Teoría de la Complejidad Computacional

David Orellana Martín

Noviembre de 2023

Estos ejercicios corresponden al libro [1], en las páginas 423–424.

1. ¿Cuáles de las siguientes son medidas de complejidad?
   (a) $f_e(x) = 0$ para todo i, x; es decir, toda computación es «libre de recursos».
   (b) $f_e(x) = \begin{cases} M_e(x) & \text{si} \quad i \notin A \\ 0 & \text{si} \quad i \in A \end{cases}$ siendo A un conjunto finito tal que φ_e es total para todo $i \in A$; es decir, los programas cuyos códigos pertenecen a A son «libres de recursos».
   (c) $f_e(x) = 2^{\varphi_e(x)}$
   (d) $f_e(x) = \begin{cases} M_e(x) & \text{si} \quad i \text{ es par} \\ \text{número de pasos en la computación } M_e(x) & \text{si} \quad i \text{ es impar} \end{cases}$

6. Sea C una medida de complejidad arbitraria. Demostrar que existe una función recursiva h tal que

$$\varphi_e(x) \leq h(x, f_e(x)) \quad \text{c.t.p.}$$

   [*Pista*: Use la medida de complejidad $M_e(x)$ y el teorema de la **recursividad relativa**.]

7. ¿Puede el resultado del anterior problema ser mejorado tal que h sea una función recursiva *unaria* tal que

$$\varphi_e(x) \leq h(f_e(x)) \quad \text{c.t.p.?}$$

   Pruebe que su respuesta es correcta.

   Ejercicios extra:

1. Defina una familia de funciones $\{f_e(x) \mid e \in \mathbb{N}\}$ que **no** verifique el primer axioma de Blum, pero **sí** verifique el segundo axioma de Blum.

2. Defina una familia de funciones $\{f_e(x) \mid e \in \mathbb{N}\}$ que **no** verifique ninguno de los axiomas de Blum.

3. Invéntese una medida de complejidad (aquí entra en juego la imaginación) y demuestre que, efectivamente, es una medida abstracta de complejidad computacional.

Referencias

[1] M. Davis, R. Sigal, and E.J. Weyuker. *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science.* Computer Science and Scientific Computing. Elsevier Science, 1994.