alpha)$ para toda secuencia α.*

Factorizamos la demostración en dos partes de la manera usual.

Lema 3.2. *$\rho_{FS}(\alpha) \leq \rho_{FSA}(\alpha)$ para toda secuencia α.*

Lema 3.3. *$\rho_{FS}(\alpha) \geq \rho_{FSA}(\alpha)$ para toda secuencia α.*

Para probar estos resultados veamos que para todo FSAC se puede construir un ILFSC cuya tasa de compresión sobre cualquier secuencia es arbitrariamente cercana a la del primero y viceversa.

El conjunto de códigos asignados por un FSAC a cada símbolo a partir de un estado cualquiera puede no ser libre de prefijos, pero podría reemplazarse por otro con esa propiedad respetando las longitudes originales.

Proposición 3.4. *Si $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ es un FSAC, existe $f : Q \times \mathcal{A}_I \to {\mathcal{A}_O}^+$ tal que $\{f(q,c) : c \in \mathcal{A}_I\}$ es libre de prefijos y $|f(q,c)| = |A(q,c)|$ para todo $c \in \mathcal{A}_I$, $q \in Q$.*

Demostración. Si las longitudes $l_1, \ldots, l_n$ satisfacen la desigualdad de Kraft,

$$\sum_{1 \leq i \leq n} |\mathcal{A}_O|^{-l_i} \leq 1,$$

existen $w_1, \ldots, w_n \in {\mathcal{A}_O}^*$ tales que $|w_i| = l_i$ para todo i y $\{w_1, \ldots, w_n\}$ es libre de prefijos (Teorema 1.11.1, [LV08], página 74).

Luego basta ver que para todo estado q las longitudes $\{|A(q,c)| : c \in \mathcal{A}_I\}$ satisfacen la desigualdad de Kraft, es decir,

$$\sum_{c \in \mathcal{A}_I} |\mathcal{A}_O|^{-|A(q,c)|} \leq 1.$$

Tomemos, entonces, $q \in Q$ arbitrario. Como los intervalos asociados a los símbolos constituyen una partición de $[0, 1)$,

$$\begin{aligned}\sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{-|A(q,c)|} &= \sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{-\lceil -\log_{|\mathcal{A}_O|} |\Phi_A(q,c)|\rceil} \\ &\leq \sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{\log_{|\mathcal{A}_O|} |\Phi_A(q,c)|} \\ &\leq \sum_{c\in\mathcal{A}_I} |\Phi_A(q,c)| = 1. \end{aligned}$$ □

Usando esta proposición podemos definir un ILFSC a partir de un FSAC de modo que generen códigos de la misma longitud para símbolos individuales.

Definición. *Sea $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ un FSAC, y $f : Q \times \mathcal{A}_I \to {\mathcal{A}_O}^+$ como en la proposición 3.4. Definimos a $C_A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ como el compresor de estados finitos tal que $\nu(q, c) = f(q, c)$.*

Para todo FSAC A con alfabeto de entrada $\mathcal{A}_I$ y $v, w \in {\mathcal{A}_I}^*$ tales que $v \neq w$ se verifica $C_A(v) \neq C_A(w)$ y por lo tanto C_A es un ILFSC. Esto se puede comprobar tomando el mínimo $i \in \mathbb{N}$ tal que $v[i] \neq w[i]$ y viendo que como $\{\nu(\delta^*(q_0, v[1..i-1]), c) : c \in \mathcal{A}_I\}$ es libre de prefijos existe j tal que $\nu(\delta^*(q_0, v[1..i-1]), v[i])[j] \neq \nu(\delta^*(q_0, v[1..i-1]), w[i])[j]$, lo que a su vez implica $C_A(v)[|C_A(v[1..i-1])| + j] \neq C_A(w)[|C_A(v[1..i-1])| + j]$.

Los códigos producidos por C_A son más largos que los de A, pero la diferencia está acotada. Si $w \in {\mathcal{A}_I}^n$,

$$\begin{aligned}|C_A(w)| &= \sum_{i=1}^{n} \lceil -\log_{|\mathcal{A}_O|} p(\delta^*(q, w[1..i-1]), w[i])\rceil \\ &\leq n + \sum_{i=1}^{n} -\log_{|\mathcal{A}_O|} p(\delta^*(q, w[1..i-1]), w[i]) \\ &\leq n - \log_{|\mathcal{A}_O|} \prod_{i=1}^{n} p(\delta^*(q, w[1..i-1]), w[i]) \\ &\leq n - \log_{|\mathcal{A}_O|} p^*(q, w) \leq n + \lceil -\log_{|\mathcal{A}_O|} p^*(q, w)\rceil = n + |A(w)|.\end{aligned}$$

