Notas de
Algoritmos, Azar y Autómatas

Manuel Panichelli

October 17, 2021

Chapter 1

Normalidad

1.1 Azar

Azar es **imposibilidad de predecir**, **falta de patrones**, imposibilidad de abreviar, comprimir.

Vamos a categorizar el azar según diferentes modelos de cómputo

- Autómatas finitos
- Autómatas de pila
- Máquinas de turing

Def. 1.1. Una secuencia es **azarosa** (para los autómatas de la clase C) cuando, esencialmente, la única forma de describirla (mediante un autómata de la clase C) es nombrando explícitamente cada uno de sus símbolos.

Esto quiere decir que no tiene patrones (porque sino podríamos nombrar menos) y que no se puede comprimir. *Esencialmente* porque se pueden hacer pequeñas conversiones. Por ejemplo, las cadenas de $\{a^n b^n \mid n \in \mathbb{N}\}$ son azarosas para AF pero no para AP (porque es un lenguaje libre de contexto pero no regular).

Hay distintos *grados de azar*:

1. **Azar puro**: Impredecibilidad / incompresibilidad para máquinas de turing
2. **Azar básico**: Impredicibilidad / incompresibilidad para autómatas finitos.

1. Una secuencia es **random** si, esencialmente, sus *segmentos iniciales* solo se pueden describir explícitamente por una Turing Machine (no pueden ser comprimidos por una TM)

2. Una secuencia es **normal** si, esencialmente, sus segmentos iniciales solo se pueden describir explicitamente por un autómata finito.

Cosas que no copié

1. Kolmogorov / program size complexity
2. Definicion de azar de Chaitin basado en kolmogorov
3. Martin Löf random

1.2 Numeros normales

Def. Una **base** es un entero ≥ 2. Para un $x \in \mathbb{R}$ en el intervalo unitario[1], su **expansión** en base b es una **secuencia** $a_1 a_2 a_3 \dots$ de enteros de $0, 1, \dots, b-1$ tales que

$$x = 0.a_1 a_2 a_3 \dots,$$

donde $x = \sum_{k \geq 1} \frac{a_k}{b_k}$ y x no termina con una cola de $b-1$ (esto lo hacemos para tener una representación única de todos los numeros racionales)

Cuando se de por sentada la base b denotamos los primeros n digitos de la expansión de x con $x[1 \dots n]$

Def. 1.2 (Números normales, Borel 1909)**.** Un número real x es,

- **Simplemente normal a base** b si en la expansión de x en base b, cada digito ocurre con una frecuencia de $1/b$ en el límite.

  $$\lim_{n \to \infty} \frac{|x[1 \dots n]|_d}{n} = \frac{1}{b}$$

  (En el límite todos los símbolos tienen la misma frecuencia)

- **Normal a base** b si para cada entero positivo k, cada bloque de k digitos (arrancando de cualquier posición) ocurre en la expansión de x en base b con una frecuencia en el límite de $1/b^k$

- **Absolutamente normal** si es normal para todas las bases.

Ejemplo. Algunos ejemplos son

- 0.01 002 0003 00004 000005 0000006 00000007 000000008 ... no es simplemente normal a base 10 (el 0 tiene más frecuencia que el resto)
- 0.0123456789 0.0123456789 0.0123456789 0.0123456789 ... es simpelemente normal a base 10, pero no es simplemente normal a base 100.

  Pasar de base 10 a base 100 es tomar combinaciones de dos dígitos en base 10 de forma contigua

[1] El intervalo unitario es el intervalo cerrado $[0, 1]$

- El ternario de cantor no es simplemente normal a base 3 (las expansiones no tienen el dígito 1)
- Los numeros racionales no son normales a ninguna base

  Si agarro un número racional, por ej 3.14

$$3.14 \rightsquigarrow 3.140000000\ldots$$

  en base 10 tiene un período que se repite
- La constante de Liouville $\sum_{n\geq 1} 10^{-n!}$ no es normal a base 10

Teorema 1.1 (Borel 1909)**.** Casi todos los números reales son absolutamente normales.

Son las constantes matemáticas usuales como π, e o $\sqrt{2}$ absolutamente normales? O al menos simplemente normales a alguna base? Es una pregunta abierta.

Teorema 1.2 (Champernowne, 1933)**.** Todos los numeros naturales en base 10 concatenados es normal a base 10.

$$0.12345678910111213141516171819202 1\ldots$$

No se sabe si es normal a bases que no son potencias de 10

Teorema 1.3 (Cassels 1959; Schmidth 1961)**.** Casi todos los números del ternario de Cantor son normales a base 2.

Teorema 1.4 (Bailey y Borwein 2012)**.** El número de Stoneham $\alpha_{2,3} = \sum_{k\geq 1} \frac{1}{3^k 2^{3^k}}$ es normal a base 2 pero no simplemente normal a base 6.

1.2.1 Normalidad y autómatas finitos

Def. 1.3. Una secuencia $x = a_1 a_2 a_3 \ldots$ es **compresible** por un trasductor finito T si y solo si en la corrida en T $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \ldots$ satisface que

$$\liminf_{n\to\infty} \frac{|v_1 v_2 \ldots v_n|}{n} < 1.$$

Recordar que los a son símbolols y los v cadenas, posiblemente vacías.

Teorema 1.5. Una secuencia es **normal** si y solo si es **incompresible por todo one-to-one transducer**.

Teorema (Becher, Casrton, Heiber 2013)**.** Los transductores finitos uno a uno no deterministicos con contadores no pueden comprimir secuencias normales.

Teorema.

Los trasductores de pila no determinísticos pueden comprimir secuencias normales.

0123456789 9876543210 00 01 02 03 . . . 98 99 99 98 97 . . . 03 02 01 00 000 001 002 . . .

