De Normalidad a Incompresibilidad vía Codificación Aritmética

Trabajo Final de la Licenciatura en Ciencias de la Computación

Facundo López Bristot[1]
Director: Pablo Ariel Heiber[1,2]

[1] Departamento de Computación,
Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires
[2] CONICET, Argentina

Resumen. En este trabajo damos una prueba completa de la caracterización de las secuencias normales como aquellas incompresibles mediante ciertos autómatas de estados finitos con output, los 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.

1. Introducción

Borel definió en 1909 [Bor09] la normalidad utilizando un enfoque combinatorio, tomando como normales a las secuencias que están estadísticamente balanceadas, una propiedad esperada en una secuencia aleatoria. Por otro lado, Kolmogorov caracterizó a las secuencias aleatorias como aquellas cuyos prefijos iniciales son incompresibles por compresores computables [LV08, Nie09]. En este trabajo reforzamos el vínculo entre las nociones de aleatoriedad de Kolmogorov y normalidad, esta última una forma débil de aleatoriedad, mostrando que ambas se pueden definir en términos de incompresibilidad.

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. De la conjunción de resultados en [SS72, DLLM04, BHV05] se puede concluir que una secuencia es incompresible mediante compresores de estados finitos si y sólo si su *dimensión de estados finitos* es 1, una condición que a su vez equivale a normalidad. Becher y Heiber [BH12] formularon en 2012 una prueba elemental y directa de la

relación entre normalidad e incompresibilidad. 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. Luego mostramos 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. En el Apéndice B mostramos cómo ese resultado puede usarse para construir 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. Los Apéndices A y C contienen algunas demostraciones y una breve reseña sobre la codificación aritmética, respectivamente.

2. 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 $\mathrm{occ}(v, w) = |\{i : v = w[i..i + |v| - 1]\}|$.

Normalidad. Comencemos recordando la definición de normalidad.

Definición 1 (Definición 4.1, páginas 87–88, y Teorema 4.5, páginas 91–93, [Bug12]). *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 $w \in \mathcal{A}^*$*

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

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 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 3 (Teorema 4.2, [Bug12], páginas 88–89). *La 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.*

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 4. *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 y $q_0 \in Q$ 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 $C(w) = \nu^*(w)$.

Restringiremos nuestro estudio a compresores en los que es posible la descompresión.

Definición 5. *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.*

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 $|C(w)| < |w| \log_{|\mathcal{A}_O|} |\mathcal{A}_I|$.

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 6. *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. 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.

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 7. *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, la función de transición es* $\delta : Q \times \mathcal{A} \to Q$, $Q_S \subset Q$ *es el conjunto de estados selectores,* $q_0 \in Q$ *es el estado inicial y 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 $S(w) = \sigma_S(q_0, w)$.

Codificación Aritmética. La codificación aritmética [WNC87, Sai04, Mac03] es una técnica de compresión que garantiza códigos de longitud óptima con respecto al conjunto de todas las codificaciones unívocamente decodificables posibles. Aquí nos limitaremos a fijar la notación para los elementos necesarios para construir un codificador aritmético dado un modelo probabilístico de la fuente. En el Apéndice C incluimos una descripción sobre esta técnica de compresión.

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

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

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. Llamaremos $\Phi(x_1x_2 \dots x_n)$ al subintervalo asignado al mensaje $x_1x_2 \dots x_n$.

3. Codificadores Aritméticos de Estados Finitos

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 de que el próximo símbolo emitido por la fuente x_j sea c sabiendo que los últimos símbolos vistos fueron $x_1, x_2, \ldots, x_{j-1}$, notada $P(x_j = c|x_1 x_2 \ldots x_{j-1})$. 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 8. *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* $c \in \mathcal{A}_I$*, y* $q_0 \in Q$ *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

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

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 de codificación aritmética 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 vigentes en el uso general de esta técnica de compresión. 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.

Codificación. En adelante supondremos que $\mathcal{A}_O = \{0, 1, 2, \ldots, |\mathcal{A}_O| - 1\}$ para simplificar la notación y con *lex* denotaremos el orden lexicográfico.

Definición 9. *Sea $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ un codificador aritmético de estados finitos. La función de codificación $A : {\mathcal{A}_I}^* \to {\mathcal{A}_O}^*$ satisface*

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

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 $w \in {\mathcal{A}_I}^*$.

Definición 10. *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*

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

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 $\rho_{FSA}(\alpha) < 1$.