Si el exceso en la codificación de una secuencia w está acotado por la cantidad de símbolos en w, podemos reducirlo recurriendo a un cambio de alfabeto en A antes de construir el FSC. Por lo tanto, si $k \in \mathbb{N}$, $\alpha \in {\mathcal{A}_I}^\omega$ y $\beta \in ({\mathcal{A}_I}^k)^\omega$, tales que α y β están vinculadas por un cambio de alfabeto,

$$\begin{aligned}\rho_{C_{A^k}}(\beta) &= \liminf_{n\to\infty} \frac{|C_{A^k}(\beta[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|^k} \\ &\leq \liminf_{n\to\infty} \frac{|A^k(\beta[1..n])| + n}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} \\ &\leq \liminf_{n\to\infty} \frac{|A(\alpha[1..nk])| + n}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} \\ &\leq \rho_A(\alpha) + (k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|)^{-1}.\end{aligned}$$

Esta desigualdad, que usaremos para probar el lema 3.2, nos dice que si queremos un FSC que alcance sobre α una tasa de compresión que no supere a la de A en ε, podemos tomar $(C_{A^k})^{-k}$ con k tal que $(k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|)^{-1} < \varepsilon$.

Demostración del lema 3.2. Si $\alpha \in {\mathcal{A}_I}^\omega$, de acuerdo con la invariancia de la compresibilidad por cambios de alfabeto (teorema 2.3),

$$\begin{aligned}
\rho_{FS}(\alpha) &= \text{ínf}\{\rho_C(\alpha) : C \text{ es un ILFSC con alfabeto de entrada } \mathcal{A}_I\} \\
&= \text{ínf}\{\rho_C(\beta) : k \in \mathbb{N},\ C \text{ es un ILFSC con alfabeto de entrada } {\mathcal{A}_I}^k \text{ y} \\
&\qquad \beta \in ({\mathcal{A}_I}^k)^\omega \text{ obtenida tras un cambio de alfabeto de } \alpha\} \\
&\leq \text{ínf}\{\rho_{C_{A^k}}(\beta) : A \text{ es un FSAC con alfabeto de entrada } \mathcal{A}_I,\ k \in \mathbb{N} \text{ y} \\
&\qquad \beta \in ({\mathcal{A}_I}^k)^\omega \text{ obtenida tras un cambio de alfabeto de } \alpha\} \\
&\leq \text{ínf}\{\rho_A(\alpha) + (k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|)^{-1} : A \text{ es un FSAC con alfabeto de entrada } \mathcal{A}_I \text{ y} \\
&\qquad k \in \mathbb{N}\} \\
&\leq \text{ínf}\{\rho_A(\alpha) : A \text{ es un FSAC con alfabeto de entrada } \mathcal{A}_I\} \\
&\leq \rho_{FSA}(\alpha).
\end{aligned}$$

□

Para definir un FSAC que produzca códigos de longitudes similares a las de un ILFSC, la probabilidad asignada por el primero a un símbolo debe ser menor mientras más largo es el código que le asigna el segundo autómata a ese símbolo. Una forma de hacer esto es la siguiente.

Definición. *Sea $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ un ILFSC. Llamamos $A_C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ al FSAC tal que*

$$p(q,c) = \frac{|\mathcal{A}_O|^{-|\nu(q,c)|}}{\sum_{d \in \mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,d)|}}.$$

Es evidente que $c \mapsto p(q,c)$ es una medida de probabilidad positiva en $\mathcal{A}_I$ para todo estado q. Si $\sum_{d \in \mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,d)|} > 1$, la longitud del código determinado por A_C para un símbolo es mayor que la del que le asigna C. Si $\sum_{d \in \mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,d)|} < 1$ ocurre lo contrario, y los códigos elegidos por A_C para un símbolo son más cortos que los que le corresponden por C.

Para precisar el exceso en la longitud de la codificación mediante A_C en comparación con el original de C podemos acotar el factor de normalización de las probabilidades, que es exactamente la suma definida en la desigualdad de Kraft. Para esto nos restringimos sin pérdida de generalidad a analizar ILFSC con todos sus estados alcanzables desde el inicial, pues el autómata que se obtiene eliminando los estados no alcanzables tiene igual comportamiento que el original en lo que respecta a codificación. En un ILFSC de esas características no puede haber dos transiciones de un estado a otro que produzcan el mismo output.

Proposición 3.5. *Si $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ es un ILFSC con todos sus estados alcanzables desde q_0,*

$$\sum_{c \in \mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,c)|} \leq |Q| + |Q| \left\lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \right\rceil$$

para todo estado q.

