10 Problema de Hilbert

Leonel Contreras

UVG

23 noviembre 2021

Introducción

En 1900 en el segundo Congreso Internacional de Matemáticas -ICM por sus siglas en inglés- el famoso matemático David Hilbert propone una lista de 23 problemas abiertos. Hilbert pensaba que estos problemas debían guiar la investigación matemática en el siglo XX.

Nosotros estaremos interesados en el décimo problema de dicha lista. Este problema se distingue por su simpleza. El problema pregunta si existe un algoritmo el cual, dado una ecuación diofantica cualquiera, determina si esta tiene solución o no.

Equivalencia del Problema en los Enteros y Narurales

Existe un procedimiento para determinar si una ecuacion diofantica tiene solucion en los naturales ssi existe un procedimiento para determinar si una ecuacion diofantica tiene solucion en los enteros

Funciones Efectivamente Calculables

Consideraremos funciones que mapean N^n a los naturales. LLamaremos a una funcion efectivamente calculable si tenemos un algoritmo para calcular el valor de la funcion para cualquier elemento del dominio. Sin embargo, para nuestro objetivo -el 10 problema de Hilbert- necesitamos una definicion formal.

Formalización del concepto

Kurt Gödel

Alan Turing

Alonzo Church

Tesis de Turing-Church

Toda función computable por una maquina de Turing es efectivamente computable.

Maquinas de Turing

Una maquina de Turing es una cuadrupla: (Q, Σ, s, δ). Donde Q es un conjunto finito de estados, Σ es un conjunto finito de simbolos, s es el estado inicial y δ es la funcion transicion $\delta : Q \times \Sigma \to \Sigma \times \{L, R\} \times Q$. Si $\delta(q_i, S_j) = (S_{i,j}, D, q_{i.j)}$ entonces cuando la maquina este en el estado q_i leyendo el simbolo S_j la maquina reemplazara S_j por $S_{i,j}$ y se movera en la direccion $D \in \{R, L\}$ y se pone en el estado $q_{i,j}$

Ejemplo Maquina de Turing

Tabla para la maquina de f(a)=a+1

Estado de Maquina	Condicion del Cuadrado Escaneado	
	0	1
1	C0	R2
2	R3	R9
3	PL4	R3
4	L5	L4
5	L5	L6
6	R2	R7
7	R8	ER7
8	R8	R3
9	PR9	L10
10	C0	ER11
11	PC0	R11

Ejemplo Maquina de Turing

Calculo para a=1

Momento	Cinta							
0	0	1	1^1	0	0	0	0	
1	0	1	1	0^2	0	0	0	
2	0	1	1	0	0^3			
3	0	1	1	0^4	1			
4	0	1	1^5	0	1	0	0	
5	0	1^6	1	0	1	0	0	
6	0	1	1^7	0	1	0	0	
7	0	1	0	0^7	1	0	0	
8	0	1	0	0	1^8	0	0	
9	0	1	0	0	1	0^3	0	
10	0	1	0	0	1^4	1	0	
11	0	1	0	0^4	1	1	0	
12	0	1	0^5	0	1	1	0	
13	0	1^5	0	0	1	1	0	
14	0^6	1	0	0	1	1	0	
15	0	1^2	0	0	1	1	0	
16	0	1	0^9	0	1	1	0	
17	0	1	1	0^9	1	1	0	
18	0	1	1	1	1^9	1	0	
19	0	1	1	1^10	1	1	0	
20	0	1	1	0	1^11	1	0	
21	0	1	1	0	1	1^11	0	
22	0	1	1	0	1	1	0^11	
23	0	1	1	0	1	1	1^0	
24	0	1	1	0	1	1	1^0	

Halting Problem

No existe una maquina de Turing que determine si una maquina de Turing se detiene en una entrada determinada

Funciones Recursivas Primitivas

Las funciones recursivas primitivas son:

1. **La función Constante** $C_0(x) = 0$
2. **La función Sucesor** $S(x) = x + 1$
3. **Las funciones Proyección** $P_i^n(x_1, ..., x_n) = x_i$

Las operaciones que podemos aplicar son:

1. **Composición** Para $f(t_1, ..., t_m)$ y $g_1(x_1, ..., x_n), ..., g_m(x_1, ..., x_n)$ la composición es $h(x_1, ..., x_n) = f(g_1(x_1, ..., x_n), ..., g_m(x_1, ..., x_n))$
2. **Recursión Primitiva** Dadas funciones $f(x_1, ..x_n$ y $g(t_1, ..., t_{n+1})$ la recursion primitiva produce una funcion $h(x_1, ..., x_n, z)$ que satisface:
   $h(x_1, ..., x_n, 0) = f(x_1, ..., x_n)$
   $h(x_1, ..., x_n, t + 1) = g(t, h(x_1, ..., x_n, t), x_1, ..., x_n)$

