Recursión Primitiva

Alberto Acosta

Febrero 2019

Antes del sigo XIX, los matemáticos utilizaban el principio de definición de una función mediante la inducción. Dedeking, en 1888, provó, usando axiomas aceptados, que tal definición define una función *única* y la aplicó para las funciones $m+n, m \times n, m^n$. Basado en el trabajo de Dedekind, Peano en 1889 y 1891 escribió los 5 axiomas para los enteros positivos. Como acompañante del quinto axioma, Peano utilizó la definición por inducción para llamar la *recursión primitiva*, dada por el siguiente esquema:

$$Scheme(V) = \begin{cases} f(0, \vec{y}) = h(\vec{y}) \\ f(x+1, \vec{y} = g(x, f(x, \vec{y}), \vec{y}) \end{cases}$$

donde g, h son funciones previamente definidas y $\vec{y}$ denota una secuencia (posiblemente vacía) de variables adicionales. [3]

Hilbert (1904) propuso provar la consistencia de la aritmética mediante lo que después fue conocido como su *"finist program"*. Propuso utilizar la finitez de las pruebas matemáticas para establecer contradicciones que no podían ser derivadas. Esto tendió a reducir las pruebas a una manipulación de cadenas finitas de símbolos derivados de significados intuitivos, que estimulaban el desarrollo de procesos mecánicos.

Posteriormente en los 1930's, Göedel, Church, Kleene, Turing y otros lograron unir la aciomatización de la geometría de Hilbert (1899). En 1930, Church había estudiado una clase de funciones efectivamente calculables llamadas *"λ-definable functions"*, para que luego, en 1933, Kleene mostró que una gran clase de números teóricos eran λ-definidos. [2]

Finalmente, las funciones computables y que fueron nombradas axiomas que probaban el teorema de Turing (una función es computable si puede ser calculada por un humano computador idealizado), son:

1. Sucesor: $S(x) = x + 1$
2. Cero: $N(x) = 0$
3. Identidad generalizada: $U_i^{(n)}(x_1, \ldots, x_n) = x_i$

[1]

De las cuales surgen las siguientes:

1. Suma: $x + y$

2. Multiplicación: $x \times y$
3. Exponenciación: x^y
4. Factorial: $0! = 1; x\prime! = x! \times x\prime$
5. Predecesor: Si x>0, entonces x -1, si no, 0.
6. Resta: $x \dot{-} y$. Si $x \geq y$, entonces $x - y$, si no, 0.
7. Mínimo: $(x_1, \ldots, x_n)$
8. Máximo: $(x_1, \ldots, x_n)$
9. Diferencia absoluta: $|x - y| = (x \dot{-} y) + (y \dot{-} x)$
10. s̃g(x): Si x = 0, entonces 1, si no, 0.
11. sg(x): Si x = 0, entonces , si no, 1.
12. $x|y$ si $y = k \times x$ para algún k entonces 0, si no, 1.
13. Residuo (x,y), también llamado Módulo(x,y)
14. x = y: sg(— x - y—)
15. $x < y : sg(x\prime \dot{-} y)$
16. Pr(x): x es un número primo si Pr(x) = x>1 & no existe c que [c—x]
17. p_i es el *i-ésimo* + 1 número primo.

Referencias

[1] Frank S. Beckman. *Mathematical Foundations of Programming*. 1981. ISBN: 0-201-14462-X.

[2] Daniel E. Severin. *Unary primitive recursive functions*. URL: `https://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.jsl/1230396909`.

[3] Robert I. Soare. *Computability and Recursion*. URL: `http://www.people.cs.uchicago.edu/~soare/History/compute.pdf`.