Va pusheando y cuando detecta el cambio empieza a desapilar. Parecido al APD que reconoce $w\#w^r$

1.3 Notación

- Un *alfabeto* es un conjunto finito de símbolos. Por ej A
- A^ω es el conjunto de todas las palabras infinitas
- A^* (la clausura de Kleene) es el conjunto de todas las palabras finitas
- $A^{\leq k}$ es el conjunto de todas las palabras de longitud hasta k
- A^k es el conjunto de palabras de longitud exactamente k.
- Si w es una cadena $|w|$ es su longitud.
- Las posiciones de las cadenas se numeran desde 1
- $w[i]$ es el simbolo iésimo de w y $w[i \dots j]$ es el substring de i a j.
- La cadena vacía es λ

Def (Ocurrencia en cadena)**.** Decimos que una palabra u *ocurre* en una cadena en una posición i si $w[i \dots i + |u| - 1] = u$. (*omitimos decir ambas posiciones para las ocurrencias*)

Def. 1.4 (Ocurrencias alineadas y no alineadas)**.** El número de ocurrencias alineadas y no alineadas de una cadena es

$$|w|_u = |\{i : w[i \dots i + |u| - 1] = u\}|,$$
$$||w||_u = |\{i : w[i \dots i + |u| - 1] = u \text{ y } i \equiv 1 \bmod |u|\}|$$

Cuando u es un símbolo las definiciones coinciden. Y la de alineadas son posiciones que son múltiplos de $|u|$

(La de alineadas tiene $\equiv 1$ *en vez de* $\equiv 0$ *ya que las posiciones se numeran de 1)*

Ejemplo. $|aaaaa|_{aa} = 4$ y $||aaaaa||_{aa} = 2$.

Prop. Las ocurrencias alineadas de una palabra de longitud r sobre un alfabeto A coinciden con las ocurrencias del símbolo correspondiente sobre el alfabeto A^r.

Proof. Sean un alfabeto A, una longitud r y un alfabeto B con $|A|^r$ símbolos (la cantidad de símbolos que tiene el alfabeto A^r). A^r (el conjunto de palabras de longitud r sobre el alfabeto A) y B son isomorfos, existe

$$\pi : A^r \to B$$

que se induce del orden lexicográfico en cada conjunto (se puede hacer un matching 1 a 1). Por lo tanto, para cada $w \in A^*$ tal que $|w|$ es múltiplo de r,

$$|\pi(w)| = |w|/r.$$

(Una palabra de longitud múltiplo de r es una cadena de r símbolos de A^r, luego la longitud de la palabra en B que tiene símbolos unitarios digamos es esa).

Luego,

$$\forall u \in A^r \ (||w||_u = |\pi(w)|_{\pi(u)}).$$

□

Por ejemplo, sean $A = \{0, 1\}$, $r = 3$, y B tal que $|A^r| = |B|$,

$$B = \{ \underset{000}{0}, \underset{001}{1}, \underset{010}{2}, \underset{011}{3}, \underset{100}{4}, \underset{101}{5}, \underset{110}{6}, \underset{111}{7} \}$$

Luego la cadena,

$$100\ 100\ 111\ 000$$

$$4470$$

La cantidad de ocurrencias de 100 coinciden con las de 4.

Def. 1.5 (Normalidad no alineada, Borel)**.** Un número real x es **normal a base *b*** si para cada bloque u,

$$\lim_{n\to\infty} \frac{|x[1\dots n]|_u}{n} = \frac{1}{b^{|u|}}.$$

En el límite,como $b^{|u|}$ son todos los bloques posibles de longitud $|u|$, $1/b^{|u|}$ seria que cada uno tiene la misma frecuencia.

Teorema 1.6 (Piatetski-Shapiro)**.** Sea x un número real, $b \geq 2$ un entero y $A = \{0, \dots, b-1\}$. Las siguientes son equivalentes

1. x es normal a base b
2. Existe una constante C tal que para infinitas longitudes ℓ y para todo $w \in A^\ell$

$$\limsup_{n\to\infty} \frac{||x[1\dots n\ell]||_w}{n} < C \cdot b^{-\ell}.$$

3. Existe una constante C tal que para infinitas longitudes ℓ y para todo $w \in A^\ell$

$$\limsup_{n\to\infty} \frac{|x[1\dots n]|_w}{n} < C \cdot b^{-\ell}.$$

Dem 3 $\Rightarrow$ 1. Para el ejercicio 4 hay que hacer la 2da.

Sea x un número real. Queremos llegar a Non-aligned normality, para cada bloque u

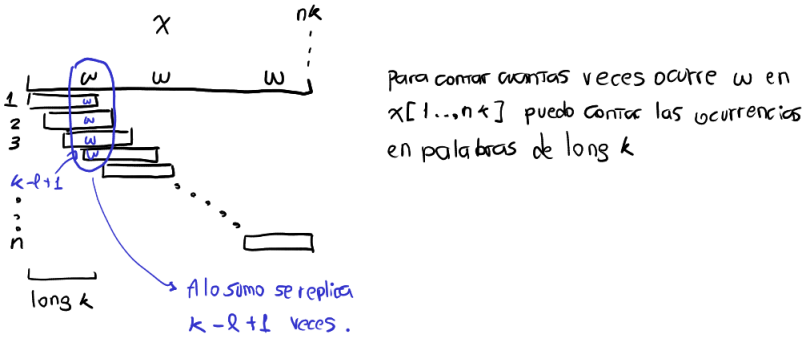
$$\lim_{n\to\infty} \frac{|x[1\dots n]|_u}{n} = \frac{1}{|u|}.$$

Sabemos que se cumple 3, es decir que existe una constante C tal que para toda longitud ℓ y $w \in A^\ell$,

$$\limsup_{n\to\infty} \frac{|x[1\dots n]|_w}{n} < C \cdot b^{-\ell}.$$

Observemos que para todo $w \in A^*$, n y k, para contar las ocurrencias de w en $x[1\dots nk]$ puedo contar las ocurrencias en todas las sub palabras de longitud k. En total puede ocurrir a lo sumo $k-\ell+1$ veces.

$$|x[1\dots nk]|_w \geq \frac{1}{k-\ell+1} \sum_{v\in A^k} |x[1\dots nk]|_v |v|_w.$$



Como sabemos algo del lim sup, y queremos usar Pequeño truco de límites, arrancamos con lim inf.

