

# Información y Azar

Primer Cuatrimestre 2011  
Ejercicios de Práctica

**Notación.** Usamos  $\mathcal{A}$  para denotar el alfabeto.  $\mathcal{A}^*$  es el conjunto de palabras finitas sobre  $\mathcal{A}$  y  $\mathcal{A}^\omega$  el conjunto de palabras infinitas sobre  $\mathcal{A}$ . Para  $s \in \mathcal{A}^*$ ,  $|s|$  es la longitud de  $s$ .  $\lambda$  es la palabra vacía,  $|\lambda| = 0$ . Si  $w \in \mathcal{A}^*$  numeramos las posiciones de  $w$  desde 1 a  $|w|$ . Un segmento inicial  $w[1..i]$ , con  $1 \leq i \leq |w|$  es un prefijo de  $w$ , un segmento final  $w[i..|w|]$  con  $1 \leq i \leq |w|$  es un sufijo de  $w$ , y un segmento intermedio  $w[i..j]$ , con  $1 \leq i \leq j \leq |w|$  es una subpalabra de  $w$ .

Escribimos  $\leq_{\text{pref}}$  para denotar el orden de prefijos en  $\mathcal{A}^*$  (este es un orden parcial), y  $\leq_{\text{lex}}$  para denotar el orden de longitud-lexicográfico en  $\mathcal{A}^*$  (es un orden total). Asumimos la biyección  $\text{string} : \mathbb{N}_0 \rightarrow \mathcal{A}^*$ , que identifica cada palabra de  $\mathcal{A}^*$  con su posición en el orden lexicográfico; notar que  $\text{string}^{-1}(\lambda) = 0$  y para  $|w| \geq 1$ ,  $\text{string}^{-1}(w) = \sum_{i=1}^{|w|} (w(i) + 1)b^{i-1}$ , donde  $b$  es la cardinalidad del alfabeto  $\mathcal{A}$ .

Indistintamente nos referimos a *función parcialmente computable*, *función computable*, *función recursiva parcial*, *programa*, *computadora*, *máquina de Turing*, o simplemente *máquina*.

Dada una función  $f : \mathcal{A}^* \rightarrow \mathcal{A}^*$  parcialmente computable la función  $f_t : \mathcal{A}^* \times \mathbb{N} \rightarrow \mathcal{A}^*$  que da el resultado de computar  $t$  pasos de la función  $f$ . Notemos que si  $f(s)$  converge en  $t$  pasos, entonces para todo  $u \geq t$ ,  $f_u(s) = f_t(s) = f(s)$ .

Recordemos que un conjunto  $A \subseteq \mathcal{A}^*$  (o  $A \subseteq \mathbb{N}$ ) es *computablemente enumerable*, que abreviamos *c.e.*, si existe una función  $f$  parcialmente computable tal que  $A = \text{dominio}(f)$ . Equivalentemente,  $A$  es computablemente enumerable si existe una función total  $g : \mathbb{N} \rightarrow \mathcal{A}^*$  tal que  $A = \text{imagen}(g)$ ; y una enumeración computable de  $A$  está dada por  $g(1), g(2), \dots$ . Si  $A$  es computablemente enumerable y su complemento  $\mathcal{A}^* \setminus A$  también, entonces  $A$  es computable. Los conjuntos computables pueden ser enumerados en un orden computable.

**Definición** (computadora universal). Sea  $(T_i)_{i \in \mathbb{N}}$  una enumeración computable de todas las computadoras  $T_i : \mathcal{A}^* \rightarrow \mathcal{A}^*$ . Una computadora  $U : \mathcal{A}^* \rightarrow \mathcal{A}^*$  es universal si hay una función computable de pares (función computable inyectiva)  $\langle \cdot, \cdot \rangle : \mathcal{A}^* \times \mathcal{A}^* \rightarrow \text{words}$ , tal que  $U(\langle i, y \rangle) = T_i(y)$ . Notar que, dado que  $U$  es computadora, existe un índice  $j$  tal que  $U = T_j$ .

## 1. La función de complejidad K

**Definición** (complejidad K). Sea  $T$  una computadora. La función  $K_T : \mathcal{A}^* \rightarrow \mathbb{N}$  es tal que para toda  $s \in \mathcal{A}^*$ ,  $K_T(s) = \min\{|p| : T(p) = s\}$  si  $s \in \text{imagen}(T)$ ,  $K_T(s) = \infty$  en caso contrario.

En la literatura la función  $K$  tiene el nombre “plain complexity” o “Kolmogorov complexity” y muchas veces se la denota con el símbolo  $C$ . Los libros [2, 3] la llaman  $C$ .