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. Su demostración, al igual que las del resto de los enunciados de esta sección, se encuentra en el Apéndice A.

Proposición 11. *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 $k \in \mathbb{N}$,*

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

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 12. *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.

Definición 13. *Sea $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ un FSC. Definimos el compresor de estados finitos $C^k = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta', \nu', q_0\rangle$ como aquel tal que*

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

Definición 14. *Si $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ es un codificador aritmético de estados finitos, $A^k = \langle Q, {\mathcal{A}_I}^k, \mathcal{A}_O, \delta', p', q_0\rangle$ 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. Podemos entonces probar el siguiente lema.

Lema 15. *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, $\rho_D(\alpha) = \rho_{D^k}(\beta)$.*

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 11 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 16. *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 17. *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 cadena $w \in ({\mathcal{A}_I}^k)^*$. 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 $w \in ({\mathcal{A}_I}^k)^*$ y con ella podemos probar el lema restante para poder demostrar la invariancia de la compresibilidad bajo cambios de alfabeto.

Proposición 18. *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 $w \in ({\mathcal{A}_I}^k)^*$, $p'^*(\langle q, \lambda\rangle, w) = p^*(q, w)$.*

Lema 19. *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, $\rho_D(\beta) = \rho_{D^{-k}}(\alpha)$.*

4. Resultados Principales

Igualdad de Tasas de Compresión de Compresores de Estados Finitos sin Pérdida de Información y Codificadores Aritméticos de Estados Finitos. Probamos el siguiente teorema, factorizando la demostración en dos partes de la manera usual.

Teorema 20. *$\rho_{FS}(\alpha) = \rho_{FSA}(\alpha)$ para toda secuencia α.*

Lema 21. *$\rho_{FS}(\alpha) \leq \rho_{FSA}(\alpha)$ para toda secuencia α.*

Lema 22. *$\rho_{FS}(\alpha) \geq \rho_{FSA}(\alpha)$ 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 23. *Si $A = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ es un FSAC, existe una función $f : Q \times \mathcal{A}_I \to {\mathcal{A}_O}^+$ tal que $\{f(q,c) : c \in \mathcal{A}_I\}$ es libre de prefijos y se cumple $|f(q,c)| = |A(q,c)|$ para todo $c \in \mathcal{A}_I$, $q \in Q$.*

Usando esta proposición, demostrada en el Apéndice A, podemos definir un ILFSC a partir de un FSAC de modo que generen códigos de la misma longitud para símbolos individuales.

Definición 24. *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 23. 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 $\nu(q,c) = f(q,c)$.*

En el Apéndice A probamos que C_A tiene pérdida de información y que los códigos que produce son más largos que los de A pero su diferencia está acotada.

Proposición 25. *Si A es un FSAC, C_A es un ILFSC. Además, si el alfabeto de entrada de A es $\mathcal{A}_I$, $n \in \mathbb{N}$ y $w \in {\mathcal{A}_I}^n$, entonces $|C_A(w)| \leq n + |A(w)|$.*

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, usada para probar el Lema 21 en el Apéndice A, 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 $(k \log_{|\mathcal{A}_O|} |\mathcal{A}_I|)^{-1} < \varepsilon$.

Veamos ahora cómo probar el Lema 22. 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 26. *Sea $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ un ILFSC. El codificador aritmético de estados finitos $A_C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, p, q_0\rangle$ es aquel 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. Ocurre lo contrario si $\sum_{d\in\mathcal{A}_I} |\mathcal{A}_O|^{-|\nu(q,d)|} < 1$, en cuyo caso 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 27. *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,*

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

para todo estado q.

Usando esta proposición podemos probar la siguiente. Sus demostraciones figuran en el Apéndice A.

Proposición 28. *Para todo ILFSC $C = \langle Q, \mathcal{A}_I, \mathcal{A}_O, \delta, \nu, q_0\rangle$ con todos sus estados alcanzables desde q_0, $n \in \mathbb{N}$, $w \in {\mathcal{A}_I}^n$, se verifica*

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

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 21.

Demostración (Lema 22). Análoga a la prueba del Lema 21. □

Equivalencia entre Normalidad e Incompresibilidad mediante Codificadores Aritméticos de Estados Finitos.