$$\begin{aligned}
\liminf_{n\to\infty} \frac{|x[1\dots nk]_w}{nk} &\geq \liminf_{n\to\infty} \frac{1}{k-\ell+1} \sum_{v\in A^k} \frac{|x[1\dots nk]|_v}{nk} |v|_w \\
&\geq \liminf_{n\to\infty} \frac{1}{k} \sum_{v\in A^k} \frac{|x[1\dots nk]|_v}{nk} |v|_w \qquad (\ell+1\geq 0) \\
&= \liminf_{n\to\infty} \sum_{v\in A^k} \frac{|x[1\dots nk]|_v}{nk} \frac{|v|_w}{k}
\end{aligned}$$

Supongamos que existe C tal que para longitudes infinitas ℓ y para todo $w \in A^\ell$

$$\limsup_{n\to\infty} \frac{|x[1\dots n]|_w}{n} < C \cdot b^{-\ell}.$$

Sea k una de esas longitudes. Fijamos $\epsilon \leq 1/b^\ell$, $w \in A^\ell$. Sea k suficientementer grande para que $|Bad(A,k,w,\epsilon)| < b^k\epsilon$

$$\begin{aligned}
\liminf_{n\to\infty} \sum_{v\in A^k} \underbrace{\frac{|x[1\dots nk]|_v}{nk} \frac{|v|_w}{k}}_{K} &= \liminf_{n\to\infty} \sum_{v\in A^k\setminus Bad(A,k,w,\epsilon)} K \quad \text{(separo en bad y no bad)} \\
&\quad + \liminf_{n\to\infty} \sum_{v\in A^k\cap Bad(A,k,w,\epsilon)} K \\
&\geq \liminf_{n\to\infty} \sum_{v\in A^k\setminus Bad(A,k,w,\epsilon)} K \quad \text{(no cuento las malas)}
\end{aligned}$$

Y como $v \notin Bad(A,k,w,\epsilon) \Rightarrow \frac{|v|_w}{k} \geq b^{-|w|} \pm \epsilon$

$$\begin{aligned}
\liminf_{\substack{n\to\infty \\ v\in A^k\setminus Bad(A,k,w,\epsilon)}} \sum \frac{|x[1\dots nk]|_v}{nk} \frac{|v|_w}{k} &\geq (1-\epsilon)b^{-\ell} \liminf_{\substack{n\to\infty \\ v\in A^k\setminus Bad(A,k,w,\epsilon)}} \sum \frac{|x[1\dots nk]|_v}{nk} \\
&= (1-\epsilon)b^{-\ell} \liminf_{n\to\infty} \left(1 - \sum_{v\in Bad(A,k,w,\epsilon)} \frac{|x[1\dots nk]|_v}{nk}\right) \quad (\textstyle\sum \text{good} = 1 - \sum \text{bad}) \\
&\geq (1-\epsilon)b^{-\ell} \left(1 - \sum_{v\in Bad(A,k,w,\epsilon)} \limsup_{n\to\infty} \frac{|x[1\dots nk]|_v}{nk}\right) \\
&\geq (1-\epsilon)b^{-\ell} \left(1 - \sum_{v\in Bad(A,k,w,\epsilon)} C\cdot b^{-k}\right) \\
&\geq (1-\varepsilon)b^{-\ell}(1-C\varepsilon) \qquad (|Bad(A,k,w,\epsilon)| < b^k\epsilon)
\end{aligned}$$

Y como es cierto para todo $\varepsilon \geq b^{-\ell}$ positivo, podemos tomar un ε suficientemente chico para que $(1-\varepsilon)b^{-\ell}(1-C\varepsilon) \approx b^{-\ell}$

$$\liminf_{n\to\infty} \frac{|x[1\ldots nk]|_w}{nk} \geq b^{-\ell}.$$

Finalmente, por el lemma Pequeño truco de límites y como es cierto para todo $w \in A^\ell$,

$$\lim_{n\to\infty} \frac{|x[1\ldots n]|_w}{n} = b^{-\ell}$$

y concluyo que x es normal. □

1.4 Tres secuencias normales

- Secuencias de Bruijn infinitas
- A la Champernowne (binario)

$$01 \;\; 00\ 01\ 10\ 11 \;\; 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111 \;\; 0000\ldots$$

- Una secuencia normal tal que la subsecuencia en las posiciones pares es idéntica a toda la secuencia.

1.5 De Bruijn

Def. 1.6 (De Bruijn 1946)**.** Definiciones de De Bruijn,

- Un **collar de De Bruijn** de orden n sobre un alfabeto A es una secuencia cíclica de longitud $|A|^n$ tal que cada palabra de longitud n ocurre en ella exactamente una vez.

  Ejemplos: 01; 0011; 00011101 (el 100 está en la pos 8 por ej.).

- Una **palabra de De Bruijn** (no cíclica) de orden n sobre el alfabeto A es una palabra de longitud $|A|^n + n - 1$ (se le agrega todo lo que uno podría hacer con un ciclo, desde la última posición una palabra con longitud n podría llegar hasta $n-1$ más al principio) tal que cada palabra de longitud n ocurre en ella exactamente una vez.

  Ejemplos: 01; 00110; 0001110100.

- Una **palabra infinita de De Bruijn** $w = a_1 a_2 \ldots$ en un alfabeto de al menos tres símbolos es una palabra infinita tal que,

  $$\forall n.\ a_1 \ldots a_{|A|^n+n-1}$$

  es una palabra de De Bruijn de orden n.

  Ejemplo: 012, una palabra de De Bruijn de orden 1, se puede extender a la siguiente de orden 2: 0122002110.

Si el alfabeto tiene dos símbolos, una palabra infinita de De Bruijn $w = a_1 a_2 \ldots$ es aquella que para cada n impar, $a_1 \ldots a_{|A|^n+n-1}$ es una palabra de De Bruijn de orden n.

Def. 1.7. Un **grafo de De Bruijn** $G_A(n)$ es un digrafo cuyos vértices son palabras de longitud n sobre el alfabeto A y sus ejes los pares (au, ub) para alguna palabra u de longitud $n-1$ y posiblemente dos símbolos diferentes a, b.

