Lógica y Computabilidad

Primer Cuatrimestre – 2024

Práctica 10 – Teoría de la Computabilidad

Un subconjunto de $\mathbb{N}^n$ se dice *recursivo*, *computable* o *decidible* si su función característica es total computable. Por otra parte, los subconjuntos de $\mathbb{N}^n$ que son dominio de alguna función parcial computable se dicen *computablemente enumerables (c.e.)* o *recursivamente enumerables (r.e.)*, mientras que los complementos de estos conjuntos se conocen como *co-computablemente enumerables (co-c.e.)*.

Funciones No Computables

1. Probar que existe una función recursiva primitiva $g(u, v, w)$ tal que

$$\Phi^{(3)}(u, v, w, z) = \Phi_{g(u,v,w)}(z).$$

2. Probar que son computables las funciones parciales

$$f(x) = \begin{cases} \Phi(x, x) + 1 & \text{si } \Phi(x, x)\downarrow, \\ \uparrow & \text{en otro caso,} \end{cases} \qquad \text{y} \qquad h(x) = \begin{cases} 0 & \text{si } \Phi(x, x)\downarrow, \\ \uparrow & \text{en otro caso.} \end{cases}$$

3. Sea $f : \mathbb{N} \to \mathbb{N}$ una función total, computable y biyectiva. Probar que $\mathrm{HALT}(f(x), x)$ no es computable. Sugerencia: considerar la función auxiliar

$$h(x) = \begin{cases} 1 & \text{si } \Phi(x, f^{-1}(x))\uparrow, \\ \uparrow & \text{en otro caso.} \end{cases}$$

4. Decidir si $g(x) = \begin{cases} 1 & \text{si } x \in \mathrm{Dom}(f), \\ \uparrow & \text{si } x \notin \mathrm{Dom}(f), \end{cases}$ es parcial computable sabiendo que f lo es.

5. Probar que la función total $f : \mathbb{N}^2 \to \mathbb{N}$ dada por

$$f(x, y) = \begin{cases} 1 & \text{si } y \notin \mathrm{Im}(\Phi_x), \\ 0 & \text{en otro caso,} \end{cases}$$

no es computable.

6. Decimos que una función parcial computable f es *extensible* si existe una función g total computable tal que $f(x) = g(x)$ para todo $x \in \mathrm{Dom}(f)$. Probar que existe una función parcial computable que no es extensible.

7. Probar que existe una función parcial computable g de una variable para la cual

$$f(x, y) = \begin{cases} 1 & \text{si } g(x) = y \\ 0 & \text{si } g(x) \neq y \end{cases}$$

resulta no computable. ¿Qué podría decirse de f cuando g es total computable?

Conjuntos Computables y Computablemente Enumerables

8. Considerar la función parcial $f(x,y) = \begin{cases} 0 & \text{si } \Phi_x(x)\downarrow, \\ \uparrow & \text{en otro caso.} \end{cases}$

   a) Probar que f es parcialmente computable y que existe una función recursiva primitiva h de una variable tal que $f(x,y) = \Phi_{h(x)}(y)$.
   b) Probar que $\Phi_{h(x)}$ es una función constante si y sólo si $\Phi_x(x)$ está definida.
   c) Probar que el conjunto $\{x \in \mathbb{N} : \Phi_x \text{ es constante}\}$ no es recursivo.

9. Probar que el conjunto $\{x \in \mathbb{N} : \mathrm{Dom}(\Phi_x) = \emptyset\}$ no es recursivo.

10. Probar que los siguientes conjuntos no son recursivos:

    a) $\{(x,y) \in \mathbb{N}\times\mathbb{N} : y \in \mathrm{Im}(\Phi_x)\}$,
    b) $\{(x,y) \in \mathbb{N}\times\mathbb{N} : \Phi_x = \Phi_y\}$,
    c) $\{x \in \mathbb{N} : \mathrm{Im}(\Phi_x) \text{ es infinito}\}$,
    d) $\{x \in \mathbb{N} : \mathrm{Im}(\Phi_x) \text{ es finito}\}$.

11. Probar que todo conjunto r.e. infinito contiene un subconjunto recursivo infinito.

12. Analizar la validez de las siguientes afirmaciones:

    a) Si B es r.e., entonces B es recursivo o $\mathbb{N}\backslash B$ es recursivo.
    b) Si $(B_n)_{n\in\mathbb{N}}$ es una familia numerable de conjuntos r.e., entonces $\bigcup_{n=1}^{\infty} B_n$ es r.e.

13. Probar que si B es r.e. y f es una función parcial computable entonces $f^{-1}(B)$ es r.e.

14. Probar que las siguientes funciones no son computables:

    a) $f(x) = \begin{cases} 1 & \text{si } \Phi_x(x) = 2x, \\ 0 & \text{en otro caso.} \end{cases}$
    b) $f(x,y) = \begin{cases} 1 & \text{si } \mathrm{Dom}(\Phi_x) = \emptyset, \\ 0 & \text{en otro caso.} \end{cases}$
    c) $f(x,y) = \begin{cases} 1 & \text{si } \Phi_x = \Phi_y, \\ 0 & \text{en otro caso.} \end{cases}$
    d) $f(x,y) = \begin{cases} 1 & \text{si } \mathrm{Im}(\Phi_x) \text{ es infinita}, \\ 0 & \text{en otro caso.} \end{cases}$
    e) $f(x,y) = \begin{cases} 1 & \text{si } 1 \in \mathrm{Dom}(\Phi_x), \\ 0 & \text{en otro caso.} \end{cases}$

15. Decidir si los siguientes conjuntos son c.e.:

    a) $\{x \in \mathbb{N} : \Phi_x(0)\downarrow\}$, b) $\{x \in \mathbb{N} : \Phi_x(x)\downarrow\}$, c) $\{x \in \mathbb{N} : \mathrm{Dom}(\Phi_x) = \emptyset\}$.

16. Probar que B es c.e. e infinito si y sólo si existe una función total $f\colon \mathbb{N} \to \mathbb{N}$ inyectiva y recursiva tal que el rango f es B.

17. Probar que $B = \{x \in \mathbb{N} : 1 \in \mathrm{Dom}(\Phi_x)\}$ es c.e. pero no es recursivo.

18. Probar que son recursivos los conjuntos

    $A = \{x \in \mathbb{N} : \Phi_x(x) \text{ se puede computar en menos de } x \text{ pasos}\}$,

    $B = \{x \in \mathbb{N} : \text{el programa con índice } x \text{ tiene menos de } x \text{ líneas}\}$.

19. Decidir si $A = \{x \in \mathbb{N} : \Phi_x(x) = 0\}$ y/o $B = \{x \in \mathbb{N} : \Phi_x(x) < x\}$ son recursivos.

20. Demostrar que los siguientes conjuntos no son ni recursivos, ni c.e., ni co-c.e.:

    a) $\mathrm{TOT} = \{x \in \mathbb{N} : \Phi_x \text{ es total}\}$,
    b) $\mathrm{ID} = \{x \in \mathbb{N} : \Phi_x \text{ es la función identidad}\}$,
    c) $\mathrm{S} = \{x \in \mathbb{N} : \Phi_x \text{ es sobreyectiva}\}$.