Teorema 29. *Si $\alpha \in \mathcal{A}^\omega$ no es normal entonces existe un FSAC para el que α es compresible.*

Demostración. Por el Teorema 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

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

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 codificador aritmético de estados finitos $A = \langle\{q\}, \mathcal{A}^k, \mathcal{A}^k, \delta, p, q\rangle$, 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 $p^*(w) = \prod_{d\in\mathcal{A}^k} f_d^{\mathrm{occ}(d,w)}$ para todo $w \in ({\mathcal{A}_I}^k)^*$. Podemos afirmar que

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

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 12. □

Teorema 30. *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 $(1-\varepsilon)^3$.

Recordemos 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 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 15 $\rho_A(\alpha) = \rho_{A^k}(\beta)$, 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 31. *Una secuencia α es normal si y sólo si es incompresible mediante compresores de estados finitos sin pérdida de información.*

Demostración. Los Teoremas 29 y 30 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 20. □

5. Conclusiones y Trabajo Futuro

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 se ve en la demostración del Teorema de Agafonov, 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, sería interesante estudiar como función de codificación alternativa a la siguiente,

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

Esta función elige entre los posibles códigos el de menor longitud y, en caso de que haya varios, prefiere el lexicográficamente menor. Las tasas de compresión alcanzables con este esquema de codificación pueden ser estrictamente menores que las que se obtienen con la función usada a lo largo de este trabajo, siendo posible incluso conseguir tasas de compresión igual a 0.

Apéndice A: Demostraciones

Demostración (Proposición 11). 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 $m = \text{máx}\{|\nu(q,w)| : q \in Q \ \wedge \ w \in ({\mathcal{A}_I}^k)^*\}$ y si es un codificador aritmético de estados finitos, $m = \text{máx}\{\lceil -\log_{|\mathcal{A}_O|} p^*(q,w) \rceil : q \in Q \ \wedge \ w \in ({\mathcal{A}_I}^k)^*\}$. 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

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

□

Demostración (Lema 15). Sea D un FSC o un FSAC con alfabeto de entrada $\mathcal{A}_I$. En virtud de la Proposición 11

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

□

Demostración (Proposición 18). Probemos por inducción en n que para $n \leq k$ y $w \in {\mathcal{A}_I}^n$, $p'^*(\langle q, \lambda \rangle, w) = \sum_{v \in {\mathcal{A}_I}^{k-|w|}} p(q, wv)$.

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,

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

□

Demostración (Lema 19). 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 11 nos permiten deducir que

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

□

Demostración (Teorema 12). Sean $\alpha \in {\mathcal{A}_I}^\omega$ y $\beta \in ({\mathcal{A}_I}^k)^\omega$. Por el Lema 15

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

Por otro lado, según el Lema 19

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

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

Demostración (Proposición 23). 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 $[0, 1)$,

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

□

Demostración (Proposición 25). 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 $C_A(v)[|C_A(v[1..i-1])| + j] \neq C_A(w)[|C_A(v[1..i-1])| + j]$.

Además, si $w \in {\mathcal{A}_I}^n$,

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

□

Demostración (Lema 21). Si $\alpha \in {\mathcal{A}_I}^\omega$, de acuerdo con la invariancia de la compresibilidad por cambios de alfabeto (Teorema 12),

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

□

Demostración (Proposición 27). Fijemos q en un estado arbitrario. Si notamos $S_{q,r} = \{c \in \mathcal{A}_I : \delta(q, c) = r\}$, 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

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

□

Demostración (Proposición 28). Usando la Proposición 27 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

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

□

Apéndice B: 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 32. *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 33. *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 34. (Demostración en el Apéndice A) *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 (Teorema 32). 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 $S = \langle Q_S, \mathcal{A}, \delta_S, Q_S, q_{0S}\rangle$, un selector de estados finitos 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 34. 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 30, la compresibilidad de α por A_S indica que esta secuencia no es normal. □

Apéndice C: 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.

Como mencionamos en la sección 2, para poder implementar un codificador aritmético se necesita estimar, para todo símbolo c, la probabilidad de que el próximo carácter emitido por la fuente x_j sea c sabiendo que los últimos vistos fueron $x_1, x_2, \ldots, x_{j-1}$, notada $P(x_j = c|x_1x_2 \ldots x_{j-1})$. Dado un modelo de esas características y fijando una enumeración $c_1, c_2, \ldots, c_n$ de los símbolos de la fuente, definimos las probabilidades condicionales acumuladas

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

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

Este método 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 $c \in \mathcal{A}$

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

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.

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.

Referencias

[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.

[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.

[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.

[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.