Lógica y Computabilidad

2do cuatrimestre 2020 - **A DISTANCIA**

Departamento de Computación - FCEyN - UBA

Computabilidad - clase 6

Teorema de la forma normal, teorema del parámetro, teorema de la recursión y aplicaciones, teorema del punto fijo

Teorema de la Forma Normal

Teorema

Sea $f : \mathbb{N}^n \to \mathbb{N}$ una función parcial computable. Entonces existe un predicado p.r. $R : \mathbb{N}^{n+1} \to \mathbb{N}$ tal que

$$f(x_1, \ldots, x_n) = l\left(\min_z R(x_1, \ldots, x_n, z)\right)$$

Demostración.

Sea e el número de algún programa para $f(x_1, \ldots, x_n)$.
Recordar que la configuración instantánea se representa como

$$\langle \text{número de instrucción}, \text{lista representando el estado} \rangle$$

El siguiente predicado $R(x_1, \ldots, x_n, z)$ es el buscado:

$$\begin{array}{c} \mathrm{STP}^{(n)}(x_1, \ldots, x_n, e, r(z)) \ \wedge \\ l(z) = \underbrace{r\underbrace{\left(\mathrm{SNAP}^{(n)}(x_1, \ldots, x_n, e, r(z))\right)}_{\text{estado final de } e \text{ con entrada } x_1, \ldots, x_n} [1]}_{\text{valor de la variable } Y \text{ en ese estado final}} \end{array}$$

$\square$

Otra caracterización de funciones computables

Teorema

Una función es parcial computable si se puede obtener a partir de las funciones iniciales por un número finito de aplicaciones de

- *composición,*
- *recursión primitiva y*
- *minimización*

Teorema

Una función es computable si se puede obtener a partir de las funciones iniciales por un número finito de aplicaciones de

- *composición,*
- *recursión primitiva y*
- *minimización propia*
  (del tipo $\text{mín}_t\, q(x_1, \ldots, x_n, t)$ *donde siempre existe al menos un* t *tal que* $q(x_1, \ldots, x_n, t)$ *es verdadero)*

Eliminando variables de entrada

Consideremos un programa P que usa la entrada X_1 y X_2:

INSTRUCCIÓN 1 $\quad \#(I_1)$

$\vdots$

INSTRUCCIÓN k $\quad \#(I_k)$

Computa la función $f : \mathbb{N}^2 \to \mathbb{N}$

$$f(x, y) = \Psi_P^{(2)}(x, y)$$

$$\#(P) = [\#(I_1), \dots, \#(I_k)] - 1$$

Busco número de programa P_0 para $f_0 : \mathbb{N} \to \mathbb{N}, f_0(x) = f(x, 0)$

[A] $X_2 \leftarrow X_2 - 1$ $\quad 109$

IF $X_2 \neq 0$ GOTO A $\quad 110$

INSTRUCCIÓN 1 $\quad \#(I_1)$

$\vdots$

INSTRUCCIÓN k $\quad \#(I_k)$

Computa la función $f_0 : \mathbb{N} \to \mathbb{N}$

$$f_0(x) = \Psi_{P_0}^{(1)}(x)$$

$$\#(P_0) = [109, 110, \#(I_1), \dots, \#(I_k)] - 1$$

(Supongo que A no aparece como etiqueta en P; si aparece elijo otro nombre de etiqueta)

Eliminando variables de entrada

Busco número de programa P_1 para $f_1 : \mathbb{N} \to \mathbb{N}, f_1(x) = f(x, 1)$

[A] $X_2 \leftarrow X_2 - 1$	109
IF $X_2 \neq 0$ GOTO A	110
$X_2 \leftarrow X_2 + 1$	26
INSTRUCCIÓN 1	$\#(I_1)$
⋮	
INSTRUCCIÓN k	$\#(I_k)$

Computa la función $f_1 : \mathbb{N} \to \mathbb{N}$

$$f_1(x) = \Psi^{(1)}_{P_1}(x)$$

$$\#(P_1) = [109, 110, 26, \#(I_1), \ldots, \#(I_k)] - 1$$

Busco número de programa P_2 para $f_2 : \mathbb{N} \to \mathbb{N}, f_2(x) = f(x, 2)$

[A] $X_2 \leftarrow X_2 - 1$	109
IF $X_2 \neq 0$ GOTO A	110
$X_2 \leftarrow X_2 + 1$	26
$X_2 \leftarrow X_2 + 1$	26
INSTRUCCIÓN 1	$\#(I_1)$
⋮	
INSTRUCCIÓN k	$\#(I_k)$

Computa la función $f_2 : \mathbb{N} \to \mathbb{N}$

$$f_2(x) = \Psi^{(1)}_{P_2}(x)$$

$$\#(P_2) = [109, 110, 26, 26, \#(I_1), \ldots, \#(I_k)] - 1$$

Teorema del Parámetro

Hay un programa P_{x_2} para la función $f_{x_2}(x_1) = f(x_1, x_2)$

La transformación $(x_2, \#(P)) \mapsto \#(P_{x_2})$ es p.r., es decir, existe una función $S : \mathbb{N}^2 \to \mathbb{N}$ p.r. tal que dado x_2 e $y = \#(P)$ calcula $\#(P_{x_2})$:

$$S(x_2, y) = \left(2^{109} \cdot 3^{110} \cdot \prod_{j=1}^{x_2} p_{j+2}^{26} \cdot \prod_{j=1}^{|y+1|} p_{j+x_2+2}^{\ (y+1)[j]} \right) - 1$$