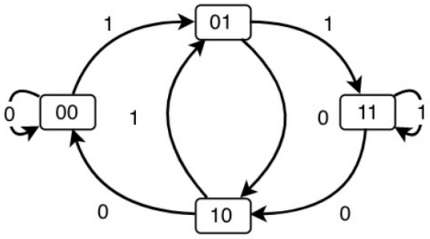


Figure 1.1: Ejemplo de grafo de De Bruijn de orden 2 para $A = \{0, 1\}$

- Tiene $|A|^n$ vertices y $|A|^{n+1}$ arcos
- Es *fuertemente conexo* (existe un camino dirigido entre todo par de vértices)[2]
- Es *regular*, $\forall v. d_{in}(v) = d_{out}(v)$ (los loops suman uno a la entrada y salida)
- Es Euleriano (por teorema de Euler, solo hace falta que sea regular y fuertemente conexo).

[2] Conexo a secas en digrafos es que el grafo subyacente (sacándole direcicones) sea conexo

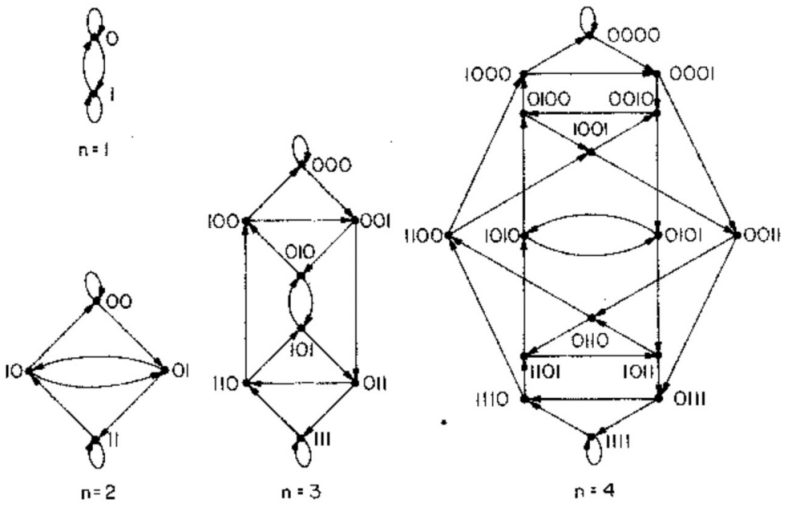


Fig. 3. The de Bruijn graphs of order $n = 1, 2, 3$, and 4.

Figure 1.2: Grafos de De Bruijn de ordenes 1, 2, 3 y 4 sobre $A = \{0, 1\}$

Def. El **grafo de línea** de un grafo G, es otro grafo que tiene como vértilos ejes de G y como ejes los caminos de longitud 2.

Prop. 1.1. Toda secuencia de De Bruijn de orden $n + 1$ sobre un alfabeto de $|A|$ simbolos se puede construir como un ciclo Euleriano en $G_A(n)$.

Prop. 1.2 (Becher, Heiber 2011)**.** Dado un alfabeto A con al menos tres símbolos, toda secuencia de De Bruijn de orden n se puede extender a una de orden $n + 1$

Proof. Dado un alfabeto A, suponiendo que E es un ciclo Euleriano de $G_A(n)$. Como $G_A(n + 1)$ es el grafo de línea de $G_A(n)$, E es un ciclo Hamiltoniano en $G_A(n + 1)$.

Todo ciclo euleriano va a ser hamiltoniano en el grafo de línea, porque los vertices son los ejes

Está la demo completa en las clases, no la terminé de ver.

□

Para computar una palabra infinita de De Bruijn puedo para cada $n \geq 1$ extender un ciclo Hamiltoniano en un grafo de De Bruijn de orden n a uno Euleriano en el mismo grafo. Esto se hace en tiempo exponencial de n, y no se conoce ningun algoritmo eficiente.

Teorema 1.7 (Ugalde 2000)**.** Las palabras infinitas de De Bruijn son normales.

Si el alfabeto A tiene dos símbolos, se puede considerar el alfabeto A' de 4 símbolos que se obtiene con el morfismo que mapea bloques de dos simbolos en A a un simbolo en A' y probar normalidad ahí.

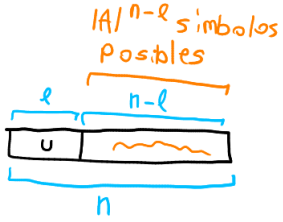
Dem. Intuitivamente, una secuencia es normal si cada bloque de dígitos ocurre con la misma frencuencia en el límite que cada otro bloque de la misma longitud (ver Def Números normales, Borel 1909). Para probarlo, el numero de ocurrencias en una posición arbitraria está acotado por el numero de ocurrencias al final del megabloque.

Quiero ver que las palabras infinitas de De Bruijn son normales. Sea ℓ una longitud cualquiera, $u \in A^\ell$ un bloque de esa longitud y $n > |A|^\ell + \ell - 1$ (u pertenecerá a una palabra de De Bruijn de orden ℓ, que tiene longitud $|A|^\ell + \ell - 1$).

u ocurre en una palabra de De Bruijn de orden n entre $|A|^{n-\ell}$ y $|A|^{n-\ell}+n-\ell$ veces

- Aparece al menos $|A|^{n-\ell}$ veces porque como en una palabra de De Bruijn de orden n aparecen todas las cadenas de tamaño n una vez, el bloque u va a aparecer con $|A|^{n-\ell}$ terminaciones distintas.

  En otras palabras, hay exactamente $|A|^{n-\ell}$ palabras de longitud n que tienen como primeros ℓ símbolos a u.



