Parcial de Computabilidad

Lógica y Computabilidad

16 febrero, 2008

Este examen se aprueba obteniendo al menos **60 puntos**. El parcial es a libro abierto y se puede suponer demostrado todo lo que se dio en clase, colocando referencias claras. En el caso de usar resultados de las guías de ejercicios, se deben incluir las demostraciones.

Ejercicio 1. (**20 p**) Dada una lista l, una *meseta* es una sublista de l cuyos elementos son todos iguales. Sea $f : \mathbb{N} \to \mathbb{N}$ una función que, dada una lista, devuelve su meseta de mayor longitud. Si hay más de una meseta que cumple con esta condición, f debe devolver cualquiera de ellas. Demostrar que f es una función primitiva recursiva.

Ejercicio 2. (**50 p**) Sea $B = \{\langle x, y\rangle \mid \text{ran}(\Phi_x) \subseteq \text{ran}(\Phi_y)\}$

a. (**25 p**) Decidir si B es r.e y demostrarlo.

b. (**25 p**) Decidir si $\overline{B}$ es r.e y demostrarlo.

Sugerencia: instanciar alguna de las coordenadas del par apropiadamente.

Ejercicio 3. (**30 p**) Dadas dos funciones parciales $f : \mathbb{N} \to \mathbb{N}$ y $g : \mathbb{N} \to \mathbb{N}$ decimos que f *domina* a g si existe un $x_0 \in \mathbb{N}$ tal que para todo $x \geq x_0$, cuando tanto $f(x)$ como $g(x)$ están definidas, entonces $f(x) \geq g(x)$. Sea la función $h : \mathbb{N} \to \mathbb{N}$:

$$h(x) \quad = \quad \begin{cases} \text{mín}_t \, \text{STP}(x, x, t) & \text{si } \Phi(x, x) \downarrow \\ \uparrow & \text{en otro caso} \end{cases}$$

a. (**5 p**) Demostrar que h es parcialmente computable.

b. (**25 p**) Demostrar que si $g : \mathbb{N} \to \mathbb{N}$ es una función total que domina a h, entonces g no es computable.

Ejercicio 1 La siguiente función es primitivo recursiva:

$$f(x) = min_{t \leq x}(esMesetaEn(t, x) \wedge (\forall_{t1 \leq x}(esMesetaEn(t1, x) \rightarrow |t1| \leq |t|)))$$

$esMesetaEn(t, x) = esMeseta(t) \wedge sublista(t, x)$
$esMeseta(t) = \forall_{1 \leq i \leq |t|}(t[i] = t[1])$
$sublista(t, x) = \exists_{1 \leq i \leq |x|} \forall_{1 \leq j \leq |t|} t[j] = x[j + i \dot{-} 1]$

Ejercicio 2.a. Supongamos que B es r.e. Sea e el número de un programa que se indefine siempre. Observar que $\text{ran}(\Phi_e) = \emptyset$. Entonces

$$\widetilde{B} = \{x \mid \langle x, e\rangle \in B\} = \{x \mid \text{ran}(\Phi_x) = \emptyset\} = \{x \mid \Phi_x \text{ se indefine siempre}\}$$

es también r.e. Luego la función $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ es parcialmente computable:

$$f(x, y) = \begin{cases} 0 & \text{si } x \in \widetilde{B}; \\ \uparrow & \text{si no.} \end{cases}$$

Por el Teorema de la Recursión, existe c tal que $\Phi_c(y) = f(c, y)$ para todo y. Instanciando $x = c$ en la definición de f tenemos

$$\Phi_c(y) = \begin{cases} 0 & \text{si } \Phi_c \text{ se indefine siempre;} \\ \uparrow & \text{si no.} \end{cases}$$

que es una contradicción. El absurdo vino de suponer que B era r.e.

Ejercicio 2.b. Supongamos que $\overline{B}$ es r.e. La función de codificación de pares $\langle \cdot, \cdot \rangle$ es sobreyectiva, entonces

$$\overline{B} = \{\langle x, y\rangle \mid \text{ran}(\Phi_x) \not\subseteq \text{ran}(\Phi_y)\}.$$

Sea e el número de un programa que computa la identidad. Observar que $\text{ran}(\Phi_e) = \mathbb{N}$. Entonces

$$\widetilde{B} = \{x \mid \langle e, x\rangle \in \overline{B}\} = \{x \mid \mathbb{N} \not\subseteq \text{ran}(\Phi_x)\} = \{x \mid \text{ran}(\Phi_x) \neq \mathbb{N}\}$$

es también r.e. Luego la función $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ es parcialmente computable:

$$f(x, y) = \begin{cases} y & \text{si } x \in \widetilde{B}; \\ \uparrow & \text{si no.} \end{cases}$$

Por el Teorema de la Recursión, existe c tal que $\Phi_c(y) = f(c, y)$ para todo y. Instanciando $x = c$ en la definición de f tenemos

$$\Phi_c(y) = \begin{cases} y & \text{si } \text{ran}(\Phi_c) \neq \mathbb{N}; \\ \uparrow & \text{si no.} \end{cases}$$

que es una contradicción. El absurdo vino de suponer que $\overline{B}$ era r.e.

Ejercicio 3.a. El siguiente programa computa $h(x)$:

```
[A] if STP(x,x,y) goto E
    y <- y + 1
    GOTO A
```

Ejercicio 3.b. Sea x_0 tal que para todo $x \geq x_0$, $g(x) \geq f(x)$. Para todo $x \geq x_0$ tenemos $\text{HALT}(x, x)$ sii $\text{STP}(x, x, g(x))$. Veamos por qué:

- Si $\text{HALT}(x, x)$ entonces existe t tal que $\text{STP}(x, x, t)$. Luego $h(x) \downarrow$ y $\text{STP}(x, x, z)$ para todo $z \geq h(x)$. En particular, para $z = g(x)$.
- Si no $\text{HALT}(x, x)$ entonces $\text{STP}(x, x, t)$ es falso para todo t. En particular $\text{STP}(x, x, g(x))$ es falso.

Si g fuese computable, $\text{STP}(x, x, g(x))$ también sería computable. Luego

$$A = \{x \mid x \geq x_0 \wedge \text{HALT}(x, x)\}$$

sería un conjunto computable. Además,

$$B = \{x \mid x < x_0 \wedge \text{HALT}(x, x)\}$$

es finito y por lo tanto computable. Luego $A \cup B = \{x \mid \text{HALT}(x, x)\}$ es computable y esto es absurdo. La contradicción vino de suponer que g era computable.