Demostración. Fijemos q en un estado arbitrario. Si $S_{q,r} = \{c \in \mathcal{A}_I : \delta(q,c) = r\}$, podemos reescribir

$$\sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,c)|} = \sum_{r\in Q}\sum_{c\in S_{q,r}} |\mathcal{A}_O|^{-|\nu(q,c)|}.$$

Sabemos que, para todo estado r, $\nu(q,c) \neq \nu(q,c')$ si $c, c' \in S_{q,r}$ y $c \neq c'$ porque C es un ILFSC. Para cada r, sea l_r un número lo suficientemente grande como para que haya al menos tantos códigos distintos de longitud menor o igual que l_r como símbolos en $S_{q,r}$. Como la suma que deseamos acotar crece si se cambia un código asignado por ν por otro más corto y además cada sumando es positivo, afirmamos que

$$\sum_{c\in S_{q,r}} |\mathcal{A}_O|^{-|\nu(q,c)|} \leq \sum_{w\in\mathcal{A}_O{}^{\leq l_r}} |\mathcal{A}_O|^{-|w|} = \sum_{i=0}^{l_r}\sum_{w\in\mathcal{A}_O{}^i} |\mathcal{A}_O|^{-i} = 1 + l_r.$$

Comprobemos que para r tal que $|S_{q,r}| > 0$ es posible elegir $l_r = \lceil \log_{|\mathcal{A}_O|} |S_{q,r}| \rceil$, viendo que hay una cantidad suficiente de códigos de longitud hasta l_r

$$\sum_{i=0}^{l_r} |\mathcal{A}_O|^i = \frac{|\mathcal{A}_O|^{l_r+1} - 1}{|\mathcal{A}_O| - 1} \geq \frac{|S_{q,r}|\ |\mathcal{A}_O| - 1}{|\mathcal{A}_O| - 1} \geq \frac{|S_{q,r}|\ |\mathcal{A}_O| - |S_{q,r}|}{|\mathcal{A}_O| - 1} = |S_{q,r}|.$$

Concluimos que

$$\begin{aligned}\sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,c)|} &= \sum_{r\in Q}\sum_{c\in S_{q,r}} |\mathcal{A}_O|^{-|\nu(q,c)|} \\ &\leq \sum_{r\in Q}(1 + \lceil \log_{|\mathcal{A}_O|} |S_{q,r}| \rceil) \\ &\leq |Q| + |Q|\ \lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil.\end{aligned}$$

□

Proposición 3.6. *Para todo ILFSC $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$, $n \in \mathbb{N}$, $w \in {\mathcal{A}_I}^n$, se verifica*

$$|A_C(w)| \leq |C(w)| + n\ \lceil \log_{|\mathcal{A}_O|}(|Q| + |Q|\ \lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)\rceil.$$

Demostración. Usando la proposición 3.5 podemos ver que

$$\begin{aligned}p^*(w) &= \prod_{i=1}^{n} p(\delta^*(w[1..i-1]), w[i]) \\ &= \prod_{i=1}^{n} \frac{|\mathcal{A}_O|^{-|\nu(\delta^*(w[1..i-1]),w[i])|}}{\sum_{c\in\mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(\delta^*(w[1..i-1]),c)|}} \\ &\geq |\mathcal{A}_O|^{-|C(w)|}\ (|Q| + |Q|\ \lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)^{-n}\end{aligned}$$

y entonces

$$\begin{aligned}|A_C(w)| &= \lceil -\log_{|\mathcal{A}_O|} p^*(w) \rceil \\ &\leq |C(w)| + \lceil n\ \log_{|\mathcal{A}_O|}(|Q| + |Q|\ \lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)\rceil.\end{aligned}$$

□

Como hicimos previamente, veamos qué ocurre con las tasas de compresión si usamos esta construcción para autómatas afectados por cambios de alfabeto. Sean C un ILFSC con alfabetos de entrada y salida $\mathcal{A}_I$ y $\mathcal{A}_O$, respectivamente, $k \in \mathbb{N}$, $\alpha \in {\mathcal{A}_I}^\omega$ y $\beta \in ({\mathcal{A}_I}^k)^\omega$

vinculadas por un cambio de alfabeto. Si el conjunto de estados de C es Q, también lo es para C^k y A_{C^k}. Entonces

$$\begin{aligned}\rho_{A_{C^k}}(\beta) &= \liminf_{n\to\infty} \frac{|A_{C^k}(\beta[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}_I|^k} \\ &\leq \liminf_{n\to\infty} \frac{|C^k(\beta[1..n])| + \lceil n \ \log_{|\mathcal{A}_O|}(|Q| + |Q| \ \lceil \log_{|\mathcal{A}_O|} |\mathcal{A}_I|^k \rceil)\rceil}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} \\ &\leq \liminf_{n\to\infty} \frac{|C(\alpha[1..nk])| + \lceil n \ \log_{|\mathcal{A}_O|}(|Q| + |Q| \ \lceil k \ \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)\rceil}{nk \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} \\ &\leq \rho_C(\alpha) + \frac{\log_{|\mathcal{A}_O|}(|Q| + |Q| \ \lceil k \ \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)}{k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|}.\end{aligned}$$

En conclusión, $(A_{C^k})^{-k}$ es un FSAC que alcanza sobre α una tasa de compresión que no supera a la de C en ε, siempre que

$$\frac{\log_{|\mathcal{A}_O|}(|Q| + |Q| \ \lceil k \ \log_{|\mathcal{A}_O|} |\mathcal{A}_I| \rceil)}{k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|} < \varepsilon.$$

Es posible elegir k suficientemente grande para que valga la desigualdad porque la expresión de la izquierda es decreciente en k y tiende a 0.