**Definición** (computadora óptima). Una computadora  $V : \mathcal{A}^* \rightarrow \mathcal{A}^*$  es óptima si para toda máquina  $T$  existe  $c_T$  tal que para toda  $s \in \mathcal{A}^*$ ,  $\min\{|p| : V(p) = s\} \leq \min\{|p| : T(p) = s\} + c_T$

**Ejercicio 1.1** (trivial). Sea  $V$  una computadora. Demostrar que son equivalentes:

- i)  $V$  una computadora óptima.
- ii) Para toda computadora  $T$ , existe  $c_T$  tal que para toda  $s \in \mathcal{A}^*$ ,  $K_V(s) \leq K_T(s) + c_T$ .

**Notación.** Fijamos  $(T_i)_{i \in \mathbb{N}}$  una enumeración computable de todas las computadoras, y fijamos la computadora universal y óptima  $V : \mathcal{A}^* \rightarrow \mathcal{A}^*$  definida por  $V(0^i 1 p) = T_i(p)$ . En adelante escribimos  $K$  en vez de  $K_V$ .

**Ejercicio 1.2.** Demostrar que para toda función computable biyectiva  $g : \mathcal{A}^* \rightarrow \mathcal{A}^*$ , existe una constante  $c_g$  tal que para toda  $s, t \in \mathcal{A}^*$ ,  $K(s) \leq K(g(s)) + c_g$ .

**Definición** (descripción mínima).  $s^* = \min_{\leq_{\text{lex}}} \{p : V(p) = s\}$ .

**Ejercicio 1.3.** Demostrar que existe una constante  $c$  tal que  $K(s^*) \geq |s^*| - c$ .

**Ejercicio 1.4.** *demostrar que hay una computadora óptima tal que para cada palabra  $s$  y para cada longitud  $n$ , hay a lo sumo una descripción de longitud  $n$  que computa  $s$ .*

**Ejercicio 1.5.** a) *Definir una computadora universal que no es óptima.*  
 b) *Definir una computadora óptima que no es universal.*

**Definición.** *La función de complejidad condicional  $K_T : \mathcal{A}^* \times \mathcal{A}^* \rightarrow \mathbb{N}$  es tal que para toda  $s, t \in \mathcal{A}^*$ ,  $K_T(s/t) = \min\{|p| : T(p, t) = s\}$  si  $s \in \text{imagen}(T)$ ,  $K_T(s/t) = \infty$  en caso contrario. Esta noción mide la complejidad de  $s$  dándole “gratis” la palabra  $t$ .*

**Ejercicio 1.6.** a) *demostrar que existe  $c$  tal que  $K(s/t) \leq K(s) + c$ .*  
 b) *demostrar que existe  $c$  tal que  $K(st/K(s)) \leq K(s) + K(t) + c$ .*

**Ejercicio 1.7.** *demostrar que para infinitos  $n \in \mathbb{N}$ , tales que  $n$  es par pero no es múltiplo de 4, la cantidad de números primos menores que  $n$  es al menos  $\frac{\log_2 n}{\log_2 \log_2 n}$ .*  
*Ayuda: Adaptar la demostración de Chaitin que aparece en el libro “Kolmogorov complexity and its applications”, página 6.*

## 2. Conjuntos Libres de Prefijos

**Definición.** *Un conjunto  $A \subseteq \mathcal{A}^*$  es libre de prefijos si y sólo si para toda  $v, w \in \mathcal{A}^*$ , si  $w \in A$  y  $v <_{\text{pref}} w$  entonces  $v \notin A$ .*

**Ejercicio 2.1.** *demostrar que si  $L_1, L_2 \subseteq \mathcal{A}^*$  son libres de prefijos entonces  $L = \{uv \mid u \in L_1 \wedge v \in L_2\}$  es libre de prefijos.*

**Ejercicio 2.2.** *Un conjunto  $A \subseteq \mathcal{A}^*$  es clausurado por sufijos sii si  $w \in A$  entonces para toda  $x \in \mathcal{A}^*$ ,  $wx \in A$ . demostrar que para todo  $A \subseteq \mathcal{A}^*$  clausurado por sufijos existen  $B \subseteq A$  libre de prefijos tal que  $BA^* = A$ , y viceversa.*

**Definición.** *Llamemos  $b = |\mathcal{A}^*|$ , y asumimos  $b \geq 2$ . Sea  $L \subseteq \mathcal{A}^*$ .  $\text{weight}(L) = \sum_{w \in L} b^{-|w|}$ .*

**Ejercicio 2.3.** *Determinar  $V$  o  $F$  y justificar.*

1.  *$\text{weight}(L) \leq 1$  si y solo si  $L$  es libre de prefijos.*
2. *Si  $L$  es libre de prefijos entonces el complemento de  $L$ ,  $\mathcal{A}^* \setminus L$ , es libre de prefijos.*
3. *Si  $\text{weight}(L) \leq 1$  entonces  $L$  es c.e.*