- A lo sumo $|A|^{n-\ell} + n - \ell$ porque hay exactamente $n - \ell$ posiciones en una palabra de De Bruijn de orden n en las cuales podría comenzar una palabra de longitud ℓ. no me termina de quedar claro

Sea $x = a_1 a_2 \dots$ una palabra infinita de De Bruijn sobre A. Por definición, para cada n

$$a_1 \dots a_{|A|^n+n-1}$$

es una palabra de De Bruijn de orden n. Fijemos N una posición en la palabra infinita que esté entre el final de la palabra de De Bruijn de orden n y $n+1$,

$$|A|^n + n - 1 \leq N < |A|^{n+1} + n.$$

Luego,

$$\begin{aligned}\frac{|a_1 \dots a_N|_u}{N} &\leq \frac{|a_1 \dots a_{|A|^{n+1}+n}|_u}{|A|^n + n - 1} && \text{(Por la cota que define N)}\\ &\leq \frac{|A|^{n+1-\ell} + n - 1}{|A|^n + n - 1} && \text{(\#ap de u en De Bruijn de orden n + 1)}\\ &< 2|A|^{-\ell+1}. && \text{(cuentita)}\end{aligned}$$

Por lo tanto,

$$\limsup_{N\to\infty} \frac{|a_1 \dots a_N|_u}{N} < 2|A|^{-\ell+1}.$$

El Teorema Piatetski-Shapiro nos dice que un número real x es normal para una base b si existe una constante C tal que para infinitas longitudes ℓ y para todo $w \in A^\ell$

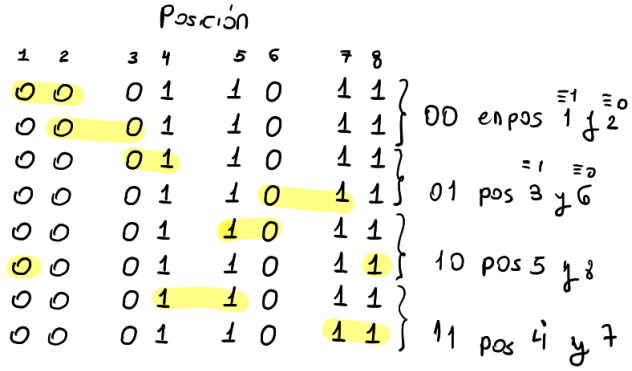
$$\limsup_{n\to\infty} \frac{|x[1 \dots n]|_w}{n} < C \cdot b^{-\ell}.$$

Interpretado para cadenas, podemos decir que b es el tamaño del alfabeto y que la expansión hasta n es un bloque de tamaño n. Por lo tanto, tomando $b = |A|$ y $C = 2|A|$ se cumple y concluimos que x (una palabra infinita de De Bruijn) es normal. □

1.6 Collares perfectos (*Perfect Necklaces*)

Consideremos todos los bloques de tamaño n, concatenados en orden lexicografico y vistos circularmente (como *collares*). Cada bloque de tamaño n ocurre exactamente n veces en posiciones diferentes modulo n.

Por ejemplo, para el alfabeto $\{0, 1\}$ y $n = 2$, los bloques concatenados en orden lexicográfico son 00 01 10 11 y



No toda permutación de los bloques de longitud n tiene esta propiedad, por ejemplo

- **00** 10 11 01: El 00 aparece solamente 1 vez.
- **000** 101 001 010 011 100 110 111: El 000 aparece solo una vez.

Def. 1.8 (Collar perfecto)**.** Un collar (cadena circular) sobre un alfabeto de b símbolos se dice **(n, k)-perfecto** si cada bloque de longitud n ocurre k veces, en posiciones diferentes modulo k, para cualquier convención de punto de partida.

Observaciones:

- Los collaras de De Bruijn son exactamente los collares $(n, 1)$-perfectos.
- Los collares (n, k)-perfectos tienen longitud kb^n

1.7 Definiciones equivalentes de normalidad

Un número real x es **simplemente normal en base b** si en la expansión de x en base b cada digito d ocurre con frecuencia $1/b$ en el límite,

$$\lim_{n\to\infty} \frac{|x[1\dots n]|_d}{n} = \frac{1}{b}$$

Def. 1.9 (Strong aligned normality)**.** *(Borel 1909)*
Un numero real x es **normal en base b** si cada real $x, bx, b^2x, \dots$ son simplemente normales a las bases $b^1, b^2, b^3, \dots$
bx es shiftear a la izquierda, correr la coma y tirar el 1er símbolo. b^2x es lo mismo pero tirando dos símbolos y quedándose con el resto.

Def. 1.10 (Aligned normality)**.** *(Pillai 1940).*
Un número real x es **normal en base b** si x es simplemente normal en bases $b^1, b^2, b^3, \dots$
No hacen falta los shifts, esto es como tomar alineada.

Def. 1.11 (Non-aligned normality)**.** *(Nivel and Zuckerman 1951)*
Un número real x es **normal en base b** si para cada bloque u,

$$\lim_{n\to\infty} \frac{|x[1\dots n]|_u}{n} = \frac{1}{|u|}$$

Teorema 1.8. Las tres definiciones de normalidad son equivalentes.

Proof. Vamos a probar que Strong aligned normality ⇒ Non-aligned normality ⇒ Aligned normality ⇒ Strong aligned normality

1. Strong aligned normality ⇒ Non-aligned normality

   Por la definición de normalidad alineada furte, sabemos que dado número x cada real $b^i x$ es simplemente normal a base b^i, es decir que para cada dígito w con $\ell = |w|$ y para cada i,

$$\lim_{n\to\infty} \frac{\|(b^i x)[1\dots \ell n]\|_w}{n} = b^{-\ell} \Leftrightarrow \lim_{n\to\infty} \frac{\|(b^i x)[1\dots n]\|_w}{n/\ell} = b^{-\ell}$$
$$\Leftrightarrow \lim_{n\to\infty} \frac{\|(b^i x)[1\dots n]\|_w}{n} = b^{-\ell}/\ell. \quad (1.1)$$

   Para cualquier $w \in A^\ell$, contar de manera no alineada es lo mismo que de forma alineada con todos los shifts ($n - i$ porque no me puedo pasar)

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

   Luego,

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

   Y por lo tanto es normal de forma no alineada.

2. Non-aligned normality ⇒ Aligned normality

   En clase

3. Aligned normality ⇒ Strong aligned normality

   En clase

□

1.7.1 Malas palabras

Intuitivamente, un segmento inicial se comporta bien o mal con respecto a lo que espero en el límite? Bien sería que se parezca, por ej. 1/2 de 1s y 1/2 de 0s. Mal sería que tenga más 0s que 1s.