Estamos en condiciones de mostrar que vale la recíproca del lema 3.2.

Demostración del lema 3.3. Análoga a la prueba del lema 3.2. □

3.2. Equivalencia entre normalidad e incompresibilidad mediante codificadores aritméticos de estados finitos

Teorema 3.7. *Si $\alpha \in \mathcal{A}^\omega$ no es normal entonces existe un FSAC para el que α es compresible.*

Demostración. Por el teorema 1.3, existe $k \in \mathbb{N}$ tal que $\beta \in (\mathcal{A}^k)^\omega$, obtenida a partir de α mediante un cambio de alfabeto, no es simplemente normal. Veamos que existe un FSAC para el que β resulta compresible. La preservación de la compresibilidad bajo cambios de alfabeto indicará que α es compresible por FSAC.

Para definir un FSAC adecuado necesitamos asignarle probabilidades a los símbolos de $\mathcal{A}^k$. Usemos para esto sus frecuencias relativas en β. Como β no es simplemente normal, debe existir $c \in \mathcal{A}^k$ tal que

$$\lim_{n\to\infty} \frac{\mathrm{occ}(c, \beta[1..n])}{n} \neq |\mathcal{A}^k|^{-1}.$$

Aunque este límite no exista, sí existen los límites superior e inferior de la misma expresión y deben ser finitos, porque se trata de una sucesión acotada. Como uno de ellos necesariamente es distinto de $|\mathcal{A}^k|^{-1}$, existe una subsucesión para la cual el límite mencionado converge a $f_c \neq |\mathcal{A}^k|^{-1}$. Llamemos $(i_j)_{j\in\mathbb{N}}$ a la enumeración de los índices de los términos de la sucesión original incluidos en esta subsucesión. A pesar de que sabemos que ahora existe una frecuencia para c en el límite, puede que haya algún $d \in \mathcal{A}^k$ distinto de c tal que

$$\lim_{n\to\infty} \frac{\mathrm{occ}(d, \beta[1..i_n])}{i_n}$$

no exista. En ese caso podemos tomar una nueva subsucesión para la cual este límite exista. Realizamos este refinamiento iterativamente hasta encontrar una sucesión $(i_j^*)_{j\in\mathbb{N}}$ para la cual

$$\lim_{n\to\infty} \frac{\operatorname{occ}(d, \beta[1..i_n^*])}{i_n^*} = f_d$$

para todo $d \in \mathcal{A}^k$.

Usemos las frecuencias de los símbolos en esa subsucesión para determinar las probabilidades del FSAC que queremos construir. Si los símbolos no son equifrecuentes, es esperable lograr la compresión de β con un esquema en el que los símbolos más frecuentes se asocien con códigos más cortos que los dados a aquellos de menor frecuencia. Consideremos entonces el FSAC $A = \langle \{q\}, \mathcal{A}^k, \mathcal{A}^k, \delta, p, q\rangle$, con $\delta(q, d) = q$ y $p(q, d) = f_d$ para todo $d \in \mathcal{A}^k$. Al haber un único estado, la probabilidad de un símbolo es independiente de los caracteres que lo preceden, y entonces tenemos que $p^*(w) = \prod_{d\in\mathcal{A}^k} f_d^{\operatorname{occ}(d,w)}$ para todo $w \in (\mathcal{A}_I{}^k)^*$. Podemos afirmar que

$$\begin{aligned}
\rho_A(\beta) &\le \liminf_{n\to\infty} \frac{|A(\beta[1..n])|}{n \log_{|\mathcal{A}^k|} |\mathcal{A}^k|} \\
&\le \lim_{n\to\infty} \frac{|A(\beta[1..i_n^*])|}{i_n^* \log_{|\mathcal{A}^k|} |\mathcal{A}^k|} \\
&\le \lim_{n\to\infty} \frac{\lceil -\log_{|\mathcal{A}^k|} p^*(q, \beta[1..i_n^*])\rceil}{i_n^*} \\
&\le \lim_{n\to\infty} \frac{-\sum_{d\in\mathcal{A}^k} \operatorname{occ}(d, \beta[1..i_n^*])\ \log_{|\mathcal{A}^k|} f_d}{i_n^*} \\
&\le -\sum_{d\in\mathcal{A}^k} f_d \log_{|\mathcal{A}^k|} f_d < 1
\end{aligned}$$

porque según Shannon [Sha48] $-\sum_{d\in\mathcal{A}^k} f_d \log_{|\mathcal{A}^k|} f_d$ se maximiza cuando todas las frecuencias son iguales, en cuyo caso la suma es 1, pero aquí $f_c \neq |\mathcal{A}^k|^{-1}$.

