Lógica Superior:

Computabilidad e Incompletitud

Profesor: Eduardo Barrio
UBA - Filosofía
2do cuatrimestre de 2015

Funciones y computabilidad

¿Qué funciones son computables?

Tesis de Church:

Cualquier función efectivamente computable es una función recursiva.

Funciones Recursivas

Funciones Computables:

- Funciones computables por M Turing
- Funciones recursivas

¿Qué es una función?

Relación entre dos conjuntos

Para todo valor en un dominio se asigna un (y sólo un) elemento en codominio.

f(x): 0 si x es 0

1 si x es mayor que 0

Hay dos condiciones que se deben cumplir para calcular la función

Funciones Recursivas

Funciones Efectivamente Computables:

- funciones para las cuales hay reglas explícitas bien definidas que al ser seguidas permitirían (en principio) calcular el valor de una función

Funciones Recursivas

¿Qué funciones son computables?

Recursividad: descomponer un problema para encontrar una solución en los componentes simples.

Descomposición que permite encontrar mecánicamente el valor de la función.

Funciones Recursivas

f(x) = si x=0, entonces f(x):1

si x mayor a 0, entonces f(x) = x . f(x-1)

f(1) = si x=1, entonces f(x):1

f(3) = si x=3, entonces f(3) = 3 . f(3-1)

f(3) = 3 . f(2)

3 . f(2) = x . f(x-1) = 2 . f(2-1) = 2 . f(1) = 2 . 1 = 2

3 . 2 = 6

Funciones Recursivas

f(x) = si x=0, entonces f(x):1

si x mayor a 0, entonces f(x) = x . f(x-1)

Hay un **algoritmo** tal que para cualquier valor que tome f(x), en un número finito de pasos, nos permite calcular su valor.

Funciones Recursivas

Las funciones recursivas coinciden con las funciones Turing-computables.

Un subconjunto importante de las funciones recursivas es el de las funciones recursivas primitivas (PR)

1. - Las Funciones PR son un subconjunto de las Funciones recursivas.
2. - Las Funciones PR son siempre funciones totales:
3. - Incluyen a las funciones aritméticas más comunes
    1. - Sucesor
    2. - Suma
    3. - Multiplicación
    4. - División
    5. - Potencia
    6. - Factorial

Recursividad

¿Qué es una función recursiva?

Simplemente se identifica un conjunto (obviamente) computable de funciones básicas que junto con algunas operaciones mecánicas (composición, recursión primitiva y minimización) generan nuevas funciones (que son recursivas).

Recursión

- Descomponer una función en sus componentes básicos.
- Reducir una función a sus funciones constitutivas.
- Eliminar una función a favor de sus funciones constitutivas.

Recursividad

Función Recursiva: exponenciación

5^3 : 5 x 5 x 5 x 1 : (1 + 1 + 1+ 1+ 1 , ..., + 1)

125 (el sucesor de x, 125 veces)

Exp.

Recursión

- Escribir una función en sus componentes más básicos

CLASE DE FUNCIONES RECURSIVAS PRIMITIVAS

1) Funciones Básicas (Unidades Atómicas)

$z(0) = 0$, $z(0'''''') = 0$

$s(0) = 0'$ $s(0') = 0''$

$id^1_1 (x) = x$ $id^2_1 (x,y) = x$ $id^2_2 (x,y) = y$

2) Operaciones de Construcción:

• Composición : componer dos funciones recursivas da otra función que es recursiva

• Recursión Primitiva: componer recursivamente una nueva función desde una función recursiva da otra función recursiva.

- caso base
- cláusula de recursión.

• Minimización: para computar Mn[f](x), se computa f(x, 0), f(x, 1), ... hasta que se encuentra un y tal que f(x, y) = 0.

Si hay tal y, entonces Mn[f](x) = el menor y;
Si no hay tal y, entonces Mn[f](x) es indefinido.

3) Nada pertenece a la clase si no es una función primitiva o construída por medio de composición o recursión.

CLASE DE FUNCIONES RECURSIVAS PRIMITIVAS

1) Funciones Básicas (Unidades Atómicas)

Funciones recursivas Básicas: z(x), s(x) and id_k

Composición

Supongamos que f es una función de *m* argumentos **PR** y cada $g_1,..., g_m$ es una función de *n* argumentos. Entonces obtenemos la función compuesta:

$h(x_1,..., x_n) = f(g_1(x_1,..., x_n), ..., g_m(x_1,..., x_n))$

Recursión primitiva

h se define por recursión primitiva desde f (n-lugares) y g (n+2-lugares) si

$h(x_1, x_2,..., x_n, 0) = f(x_1,..., x_n)$
$h(x_1, x_2,..., x_n, y') = g(x_1, x_2,..., x_n, y, h(x_1,..., x_n, y))$