Def (Malas palabras)**.** Sean A un alfabeto de b simbolos, k un entero positivo y ϵ un real entre 0 y 1. Definimos el conjunto de palabras de longitud k tales que una palabra w tiene un número de ocurrencias que difiere de lo esperado $\pm\epsilon k$,

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

Donde $|v|_w$ es lo *observado* y $\frac{k}{b^{|w|}}$ es lo *esperado*

Ejemplo. Por ejemplo,

- $A = \{0,1\}, k = 4, \epsilon = 1/4, w = 11.$

  Tenemos lo esperado $\frac{k}{b^{|w|}} = \frac{4}{2^2} = 1$, y la tolerancia $\epsilon k = 1$. Entonces $Bad(A,k,w,\epsilon) = \{1111\}$ es el conjunto de palabras con 3 ocurrencias de w (no alineadas).

- $A = \{0,1\}, k = 4, \epsilon = 1/4, w = 1.$

  Tenemos $\frac{k}{b^{|w|}} = \frac{4}{2^1} = 2, \epsilon k = 1$. Entonces $Bad(A,k,w,\epsilon) = \{1111, 0000\}$ es el conjunto de palabras con 4, 0 ocurrencias de w.

Los siguientes teoremas muestran que hay pocas malas palabras

Lema 1.1 (Hardy and Wright, Theorem 148)**.** Sean $b \geq 2$ y $k \geq 0$ enteros. Si $6/k \leq \epsilon \leq 1/b$ entonces para cada **símbolo** d en A,

$$|Bad(A,k,d,\epsilon)| < 4e^{-b\epsilon^2 k/6} b^k.$$

Obs para ejercicio 4: puedo ver dígitos en el alfabeto A^k para long k, y ahí estoy viendo ocurrencias alineadas.

Lema 1.2. Sea A un alfabeto de b símbolos, k, ℓ enteros positivos y ϵ un real tal que $6/\lfloor k/\ell \rfloor \leq \epsilon \leq 1/b^\ell$. Entonces,

$$\left| \bigcup_{w \in A^\ell} Bad(A,k,w,\epsilon) \right| < 2\ell \; b^{2\ell} \; e^{-b^\ell \epsilon^2 k/(6\ell)} b^k.$$

Teorema 1.9. Casi todas las secuencias son normales

Proof. En clase 3 diapo 18, usa Bad. □

Lema 1.3 (Pequeño truco de límites)**.** *Relación entre lim inf / lim sup y lim*

Sean $(x_{1,n})_{n\geq 0}, (x_{2,n})_{n\geq 0}, \ldots (x_{k,n})_{n\geq 0}$ una secuencia de números reales tales que $\sum_{i=1}^k x_{i,n} = 1$ y $c_1, c_2, \ldots c_k$ numeros reales tales que $\sum_{i=1}^k c_i = 1$. Luego,

1. $\forall i \;\; \liminf_{n\to\infty} x_{i,n} \geq c_i \implies \forall i \;\; \lim_{n\to\infty} x_{i,n} = c_i$
2. $\forall i \;\; \limsup_{n\to\infty} x_{i,n} \leq c_i \implies \forall i \;\; \lim_{n\to\infty} x_{i,n} = c_i$

Vamos a aplicar esto para $k = b^\ell$, $x_{n,i} = \frac{|x[1..n]|_{w_i}}{b^\ell}$ con $i = 1, 2, \dots b^\ell$.
Notar que

$$\sum_{i=1}^{b^\ell} \frac{|x[1..n]|_{w_i}}{b^\ell} = 1,$$

donde

- n = # palabras de ℓ caracteres (n posiciones que quiero mirar)
- w_i = palabras de longitud ℓ

en total tengo n ocurrencias. ℓ fijo y voy moviendo la posición. Es como un sliding window.

Supongo que es con x normal

Dem. "Puedo mirar las familias de lim inf y me alcanza para afirmar el límite". Es además un problema dual. Asumo 1. y veo que me da 2, el otro es análogo.

$$\begin{aligned}
\limsup_{n\to\infty} x_{i,n} &= \limsup_{n\to\infty} (1 - \sum_{j\neq i} x_{j,n}) && (\text{pues } \sum_{i=1}^{k} x_{i,n} = 1) \\
&= \underbrace{\limsup_{n\to\infty} 1}_{=1} + \limsup_{n\to\infty} (-\sum_{j\neq i} x_{j,n}) \\
&= 1 - \liminf_{n\to\infty} \sum_{j\neq i} x_{j,n} \\
&\leq 1 - \sum_{j\neq i} \liminf_{n\to\infty} x_{j,n} \\
&\leq 1 - \sum_{j\neq i} c_j && (\text{por hip. 2. } \liminf_{n\to\infty} x_{j,n} \geq c_i) \\
&= c_i && (\text{por hip.} \sum_{i=1}^{k} c_i = 1)
\end{aligned}$$

Y como

$$\limsup_{n\to\infty} x_{i,n} \leq c_i \leq \liminf_{n\to\infty} x_{i,n}$$

y lim sup $\geq$ lim inf, necesariamente

$$\limsup_{n\to\infty} x_{i,n} = c_i = \liminf_{n\to\infty} x_{i,n}$$

y entonces

$$\lim_{n\to\infty} x_{i,n} = c_i.$$

□

1.8 Algoritmos para absoluta normalidad

Un número x es **absolutamente normal** si es normal para todas las bases.

Def. 1.12 (Número computable)**.** Un número real x es computable si hay un programa de computadora que tiene de output su expansión fraccionaria en alguna base, dígito por dígito.

Teorema 1.10 (Turing 1936)**.** Sea x un número real en el intervalo unitario ($[0,1]$). Los siguientes son equivalentes:

1. x es computable
2. Existe uan función computable $f : \mathbb{N} \to \{0,1\}$ tal que $f(n)$ es el n-ésimo dígito en la expansión fraccionaria de x en base 2.
3. Hay una secuencia computable no decreciente de numeros racionales $(q_j)_{j\geq 1}$ tal que $\lim_{j\to\infty} q_j = x$ y para cada j, $|x - q_j| \leq 2^{-j}$

   Los primeros j símbolos están bien, porque sino la resta no daría $\leq 2^{-j}$. Es como decir "presición j", todo lo que venga despu[es de la pos j son 0s y 1s.
4. Hay una secuencia computable de intervalos $I_1, I_2, I_3, \ldots$ con bordes racionales anidados, cuyas longitudes van a 0 de forma tal que $x \in \bigcap_{j\geq 1} I_j$.