Como A logra comprimir a β, α es compresible con A^{-k} de acuerdo con el teorema 2.3. □

Teorema 3.8. *Si $\alpha \in \mathcal{A}^\omega$ es normal entonces es incompresible mediante FSAC.*

Demostración. Tomemos un FSAC $A = \langle Q, \mathcal{A}, \mathcal{A}_O, \delta, p, q_0\rangle$ arbitrario y un real $\varepsilon > 0$ arbitrariamente pequeño y mostremos que la tasa de compresión de A sobre α, $\rho_A(\alpha)$, es estrictamente mayor que $(1-\varepsilon)^3$.

En la sección 1.2 dijimos que una cadena $w \in \mathcal{A}^*$ es compresible si su código es menor que $|w| \log_{|\mathcal{A}_O|} |\mathcal{A}|$. Para todo $k \in \mathbb{N}$, llamaremos $\mathcal{W}_k$ al conjunto de cadenas de $\mathcal{A}^k$ que, cuando se quiere codificar con A cualquiera de sus supercadenas, siempre realizan un aporte mayor que $(1-\varepsilon)|w| \log_{|\mathcal{A}_O|} |\mathcal{A}|$ a la longitud del código final, es decir, es el conjunto de cadenas cuya probabilidad estimada no es más alta que $|\mathcal{A}|^{k(\varepsilon-1)}$ en ningún estado.

$$\mathcal{W}_k = \{w \in \mathcal{A}^k : p^*(q, w) < |\mathcal{A}|^{k(\varepsilon-1)} \text{ para todo } q \in Q\}.$$

Estas cadenas son las menos comprimidas por este autómata. Podemos ver que la proporción que representa el conjunto $\mathcal{W}_k$ dentro de $\mathcal{A}^k$ tiende a 1 mientras más grande es k

acotando superiormente la cardinalidad de su complemento. La condición que impide que haya demasiadas probabilidades altas es que su suma no debe exceder 1.

$$\begin{aligned}|\mathcal{A}^k \setminus \mathcal{W}_k| &= |\{w \in \mathcal{A}^k : p^*(q,w) \geq |\mathcal{A}|^{k(\varepsilon-1)} \text{ para algún } q \in Q\}| \\ &\leq |Q| \ \text{máx}\{m \in \mathbb{N} : m \ |\mathcal{A}|^{k(\varepsilon-1)} \leq 1\} \\ &\leq |Q| \ |\mathcal{A}|^{k(1-\varepsilon)}\end{aligned}$$

$$|\mathcal{W}_k| \geq |\mathcal{A}|^k - |Q| \ |\mathcal{A}|^{k(1-\varepsilon)} = |\mathcal{A}|^k(1 - |Q| \ |\mathcal{A}|^{-\varepsilon k})$$

Como $|Q| \ |\mathcal{A}|^{-\varepsilon k}$ tiende a cero a medida que k crece, tomemos k lo suficientemente grande tal que $|\mathcal{W}_k| \geq |\mathcal{A}|^k(1-\varepsilon)$. Mostremos que $\rho_A(\alpha) > (1-\varepsilon)^3$ probando que para $\beta \in (\mathcal{A}^k)^\omega$, vinculada a α por un cambio de alfabeto, $\rho_{A^k}(\beta) > (1-\varepsilon)^3$.

Por el teorema 1.3, la normalidad de α implica que $\beta \in (\mathcal{A}^k)^\omega$, obtenida a partir de α mediante un cambio de alfabeto, es simplemente normal. Existe, entonces, una longitud n_0 tal que, para los prefijos iniciales de β que la superen, las frecuencias relativas de los símbolos ya están lo suficientemente cerca de la equiprobabilidad, es decir,

$$\forall n > n_0, \forall c \in \mathcal{A}^k, \ \frac{\text{occ}(c, \beta[1..n])}{n} > |\mathcal{A}^k|^{-1}(1-\varepsilon).$$

Si $n > n_0$, aún contando solamente la contribución de los símbolos en $\mathcal{W}_k$, es decir, los mayores aportes, las longitudes de los códigos de los prefijos de β de largo n no pueden ser mucho menores que $n \log_{|\mathcal{A}_O|} |\mathcal{A}^k|$.