**Ejercicio 2.4.** *demostrar que para todo conjunto  $L \subseteq \mathcal{A}^*$  existe siempre un conjunto  $L' \subseteq \mathcal{A}^*$  tal que existe una relación uno a uno entre  $L$  y  $L'$ , y  $L'$  es libre de prefijos. ¿Siempre  $\text{weight}(L') \leq \text{weight}(L)$ ?*

**Ejercicio 2.5.** *Llamemos  $b$  a la cantidad de símbolos de alfabeto  $A$ . Sea  $A \subseteq \mathcal{A}^*$  libre de prefijos. y para cada  $m$  sea  $a_m = |A \cap \mathcal{A}^{*m}|$  (el subconjunto de palabras de  $A$  de longitud  $m$ ). demostrar que  $\sum_{m \in \mathbb{N}} b^{-m + \log_b a_m} \leq 1$ .*

**Ejercicio 2.6.** *Dado  $A \subseteq \mathcal{A}^*$  c.e. y libre de prefijos demostrar que existe  $B \subseteq \bigcup_{n=0}^{\infty} \mathcal{A}^{2^n}$  computable tal que  $\text{weight}(A) = \text{weight}(B)$ .*

**Ejercicio 2.7.** *Sea  $(T_i)_{i \in \mathbb{N}}$  una enumeración computable de las máquinas de Turing. Sea  $h : \mathbb{N} \rightarrow \{a, b\}$ , tal que  $h(i) = a$  si  $T_i(\lambda) \downarrow$ , y  $h(i) = b$ , en caso contrario; y sea  $\bar{h} : \mathbb{N} \rightarrow \{a, b\}$ , la opuesta de  $h$ , es decir, tal que  $\bar{h}(i) = a$  sii  $h(i) = b$ . Sea  $L = \{h(1) \dots h(i-1)\bar{h}(i) \mid i > 1\}$ .*

- *¿Es  $L$  c.e.?*
- *¿Es  $L$  libre de prefijos?*
- *Calcular  $\text{weight}(L)$ .*
- *Sea  $L' = \{w \in L : \exists v w = vb\}$ . ¿Es  $\text{weight}(L')$  aproximable desde abajo algorítmicamente?*

### 3. La función de complejidad H

Las computadoras libres de prefijos, en inglés *prefix-free machines* o *self-delimiting machine*), son un subconjunto propio de las funciones parcialmente computables.

**Definición.** Una función  $M : \mathcal{A}^* \rightarrow \mathcal{A}^*$  es una computadora libre de prefijos si  $M$  es una función parcialmente computable y el dominio de  $M$  es un conjunto libre de prefijos.

**Notación.** Fijamos  $(M_i)_{i \in \mathbb{N}}$  una enumeración computable de todas las computadoras libres de prefijos. Fijamos  $U : \mathcal{A}^* \rightarrow \mathcal{A}^*$ ,  $U(0^i 1 p) = M_i(p)$ , que es universal y óptima.

**Ejercicio 3.1.** Definir una computadora libre de prefijos universal y óptima para la cual haya al menos dos descripciones mínimas para cada palabra  $s \in \mathcal{A}^*$ .

**Definición.** Sea  $M : \mathcal{A}^* \rightarrow \mathcal{A}^*$  una computadora libre de prefijos.  $\Omega_U = \text{weight}(\text{dominio } U)$ .

**Ejercicio 3.2.** Definir una computadora libre de prefijos óptima  $W : \{0, 1\}^* \rightarrow \{0, 1\}^*$

1. para la cual  $\Omega_W > 3/4$ .
2. los primeros tres dígitos de la expansión decimal de  $(1 - \Omega_W)$  en binario sean 101.

**Definición** (complejidad H). Sea  $M$  una computadora libre de prefijos. La función  $H_M : \mathcal{A}^* \rightarrow \mathbb{N}$  es tal que para toda  $s \in \mathcal{A}^*$ ,  $H_M(s) = \min\{|p| \mid M(p) = s\}$  si  $s \in \text{imagen}(M)$ ,  $H_M(s) = \infty$  en caso contrario. Escribimos  $H$  para denotar  $H_U$ . Para  $s \in \mathcal{A}^*$ , escribimos  $s^* = \min_{\leq_{\text{lex}}} \{p : U(p) = s\}$ .

Esta es la notación que da Chaitin [1]. En la literatura [3, 2] aparece como la función  $K$ .