   Por ejemplo I_1 puede ser $[0,1]$, y luego I_2 tiene que estar anidado, por ejemplo la mitad. Y en el límite, en la intersección infinita está x.

Ejemplo. Ejemplos son 0, $\sqrt{2}$, π, e. Y un contraejemplo es el argumento diagonal de Cantor (tengo un ejemplo en mis notas).

1.8.1 Algoritmo de Turing

Teorema (Turing 1937?)**.** Hay un algoritmo que computa la expansión en base 2 de un número absolutamente normal en el intervalo unitario.

El algoritmo usa intervalos diádicos (potencias de 2). Para seleccionar $I_1, I_2, I_3, \ldots$ la estrategia es "seguir la medida". El número computado x entonces es la traza de las elecciones left / right. La defínción de normalidad absoluta que usa es

Def (Normalidad absoluta)**.** Un número real x es absolutamente normal si es simplemente normal en todas las bases enteras $b \geq 2$.

Def (Discrepancia simple)**.** Sean $x \in [0,1]$ un real y x_b su expansión en base b, definimos

$$\Delta_N(x_b) = \max_{d\in\{0,\ldots,b-1\}} \left| \frac{|x_b[1\ldots N]|_d}{N} - \frac{1}{b} \right|$$

Es como el desvío máximo de la frecuencia de un dígito de lo esperado en el segmento inicial de tamaño N. Discrepancia simple. $\Delta_N(x_b)$ se dice el dígito con la peor proporción

Luego x es simplemente normal en base b si

$$\lim_{N\to\infty} \Delta_N(x_b) = 0$$

Def (Pasos del algoritmo)**.** Usamos n como el paso del algoritmo y definimos las sig funciones,

> $\boldsymbol{N_n} = 2^{n_0+2n}$, el número de dígitos vistos en el paso n, donde $n_0 = 11$ (n_0 *solo está ahí para simplificar las cuentas*)
>
> $\boldsymbol{b_n} = \lfloor \log N_n \rfloor$ es la base más grande considerada en el paso n
>
> $\boldsymbol{\epsilon_n} = 1/b_n$ es la diferencia entre la frecuencia esperada de dígitos y la frecuencia actual en el paso n. El *nivel de tolerancia*, dependiente de la cantidad de bases que se están viendo.

$b_n \geq 2$ y es no decreciente y no acotada, N_n es no decreciente y no acotada, y ϵ_n es no creciente y va a 0.

Def (Conjuntos de candidatos)**.** Definimos los conjuntos de números reales

$$\begin{aligned} E_0 &= (0,1) \\ E_n &= \bigcap_{b\in\{2,\dots,b_n\}} \{x \in (0,1) : \Delta_{N_n}(x_b) < \epsilon_n\} \end{aligned}$$

Para cada n, el conjunto E_n consiste de todos los números reales cuyas expansiones en las bases $2, 3, \dots, b_n$ exhiben **buenas frecuencias** de digitos hasta ϵ_n, en los primeros N_n digitos (en el segmento inicial de tamaño N_n)

Lema. El conjunto $\bigcap_{n\geq 0} E_n$ tiene medida positiva y consiste de solo números absolutamente normales.

El algoritmo en sí es

> **Paso inicial, $\boldsymbol{n = 0}$.** $I_0 = (0,1)$, $E_0 = (0,1)$
>
> **Paso recursivo, $\boldsymbol{n > 1}$**
>
> En el paso anterior computamos I_{n-1}. Sea I_n^0 la mitad izquierda de I_{n-1} y I_n^1 la mitad derecha.
>
> - Si $\mu\left(I_n^0 \cap \bigcap_{j=0}^n E_j\right) > 1/N_n$ entonces $I_n = I_n^0$ y $y_n = 0$.
>
> No entiendo la condición
> - Sino, $I_n = I_n^1$ y $y_n = 1$.

$\mu A = |A|$ es la medida de Lebesgue de A. La probabilidad de que un número real positivo arbitrario aparezca en eso

El output es $y_1 y_2 y_3 \dots$.

Es un algoritmo *goloso*: Parte el intervalo obtenido en el paso anterior a la mitad, y se queda con la mitad que tiene más números que empiecen bien. Si elije el de la izq toma como dígito 0, y sino elige el de la der y toma 1.

Demostración de correctitud de algoritmo de Turing

Prop. Para cada n, E_n es una unión finita de intervalos abiertos con bordes racionales, y para $n \geq n_0$, $\mu E_n > 1 - \frac{1}{N_n^2}$.

Dem. Los valores de N_n y ϵ_n satisfacen las hipótesis del Lema 1.1 □

Prop.

1.8.2 BHS

Def (b-ario)**.** Decimos que un intervalo es b-ario (o b-ádico) de orden n si tiene la forma

$$\left(\frac{a}{b^n}, \frac{a+1}{b^n}\right)$$

para algun entero a tal que $0 \leq a < b^n$.

Si σ_b y τ_b son intervalos b-arios y $\tau_b \subseteq \sigma_b$ decimos que el *orden relativo* de τ_b con respecto a σ_b es el orden de τ_b menos el orden de σ_b.

Lema 1.4 (Lemma 3.1 BHS 2013)**.** Sean u y v bloques y ε un real positivo,

1. Si $\Delta(u) < \varepsilon$ y $\Delta(v) < \varepsilon$ entonces $\Delta(uv) < \varepsilon$.

   Si u es buena y v también, entonces su concatenación lo es. Esto me dice que si cuento dígito, puedo considerar la suma de las proporciones y va a andar todo bien.

2. Si $\Delta(u) < \varepsilon, v = a_1 \dots a_{|v|}$ y $|v|/|u| < \varepsilon$ entonces $\Delta(vu) < 2\varepsilon$ y para cada ℓ tal que $1 \leq \ell \leq |v|, \Delta(ua_1a_2 \dots a_l) < 2\varepsilon$.

   Concatenar una palabrita mucho mas chica que una buena al lado de una buena a lo sumo empeora el doble la discrepancia.