$$\begin{aligned}|A^k(\beta[1..n])| &= \lceil -\log_{|\mathcal{A}_O|} p^*(\beta[1..n]) \rceil \\ &\geq -\sum_{c \in \mathcal{W}_k} \text{occ}(c, \beta[1..n]) \log_{|\mathcal{A}_O|} |\mathcal{A}|^{k(\varepsilon-1)} \\ &> \sum_{c \in \mathcal{W}_k} n \ |\mathcal{A}^k|^{-1}(1-\varepsilon)(1-\varepsilon) \log_{|\mathcal{A}_O|} |\mathcal{A}^k| \\ &> |\mathcal{W}_k| \ n \ |\mathcal{A}|^{-k}(1-\varepsilon)^2 \log_{|\mathcal{A}_O|} |\mathcal{A}^k| \\ &> (1-\varepsilon)^3 n \log_{|\mathcal{A}_O|} |\mathcal{A}^k|\end{aligned}$$

En consecuencia, la tasa de compresión de A^k sobre β es estrictamente mayor que $(1-\varepsilon)^3$.

$$\rho_{A^k}(\beta) = \liminf_{n \to \infty} \frac{|A^k(\beta[1..n])|}{n \log_{|\mathcal{A}_O|} |\mathcal{A}^k|} > (1-\varepsilon)^3$$

Según el lema 2.4 $\rho_A(\alpha) = \rho_{A^k}(\beta)$, de modo que $\rho_A(\alpha) > (1-\varepsilon)^3$. Como la desigualdad vale para todo $\varepsilon > 0$, $\rho_A(\alpha) = 1$. □

Ya estamos en condiciones de probar el teorema que caracteriza a la normalidad como incompresibilidad mediante autómatas finitos.

Teorema 3.9. *Una secuencia α es normal si y sólo si es incompresible mediante compresores de estados finitos sin pérdida de información.*

Demostración. Los teoremas 3.7 y 3.8 implican que una secuencia es normal si y sólo si es incompresible con codificadores aritméticos de estados finitos, que a su vez equivale a que sea incompresible mediante ILFSC por el teorema 3.1. □

3.3. Teorema de Agafonov

El teorema de Agafonov, originalmente formulado sólo para secuencias binarias, establece otra caracterización de las secuencias normales. La normalidad se preserva en subsecuencias obtenidas mediante selectores de estados finitos.

Teorema 3.10. *Una secuencia $\alpha \in \mathcal{A}^\omega$ es normal si y sólo si $S(\alpha)$ es normal para todo selector de estados finitos S.*

La equivalencia entre normalidad e incompresibilidad mediante FSAC nos permite obtener una demostración sencilla de este teorema. El esquema de la prueba, basado en la demostración de Becher y Heiber del mismo teorema [BH12], consiste en mostrar que si $S(\alpha)$ no es normal para algún selector S entonces podemos construir un codificador aritmético de estados finitos que comprima α, que por lo tanto no es normal. Becher y Heiber usan ILFSC en lugar de FSAC, lo que resulta en una prueba más compleja porque es necesario verificar que el compresor de estados finitos construido no tiene pérdida de información.

En la prueba nos interesa qué proporción de una secuencia infinita es seleccionada por un selector de estados finitos.

Definición. *Si S es un selector de estados finitos y $\alpha \in {\mathcal{A}_I}^\omega$, decimos que la* tasa de selección *de S sobre α es*

$$\rho_S(\alpha) = \liminf_{n\to\infty} \frac{|S(\alpha[1..n])|}{n}.$$

Explotamos además el hecho de que esa magnitud siempre es positiva, un hecho conocido en el área [LS77].

Lema 3.11. *Si S es un selector de estados finitos de k estados con alfabeto de entrada $\mathcal{A}$ y $\alpha \in \mathcal{A}^\omega$, $\rho_S(\alpha) \geq k^{-1}$.*

Demostración. Consideremos la familia de bloques de k que componen α, $(w_i)_{i\in\mathbb{N}}$ con $w_i = \alpha[(i-1)k+1..ik]$. Para cada i, el recorrido del autómata que describe el procesamiento de w_i por S contiene al menos un ciclo. Como los selectores de estados finitos están libres de ciclos sin estados selectores, al menos uno de los estados visitados es selector. Luego, al menos un símbolo de w_i es seleccionado. Esto indica que al menos uno de cada k símbolos consecutivos es seleccionado en el límite. □

Demostración del Teorema 3.10. Sea α una secuencia infinita. Si $S(\alpha)$ es normal para todo selector de estados finitos S, α es normal pues tomamos el selector en el que todos sus estados son selectores.

Probemos la otra dirección del teorema mediante su contrarrecíproca. Dada $\alpha \in \mathcal{A}^\omega$, supongamos que existe un selector de estados finitos $S = \langle Q_S, \mathcal{A}, \delta_S, Q_S, q_{0S}\rangle$ tal que $S(\alpha)$ no es normal. Veamos que entonces α tampoco es normal definiendo un FSAC para el que α resulta compresible.