**Ejercicio 3.3.** Demostrar que existe una constante  $c$  tal que para todo  $s$ ,  $K(s) \leq H(s) + c$ .

**Ejercicio 3.4.** Demostrar que existe una constante  $c$  tal que

- a) Para todo  $d$ , para todo  $n \in \mathbb{N}$ ,  $\{s \in \mathcal{A}^n \mid H(s) \leq n + H(n) - d\} < 2^c 2^{n-d}$ .
- b) Para todo  $d$ , para todo  $n \in \mathbb{N}$ ,  $\{s \in \mathcal{A}^n \mid H(s) \leq H(n) + d\} < 2^c 2^d$ .

**Ejercicio 3.5.** Demostrar:

1. Para  $g : \mathcal{A}^* \rightarrow \mathcal{A}^*$  computable biyectiva existe  $c$  tal que para todo  $s$ ,  $|H(s) - H(g(s))| < c$ .
2. Sean  $f, g : \mathbb{N} \rightarrow \mathcal{A}^*$  funciones computables inyectivas. Probar que existe una constante  $c$  tal que para todo  $n \in \mathbb{N}$ ,  $|H(f(n)) - H(g(n))| \leq c$ .
3. Dadas  $U_1$  y  $U_2$  máquinas libres de prefijos óptimas, demostrar que existe una constante  $c$  tal que para todo  $s \in \mathcal{A}^*$ ,  $|H_{U_1}(s) - H_{U_2}(s)| \leq c$ .

**Notación.**  $H(s, t) = H(\langle s, t \rangle)$  para una función de pares  $\langle \cdot, \cdot \rangle$  computable fijada de antemano.

**Definición.** Sea  $M^2 : \mathcal{A}^* \times \mathcal{A}^* \rightarrow \mathcal{A}^*$  parcialmente computable tal que para todo  $t \in \mathcal{A}^*$ ,  $\{p \in \mathcal{A}^* \mid M^2(p, t) \downarrow\}$  es libre de prefijos. Definimos  $H_{M^2}(s/t) = \min\{|p| \mid M^2(p, t) = s\}$  si  $s \in \text{imagen}(M^2(\cdot, t))$ ,  $H_{M^2}(s/t) = \infty$  en caso contrario. Escribimos  $H$  para denotar  $H_{U^2}$ .

Observar que el conjunto  $\{s : H(s/t^*) \leq n\}$  es c.e. dados  $n$  y  $t^*$ .

**Ejercicio 3.6.** Demostrar:

1.  $H(s, t) = H(t, s) + O(1)$ .
2.  $H(s, |s|) = H(s) + O(1)$ .
3.  $H(s, H(s)) = H(s) + O(1)$ .
4.  $H(H(s)/s^*) = O(1)$ .

**Definición.** Llamamos  $b = |\mathcal{A}|$ . Sea  $M : \mathcal{A}^* \rightarrow \mathcal{A}^*$  computadora libre de prefijos.

$$P_M(s) = \sum_{\{p \in \mathcal{A}^* \mid M(p)=s\}} b^{-|p|}$$

Observar que  $P_M(s) = \text{weight}(M^{-1}(s))$ .

**Ejercicio 3.7.** Definir una computadora  $M$  libre de prefijos óptima tal que para toda  $s \in \text{words}$ ,  $P_M(s)$  es un número racional.

**Ejercicio 3.8.** Sea  $C : \{0, 1\}^* \rightarrow \{0, 1\}^*$  una computadora libre de prefijos, sea  $t \in \mathcal{A}^*$  fijo y sea  $T = \{“P_C(s/t^*) > 2^{-n}” : s \in \mathcal{A}^*\}$ ; es decir,  $T$  es el conjunto de las proposiciones verdaderas que afirman que la probabilidad de que la computadora  $C$  con entrada  $t^*$  arroje una cadena  $s$  es mayor que  $2^{-n}$ , para distintos valores de  $n$ . Sea  $\text{Req} = \{(s, n+1) \mid “P_C(s/t^*) > 2^{-n} \in T”\}$ . Demostrar que  $\sum_{(s,i) \in \text{Req}} 2^{-i} \leq 1$ .

**Ejercicio 3.9.** Definir una computadora  $W$  libre de prefijos óptima que tenga exactamente 4 programas mínimos para cada cadena  $s \in \mathcal{A}^*$  y cumpla  $H_W(s) = -\log_2 P_U(s) + O(1)$ .

## Referencias

- [1] Gregory Chaitin, A theory of program size formally identical to information theory, *Journal of the ACM* 22, 329-340, 1975.
- [2] Rod Downey, Denis Hirschfeldt. *Algorithmic Randomness and Complexity*, Springer, 2010.
- [3] Andre Nies. *Computability and Randomness*, Oxford University Press, 2009.