   $|v|/|u| < \varepsilon \Leftrightarrow |v| < \varepsilon|u|$, y como ε es un número chiquito entre 0 y 1, se interpreta como que u es muy chica en comparación a v.

   Para el algoritmo, nos dice que lo que ya viste no hace falta volverlo a mirar, podés mirar cosas nuevas. Si tengo un buen segmento inicial y lo que elijo para agregar es cortito, a lo sumo llevo mi discrepancia al doble.

Lema 1.5 (Lemma 3.4 BHS2013)**.** Para cualquier intervalo I y cualquier base b, hay un subintervalo b-ario J tal que $\mu J \geq \mu I/(2b)$.

Podemos encontrar J grande y b-ádico que se parezca mucho en medida

Proof. La idea es que queremos que sea lo más grande posible, porque si es muy chico va a tener símbolos con muchos dígitos asociados como para mirar. Idealmente quiero un dígito por cada base para el algoritmo.

Como J es b-ádico, sabemos que para algún a

$$J = \left(\frac{a}{b^n}, \frac{a+1}{b^n} \right),$$

y luego

$$\mu J = \frac{a+1}{b^n} - \frac{a}{b^n} = \frac{a - a + 1}{b^n} = \frac{1}{b^n} = b^{-n}$$

Como I no es exactamente b-ádico, hay dos casos,

1. J calza exactamente.
2. J va a caballo de dos b-ádicos.

más en las notas draft de la clase □

Def (t-sequences)**.** Una t-sequence $\vec{\sigma}$ es una secuencia de intervalos $(\sigma_2, \ldots, \sigma_t)$ tales que

- Para cada base $b = 2, \ldots, t$, σ_b es b-ario.
- Para cada base $b = 3, \ldots, t$, $\sigma_b \subset \sigma_{b-1}$ y $\mu\sigma_b \geq \mu\sigma_{b-1}/(2b)$.

Son intervalos anidados, con la medida lo más grande posible.

La definición implica que $\mu\sigma_t \geq (\mu\sigma_2)/(2^t t!)$ *(es un peor caso, no siempre ocurre y a veces calza justo)*

Para el ejercicio 6
Como hay que hacer para bases 2 y 3, vamos a necesitar

$$\sigma_2, \sigma_3, \sigma_{2^i}, \sigma_{3^i}$$

(si se hace demás, no está mal pero hay que ser *económicos*)

Def (Refinamiento)**.** Dos definiciones,

- Una t-sequence $\vec{\tau} = (\tau_2, \ldots, \tau_t)$ *refina* una t'-secuencia $\vec{\sigma} = (\sigma_2, \ldots, \sigma_{t'})$ si $t' \leq t$ y $\tau_b \subset \sigma_b$ para cada $b = 2, \ldots, t'$.
- Un refinamiento tiene *discrepancia* menor a ε si para cada $b = 2, \ldots, t'$ hay palabras u, v tales que $\sigma_b = I_u, \tau_b = I_{uv}$ y $\Delta(v) < \varepsilon$.

El τ que elegiste tiene una discrepancia $< \varepsilon$.

Lema 1.6. Sean $t \geq 2$ un entero, $t' = t \vee t+1$, $\varepsilon < 1/t$ un real positivo. Luego, toda t-sequence $\vec{\sigma} = (\sigma_2, \ldots, \sigma_t)$ admite un refinamiento $\vec{\tau} = (\tau_2, \ldots, tau_{t'})$ con discrepancia menor a ε.

El orden relativo de τ_2 puede ser cualquier entero k tal que

$$k \geq max \left\{ \frac{6}{\varepsilon}, \frac{24(\log_2 t)(\log(t!))}{\varepsilon^2} \right\}.$$

Dem. En clase diapo 5. □

Def (Pasos del algoritmo)**.** El algoritmo considera las siguientes funciones para un paso n dado,

> $\boldsymbol{t_n} = \max(2, \lfloor \sqrt[4]{\log n} \rfloor)$ es la máxima base a considerar en el paso n,
>
> $\boldsymbol{\varepsilon_n} = 1/t_n$ es la discrepancia máxima tolerada en el paso n, y
>
> $\boldsymbol{N_n} = \lfloor \log n \rfloor + n_{\text{start}}$ es el número de dígitos en base 2 agregados en el paso n,

Donde n_{start} es el minimo entero que valida la condición del Lema 1.6. Por lo tanto, requerimos que para todo $n > 0$,

$$\lfloor \log n \rfloor + n_{\text{start}} \geq max \left\{ \frac{6}{\varepsilon}, \frac{24(\log_2 t)(\log(t!))}{\varepsilon^2} \right\}.$$

El algoritmo construye $\vec{\sigma}_0, \vec{\sigma}_1, \vec{\sigma}_2, \ldots$ tal que $\vec{\sigma}_0 = (0, 1)$ y para cada $n \geq 1$, $\vec{\sigma}_n$ es una t_n-sequencia que refina $\vec{\sigma}_{n-1}$ con discrepancia ε_n y tal que el orden de $\sigma_{n,2}$ es N_n mas el orden de $\sigma_{n-1}, 2$.

Tiene de output $x = y_1 y_2 y_3 \ldots$ que son los símbolos en la expansión en base 2 de un número absolutamente normal.

> **Paso inicial,$\boldsymbol{n = 1}$.** $\vec{\sigma}_1 = (\sigma_2)$, con $sigma_2 = (0, 1)$.
>
> **Paso recursivo, $\boldsymbol{n > 1}$**
>
> En el paso anterior computamos $\vec{\sigma}_{n-1} = (\sigma_2, \ldots, \sigma_{t_n - 1})$.
>
> Tomamos $\sigma_n = (\tau_2, \ldots, \tau_{t_n})$ la t_n-sequence más a la izquierda que refina $\vec{\sigma}_{n-1}$ con discrepancia menor que ε_n tal que si $\sigma_2 = I_u$ entonces $\tau_2 = I_{uv}$ con $|v| = N_n$.
>
> Ponemos $y_{M_n+1} \ldots y_{M_n+N_n} = v$ donde $M_n = \sum_{j=1}^{n} N_n$.

Proof. En clase diapo 13. □