Como $S(\alpha)$ no es normal, existe un FSAC $A = \langle Q_A, \mathcal{A}, \mathcal{A}, \delta_A, p, q_{0A}\rangle$ que logra comprimir a $S(\alpha)$, es decir, $\rho_A(S(\alpha)) < 1$. Definimos entonces un codificador aritmético de estados finitos $A_S = \langle Q_S \times Q_A, \mathcal{A}, \mathcal{A}, \delta, p', \langle q_{0S}, q_{0A}\rangle\rangle$ que aproveche la compresión que puede lograr el esquema de A sobre los símbolos seleccionados por S. Sobre el resto no

realiza ningún tipo de compresión, pues se utiliza una distribución uniforme cuando el selector indica el descarte de un carácter. Las funciones δ y p' satisfacen

$$\delta(\langle q_S, q_A\rangle, c) = \begin{cases} \langle \delta_S(q_S, c), q_A\rangle & \text{si } q_S \notin Q_S \\ \langle \delta_S(q_S, c), \delta_A(q_A, c)\rangle & \text{si } q_S \in Q_S \end{cases}$$
$$p'(\langle q_S, q_A\rangle, c) = \begin{cases} |\mathcal{A}|^{-1} & \text{si } q_S \notin Q_S \\ p(q_A, c) & \text{si } q_S \in Q_S. \end{cases}$$

Podemos comprobar que el producto de las probabilidades asignadas por A_S a los símbolos de una cadena $w \in \mathcal{A}^*$ seleccionados por S es exactamente la probabilidad otorgada a $S(w)$ por A,

$$p'^*(w) = p^*(S(w))|\mathcal{A}|^{-(|w|-|S(w)|)},$$

que implica

$$|A_S(w)| = -\log_{|\mathcal{A}|} p'^*(w) = |A(S(w))| + |w| - |S(w)|.$$

La tasa de compresión de α con A_S verifica entonces

$$\begin{aligned} \rho_{A_S}(\alpha) &= \liminf_{n\to\infty} \frac{|A(S(\alpha[1..n]))| + n - |S(\alpha[1..n])|}{n} \\ &= \liminf_{n\to\infty} \frac{|A(S(\alpha[1..n]))|}{|S(\alpha[1..n])|}\frac{|S(\alpha[1..n])|}{n} + 1 - \frac{|S(\alpha[1..n])|}{n} \\ &= \liminf_{n\to\infty} 1 - \frac{|S(\alpha[1..n])|}{n}\left(1 - \frac{|A(S(\alpha[1..n]))|}{|S(\alpha[1..n])|}\right). \end{aligned}$$

Sea $(i_n)_{n\in\mathbb{N}}$ una sucesión tal que

$$\lim_{n\to\infty} \frac{|A(S(\alpha[1..i_n]))|}{|S(\alpha[1..i_n])|} = \liminf_{n\to\infty} \frac{|A(S(\alpha[1..n]))|}{|S(\alpha[1..n])|} = \rho_A(S(\alpha)).$$

Refinemos aún más esta sucesión, tomando $(j_n)_{n\in\mathbb{N}}$ tal que $\{j_n : n \in \mathbb{N}\} \subseteq \{i_n : n \in \mathbb{N}\}$ y exista el límite

$$\lim_{n\to\infty} \frac{|S(\alpha[1..j_n])|}{j_n} = \ell_S,$$

que sabemos que satisface $\ell_S \geq k^{-1}$ por el lema 3.11. Considerando en el límite inferior correspondiente a $\rho_{A_S}(\alpha)$ sólo los términos enumerados por esta sucesión, tenemos que

$$\begin{aligned} \rho_{A_S}(\alpha) &= \liminf_{n\to\infty} 1 - \frac{|S(\alpha[1..n])|}{n}\left(1 - \frac{|A(S(\alpha[1..n]))|}{|S(\alpha[1..n])|}\right) \\ &\leq \liminf_{n\to\infty} 1 - \frac{|S(\alpha[1..j_n])|}{j_n}\left(1 - \frac{|A(S(\alpha[1..j_n]))|}{|S(\alpha[1..j_n])|}\right) \\ &\leq 1 - \ell_S(1 - \rho_A(S(\alpha))) \leq 1 - k^{-1}(1 - \rho_A(S(\alpha))) < 1 \end{aligned}$$

porque $k > 0$ y $\rho_A(S(\alpha)) < 1$. Por el teorema 3.8, la compresibilidad de α por A_S indica que esta secuencia no es normal. □

4. CONCLUSIONES Y TRABAJO FUTURO

Hemos presentado una prueba completa de la equivalencia entre normalidad de secuencias e incompresibilidad por compresores de estados finitos sin pérdida de información. Para esto presentamos los codificadores aritméticos de estados finitos y probamos que sus tasas de compresión coinciden con las de los compresores de estados finitos sin pérdida de información. Demostramos también que la compresibilidad se preserva tras cambios de alfabeto, al igual que la normalidad. Finalmente usamos codificadores aritméticos de estados finitos para construir una prueba del teorema de Agafonov.