Minimización

Hay FUNCIONES PARCIALES RECURSIVAS
Si f es una función de n+1 argumentos,

Mn[f]($x_1,..., x_n$) = { y, si $f(x_1,..., x_n, y) = 0$ y $f(x_1,..., x_n, t)$ es definido y > 0 para t < y

{indefinido, si no hay tal y.}

CLASE DE FUNCIONES RECURSIVAS PRIMITIVAS

1) Funciones Básicas (Unidades Atómicas)

Composición

Supongamos que f es una función de *m* argumentos **PR** y cada $g_1,\ldots, g_m$ es una función de *n* argumentos.
Entonces obtenemos la función compuesta:

$h(x_1,\ldots, x_n) = f(g_1(x_1,\ldots, x_n), \ldots, g_m(x_1,\ldots, x_n))$

A. Composición. Ejemplos.

1: Es $f(x) = x + 2$ p.r.? $f(x)= s(s(x))$

Método:

Si se puede escribir una nueva función h como una composición de funciones que sabemos son p.r., entonces h es p.r.

CLASE DE FUNCIONES RECURSIVAS PRIMITIVAS

1) Funciones Básicas (Unidades Atómicas)

Recursión primitiva

h se define por recursión primitiva desde f (n-lugares) y g (n+2-lugares) si

$h(x_1, x_2,..., x_n ,0) = f(x_1,..., x_n)$
$h(x_1, x_2,..., x_n ,y') = g(x_1, x_2,..., x_n, y, h(x_1,..., x_n,y))$

1. sum(x, 0) = x [clausula base]
   sum(x, y') = [sum(x, y)]‘ [clausula de recursión]

CLASE DE FUNCIONES RECURSIVAS

1) Funciones Básicas (Unidades Atómicas)

Minimización

Hay FUNCIONES PARCIALES RECURSIVAS

1: Mn[sum].

Mn[sum](x) = el menor y tal que sum(x, y) = 0, o indefinido

Mn[sum](0) = 0, y Mn[sum](x) es indefinido, si $x \neq 0$.

Para computar Mn[f](x), se computa f(x, 0), f(x, 1), ... hasta que se encuentra un y tal que f(x, y) = 0.

Si hay tal y, entonces Mn[f](x) = el menor y;
Si no hay tal y, entonces Mn[f](x) es indefinido.

Definir la función raíz tal que raíz(x) = $\sqrt{x}$.

Para valores cuya raíz cuadrada no es entera, la función no está definida.

raiz(x) = $\sqrt{x}$ = min {$t \in N_{\text{úmero natural}}$ tal que $t^2 = x$}

MÁS ALLÁ DE LAS FUNCIONES RECURSIVAS

1. Funciones complicadas de la aritmética (SUMA, MULTIPLICACION, POTENCIA, RAÍZ, etc) se obtienen por composición (composition), recursición primimiva (primitive recursion) y minimización (minimization).
2. ¿Toda función de la aritmética (números naturales) será recursiva (o recursiva primitiva o obtenible a partir de composición, recursión y minimización?

MÁS ALLÁ DE LAS FUNCIONES RECURSIVAS PRIMITIVAS

No todas las funciones computables son funciones primitivas recursivas.

Argumento:

- Toda función p.r. es una función total.
- Cada función p.r. puede expresarse como una cadena finita de símbolos a partir de un vocabulario finito.
- La lista de funciones p.r. totales puede diagonalizarse.
- La diagonalización produce una función computable que no es p.r.

Por lo tanto, no toda función computable es p.r.

Objetivo: obtener todas las funciones computables.

No todas las funciones computables son funciones primitivas recursivas.

- Estamos forzados a considerar **funciones parciales.**

Dos argumentos para considerar funciones parciales:

- Hay funciones parciales que son computables y que no pueden reducirse a una función total.

- La diagonalización (en el caso de las parciales) no permite obtener una función computable

Funciones parciales y diagonalización:

Argumento:

La diagonalización sobre funciones parciales no produce una función computable que no esté en la lista.
- Por ejemplo, supongamos que hemos codificado numéricamente todos los algoritmos y tenemos la lista de ellos. Aea fn una función parcial computada por el algoritmo n.
Supongamos que g(x): f(x) + 1 si f(x) está definido y g(x) está indefinido (en otro caso).
Si g(x1) corresponde al 1er algoritmo, ya que f(x) es parcial, la diagonalización no implica que g ≠ f(x1), ya que su valor para f(x1) puede estar indefinido.

Por lo tanto, no se obtiene una función computable que no está en la lista.

MÁS ALLÁ DE LAS FUNCIONES RECURSIVAS

¿Existirán funciones de números naturales que no son funciones recursivas?

- Tesis de Church: Las funciones computables son funciones recursivas

- Según la tesis de Church, sería lo mismo que preguntarnos si existen funciones de números naturales que no son calculables (mecánicamente).

- Es decir, su valor estaría determinado, pero no existiría un algorítmo para calcularlas.

RESPUESTA: Hay funciones que no son recursivas.