Teorema

Hay una función p.r. $S : \mathbb{N}^2 \to \mathbb{N}$ *tal que*

$$\Phi_y^{(2)}(x_1, x_2) = \Phi_{S(x_2,y)}^{(1)}(x_1).$$

Teorema

Para cada $n, m > 0$ *hay una función p.r. inyectiva* $S_m^n : \mathbb{N}^{n+1} \to \mathbb{N}$ *tal que*

$$\Phi_y^{(n+m)}(x_1, \ldots, x_m, u_1, \ldots, u_n) = \Phi_{S_m^n(u_1,\ldots,u_n,y)}^{(m)}(x_1, \ldots, x_m)$$

Programas autoreferentes

- en la demostración del Halting Problem construimos un programa P que, cuando se ejecuta con su mismo número de programa (i.e. $\#(P)$), evidencia una contradicción
- en general, los programas pueden dar por supuesto que conocen su mismo número de programa
- pero si un programa P conoce su número de programa, podría, por ejemplo, devolver su mismo número, i.e. $\#(P)$

Teorema de la Recursión

Teorema

Si $g : \mathbb{N}^{n+1} \to \mathbb{N}$ *es parcial computable, existe un* e *tal que*

$$\Phi_e^{(n)}(x_1, \ldots, x_n) = g(e, x_1, \ldots, x_n)$$

Demostración.

Sea S_n^1 la función del Teorema del Parámetro:

$$\Phi_y^{(n+1)}(x_1, \ldots, x_n, u) = \Phi_{S_n^1(u,y)}^{(n)}(x_1, \ldots, x_n).$$

La función $(x_1, \ldots, x_n, v) \mapsto g(S_n^1(v, v), x_1, \ldots, x_n)$ es parcial computable, de modo que existe d tal que

$$\begin{aligned} g(S_n^1(v, v), x_1, \ldots, x_n) &= \Phi_d^{(n+1)}(x_1, \ldots, x_n, v) \\ &= \Phi_{S_n^1(v,d)}^{(n)}(x_1, \ldots, x_n) \end{aligned}$$

d está fijo; v es variable. Elegimos $v = d$ y $e = S_n^1(d, d)$. $\square$

Teorema de la Recursión

Corolario

Si $g : \mathbb{N}^{n+1} \to \mathbb{N}$ *es parcial computable, existen infinitos* e *tal que*

$$\Phi_e^{(n)}(x_1, \ldots, x_n) = g(e, x_1, \ldots, x_n)$$

Demostración.

En la demostración del teorema anterior, existen infinitos d tal que

$$\Phi_d^{(n+1)} = g(S_n^1(v, v), x_1, \ldots, x_n).$$

$v \mapsto S_n^1(v, v)$ es inyectiva de modo que existen infinitos

$$e = S_n^1(d, d).$$

$\square$

Quines

Un quine es un programa que cuando se ejecuta, devuelve como salida el mismo programa.

Por ejemplo:

```
char*f="char*f=%c%s%c;main()
{printf(f,34,f,34,10);}%c";
main(){printf(f,34,f,34,10);}
```

Quines

¿Existe e tal que $\Phi_e(x) = e$?

Sí, el programa vacío tiene numero 0 y computa la función constante 0, i.e. $\Phi_0(x) = 0$.

Proposición

Hay infinitos e tal que $\Phi_e(x) = e$.

Demostración.

Considerar la función $g : \mathbb{N}^2 \to \mathbb{N}, g(z, x) = z$.
Aplicando el Teorema de la Recursión, existen infinitos e tal que

$$\Phi_e(x) = g(e, x) = e.$$

$\square$

Quines

No hay nada especial con que la salida del programa sea su propio número en el resultado anterior. Funciona para cualquier h parcial computable.

¿Existe e tal que $\Phi_e(x) = h(e)$?

Proposición

Hay infinitos e tal que $\Phi_e(x) = h(e)$.

Demostración.

Considerar la función $g : \mathbb{N}^2 \to \mathbb{N}, g(z, x) = h(z)$.
Aplicando el Teorema de la Recursión, existen infinitos e tal que

$$\Phi_e(x) = g(e, x) = h(e).$$

□

Teorema del Punto Fijo

Teorema

Si $f : \mathbb{N} \to \mathbb{N}$ *es computable, existe un* e *tal que* $\Phi_{f(e)} = \Phi_e$*.*

Demostración.

Considerar la función $g : \mathbb{N}^2 \to \mathbb{N}$,

$$g(z, x) = \Phi_{f(z)}(x).$$

Aplicando el Teorema de la Recursión, existe un e tal que para todo x,

$$\Phi_e(x) = g(e, x) = \Phi_{f(e)}(x)$$

$\square$

Ejercicio

Probar que $f : \mathbb{N} \to \mathbb{N}$,

$$f(x) = \begin{cases} 1 & \Phi_x \text{ es total} \\ 0 & \text{si no} \end{cases}$$

no es computable.

Supongamos f computable. Puedo definir el siguiente programa P:

$$[A] \qquad \text{IF } f(X) = 1 \text{ GOTO } A$$

Tenemos

$$\Psi_P^{(2)}(x, y) = g(x, y) = \begin{cases} \uparrow & \Phi_x \text{ es total} \\ 0 & \text{si no} \end{cases}$$

es parcial computable. Por el Teorema de la Recursión, sea e tal que $\Phi_e(y) = g(e, y)$.

- Φ_e es total $\Rightarrow g(e, y) \uparrow$ para todo $y \Rightarrow \Phi_e(y) \uparrow$ para todo y $\Rightarrow \Phi_e$ no es total
- Φ_e no es total $\Rightarrow g(e, y) = 0$ para todo $y \Rightarrow \Phi_e(y) = 0$ para todo $y \Rightarrow \Phi_e$ es total