A pesar de la importancia de contar con una prueba de la caracterización de la normalidad directamente en términos de compresibilidad en lugar de usar martingalas como puente entre ambas nociones, la codificación aritmética de estados finitos también merece atención. Como vimos, su uso puede resultar en pruebas menos complejas que aquellas que dependen de compresores de estados finitos sin pérdida de información, en cuyo caso se suma el trabajo de verificar la factibilidad de la descompresión. En aplicaciones prácticas puede aprovecharse la ventaja de la codificación aritmética por sobre las codificaciones símbolo a símbolo como la de Huffman para obtener códigos más breves que los producidos por compresores de estados finitos con la misma cantidad de estados. Del lado del trabajo teórico, aquí presentamos dos funciones de codificación para un codificador aritmético de estados finitos A, A y $A^\star$, pero no estudiamos esta última, que puede tener propiedades interesantes a pesar de que su tasa de compresión pueda ser estrictamente menor que la de la primera, siendo posible incluso que llegue a ser 0.

Bibliografía

[Aga68] V. N. Agafonov: Normal sequences and finite automata. *Soviet Mathematics Doklady*, 9:324–325, 1968.

[BC01] D. H. Bailey y R. E. Crandall: On the Random Character of Fundamental Constant Expansions. *Exper. Math.*, 10:175–190, 2001.

[BH12] V. Becher y P. Heiber: Normal Numbers and Finite Automata. *Pendiente de publicación*, 2012.

[BHV05] C. Bourke, J. Hitchcock y N. Vinodch: Entropy rates and finite-state dimension. *Theoretical Computer Science*, 349:392–406, 2005.

[Bor09] É. Borel: Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27:247–271, 1909.

[Bug12] Y. Bugeaud: *Distribution Modulo One and Diophantine Approximation*. Series: Cambridge Tracts in Mathematics 193. Cambridge University Press, 2012.

[CE46] A. H. Copeland y P. Erdös: Note on normal numbers. *Bull. Amer. Math. Soc.*, 52:857–860, 1946.

[Cha33] D. G. Champernowne: The Construction of Decimals Normal in the Scale of Ten. *J. London Math. Soc.*, 8:254–260, 1933.

[Chu67] K. L. Chung: *Markov chains with stationary transition probabilities*. Berlin– Göttingen– Heidelberg: Springer, 1967.

[Del93] J. P. Delahaye: *Randomness, Unpredictability and Absence of Order*. En J. P. Dubucs (editor): *Philosophy of Probability*, páginas 145–167. Springer, 1993.

[DLLM04] J. Dai, J. Lathrop, J. Lutz y E. Mayordomo: Finite-State Dimension. *Theoretical Computer Science*, 310:1–33, 2004.

[Eag12] A. Eagle: *Chance versus Randomness*. En E. N. Zalta (editor): *The Stanford Encyclopedia of Philosophy (Spring 2012 Edition)*. 2012.

[HMU07] J. E. Hopcroft, R. Motwani y J. D. Ullman: *Introduction to automata theory, languages, and computation*. Pearson/Addison Wesley, 2007.

[Huf52] D. Huffman: *A Method for the Construction of Minimum-Redundancy Codes*. En *Institute of Radio Engineers*, páginas 1098–1102, 1952.

[Huf59] D. Huffman: Canonical forms for information-lossless finite-state logical machines. *Information Theory, IRE Transactions on*, 5(5):41–59, 1959.

[LS77] R. Lindner y L. Staiger: *Algebraische Codierungstheorie – Theorie der sequentiellen Codierungen*. Akademie-Verlag, Berlin, 1977.

[LV08] M. Li y P. M .B. Vitányi: *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 2008.

[Mac03] D. J. C. MacKay: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[ML66] P. Martin-Löf: The definition of random sequences. *Information and Control*, 9(6):602–619, 1966.

[Nie09] A. Nies: *Computability and Randomness*. Oxford Logic Guides. OUP Oxford, 2009.

[Sai04] A. Said: *Introducing to Arithmetic Coding - Theory and Practice*. HPL-2004-76. Imaging Systems Laboratory, HP Laboratories Palo Alto, 2004.

[Sch73] C. P. Schnorr: Process complexity and effective random tests. *J. Comput. Syst. Sci.*, 7(4):376–388, 1973.

[Sch94] D. Scheinwald: On the Lempel-Ziv proof and related topics. *Proceedings of the IEEE*, 82:866–871, 1994.

[Sha48] C. E. Shannon: A mathematical theory of communication. *Bell System Technical Journal*, 1948.

[SS72] C. P. Schnorr y H. Stimm: Endliche Automaten und Zufallsfolgen. *Acta Informatica*, 1:345–359, 1972.

[WNC87] I. H. Witten, R. M. Neal y J. G. Cleary: Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540, 1987.