La aleatoriedad no es recursivamente invariante

Sean $A, B \subseteq \Sigma^*$.

Definición 1. *A es* 1-reducible *a B ($A \leq_1 B$) si existe una función $f : \Sigma^* \to \Sigma^*$ recursiva, total y 1-1 tal que para toda $w \in \Sigma^*$, $w \in A$ sii $f(w) \in B$.*

Definición 2. *A y B son* recurivamente isomorfos *si $A \leq_1 B$ y $B \leq_1 A$.*

Por ejemplo, el conjunto de las cadenas pares (que terminan en 0) y el de las impares (que terminan en 1) son recursivamente isomorfos.

Proposición 1. *Existen conjuntos $A, B \subseteq \Sigma^*$ recursivamente isomorfos tales que $\mu(A\Sigma^\omega)$ es aleatoria y $\mu(B\Sigma^\omega)$ no lo es.*

Demostración. Consideremos una máquina universal de Chaitin V definida así:

$$V(p) = \begin{cases} U(q) & \text{si } p = 0q \\ \uparrow & \text{en otro caso} \end{cases}$$

Sea $A = Dom(V)$ y $B = \{0^i 1 / U(string(i)) \downarrow\}$. Notemos que $0\,string(i) \in A$ si y sólo si $0^i 1 \in B$. Para ver que $A \leq_1 B$ definimos la función $f : \Sigma^* \to \Sigma^*$:

$$f(p) = \begin{cases} \lambda & \text{si } p = \lambda \\ 0^i 1 & \text{si } p = 0\,string(i) \\ 1p & \text{en otro caso } (p = 1q) \end{cases}$$

f es total, recursiva y 1-1. Veamos que $p \in A$ sii $f(p) \in B$, para toda $p \in \Sigma^*$:

$$\begin{aligned} p \in A \Leftrightarrow & V(p) \downarrow \Leftrightarrow p = 0q \wedge U(q) \downarrow \Leftrightarrow p = 0q \wedge q = string(i) \wedge U(string(i)) \downarrow \\ \Leftrightarrow & f(p) = 0^i 1 \wedge U(string(i)) \downarrow \Leftrightarrow f(p) \in B \end{aligned}$$

Para ver que $B \leq_1 A$, definimos la función $g : \Sigma^* \to \Sigma^*$:

$$g(p) = \begin{cases} \lambda & \text{si } p = \lambda \\ 0\,string(i) & \text{si } p = 0^i 1 \\ 1p & \text{en otro caso } (p = 0q \text{ y } p \neq 0^i 1, \text{ ó } p = 1q \text{ y } q \neq \lambda) \end{cases}$$

g es recursiva, total y 1-1. Falta ver que $p \in B$ sii $g(p) \in A$ para $p \in \Sigma^*$:

$$\begin{aligned} p \in B \Leftrightarrow & p = 0^i 1 \wedge U(string(i)) \downarrow \Leftrightarrow g(p) = 0\,string(i) \wedge U(string(i)) \downarrow \\ \Leftrightarrow & V(g(p)) \downarrow \Leftrightarrow g(p) \in A \end{aligned}$$

Por lo tanto, A y B son recursivamente isomorfos. Ahora veamos que $\mu(A\Sigma^\omega)$ es aleatoria y $\mu(B\Sigma^\omega)$ no. $\mu(A\Sigma^\omega) = \frac{1}{2}\Omega$ y entonces es aleatoria. ¿Qué ocurre con $\mu(B\Sigma^\omega)$? El $(i+1)$-ésimo bit de $\mu(B\Sigma^\omega)$ es 1 sii $U(string(i)) \downarrow$. Luego, si conocemos la cantidad m de todos los programas que se detienen en U entre las primeras n cadenas, podemos determinar el prefijo de longitud $n+1$ de $\mu(B\Sigma^\omega)$. La siguiente máquina de Chaitin $C(n^* m^*)$ computa $\mu(B\Sigma^\omega) \upharpoonright (n+1)$:

1. `Computar` $n = U(n^*)$ `y` $m = U(m^*)$
2. `Inicializar` $\mu(B\Sigma^\omega) \upharpoonright (n+1) = 0^{n+1}$
3. `Hacer dovetailing entre las` n `primeras cadenas, hasta hallar` m `de ellas que se detengan en` U
4. `Para cada` i `tal que` $U(string(i)) \downarrow$`,` $0 \leq i < n$`, poner el` $(i+1)$`-ésimo bit de` $\mu(B\Sigma^\omega) \upharpoonright (n+1)$ `en 1`
5. `Retornar` $\mu(B\Sigma^\omega) \upharpoonright (n+1)$

Por lo tanto, $H_C(\mu(B\Sigma^\omega) \upharpoonright (n+1)) \leq |n^*| + |m^*| = H(n) + H(m) \leq 2\log_2(n) + 2\log_2(m) + O(1)$. Como $m \leq n$, $log_2(m) \leq \log_2(n)$, entonces $H_C(\mu(B\Sigma^\omega) \upharpoonright (n+1)) \leq 4\log_2(n) + O(1)$. Aplicando el Teorema de Invarianza, $H(\mu(B\Sigma^\omega) \upharpoonright (n+1)) \leq H_C(\mu(B\Sigma^\omega) \upharpoonright (n+1)) + sim_C \leq 4\log_2(n) + k$ para alguna constante positiva k. Y, dada k, es posible hallar, para cada $c \geq 0$, un valor de n suficientemente grande de manera que $4\log_2(n) + k \leq (n+1) - c$. Para este n valdrá que $H(\mu(B\Sigma^\omega) \upharpoonright (n+1)) \leq (n+1) - c$. Luego resulta $\forall c\, \exists n\, H(\mu(B\Sigma^\omega) \upharpoonright n) \leq n - c$, es decir, $\mu(B\Sigma^\omega)$ no es aleatoria. $\square$