Funcion de Ackerman

Se puede probar que todas las funciones primitivamente recursivas son computables; el converso no es cierto.

$$A(m,n) = \begin{cases} n+1, & \text{si } m = 0; \\ A(m-1,1), & \text{si } m > 0 \text{ y } n = 0; \\ A(m-1, A(m, n-1)), & \text{si } m > 0 \text{ y } n > 0 \end{cases}$$

Funciones Recursivas Parciales

Göedel agrega otra operacion a nuestro conjunto anterior -el de las funciones recursivas primitivas- y obtenemos este conjunto nuevo de funciones recursivas.

La nueva operacion se llama **minimización** la cual, dada una funcion $f(z, x_1, ..., x_n)$ produce una nueva funcion parcial* $h(x_1, ..., x_n) = min_z\{f(z, x_1, ..., x_n) = 0\}$.Si tal z no existe entonces h esta indefinida.

Al agregar la minimizacion a la composicion y recursion primitiva como operacion sobre la funcion constante, sucesor y proyeccion tenemos el conjunto de funciones recursivas parciales.

Funciones Recursivas

Ahora bien si agregamos la condicion que la funcion h producida por minimizacion sea total*, las funciones resultantes son las funciones recursivas.

Por qué nos interesan las funciones recursivas?

Los polinomios con coeficientes naturales son primitivamente recursivos. Esto debido a que son obtenidos por una aplicacion finita de la suma y multiplicacion de constantes y variables. Ademas para probar que una funcion es recursiva ssi es Diofantina, usamos las funciones: Cociente, Residuo, suma, multiplicacion, la funcion beta de Gödel y la funcion de emparejamiento de Cantor; las cuales son recursivas.

$$Quo(a,b) = min_q\{Z(sub((q+1)(b+1),a)) = 0\} = min_q\{Z(sub(M(S(q),S(b)),a)) = 0\}$$

$$Rem(a,b) = sub(a, q(b+1)) = sub(a, M(Quo(a,b), S(b)))$$

$$P(x.y) = \frac{(x+y)(x+y+1)}{2} + x = Quo((x+y)(x+y+1),1) + x = A(Quo(M(A(x,y),A(x,S(y))),1),x))$$

Teorema

Una funcion es computable ssi es recursiva

Conjuto Computablemente Enumerable

Un conjunto de numeros naturales es computablemente enumerable si es vacio o es igual al rango de una funcion computable f.

Conjunto Decidible

Si un conjunto de numeros naturales y su complemento son computablemente enumerables entonces decimos que dicho conjunto es decidible.

Teorema

Existe un conjunto de números naturales que es computablemente enumerable pero no es decidible. Existe un conjunto de números naturales que no es computablemente enumerable

$U = \{2^m 3^n \mid (m, n) \in S\}$ es un conjunto de numeros naturales que es computablemente enumerable pero no decidible.

Conjuntos Diofanticos

Un S conjunto de n-tuplas de numeros naturales es Diofantico so existe un numero natural m y un polinomio $P(x_1, ..., x_n, y_1, ..., y_m)$ con coeficientes enteros tal que
$(x_1, ..., x_n) \in S \exists y_1, ..., \exists y_m, P(x_1, ..., x_n, y_1, ..., y_m) = 0$

Funcion Diofantica

Un mapeo $f : N^n \to N$ es Diofantico si el conjunto $\{(x_1, ..., x_n, y) \mid y = f(x_1, ..., x_n)\}$ es un conjunto Diofantico.

Teorema

Una funcion es Diofantica ssi es recursiva

Teorema

Un conjunto de números naturales es Diofantico si y solo si es computablemente enumerable

Asume la tesis de Turing Church

Solucion al 10 Problema de Hilbert

No existe un algoritmo que determine si una ecuacion Diofantica arbitraria tiene solucion en los numeros naturales

Por teorema existe un conjunto U de números naturales que es computablemente enumerable pero no decididle. Ya que U es computablemente enumarable, por el teorema 6.1 es Diofantico. Entonces $x \in U \leftrightarrow \exists y_1, ..., \exists y_n \ni P(x, y_1, ..., y_m) = 0$. Suponga que si existe un algoritmo para determinar si dada una ecuación diofantica tiene soluciones en los naturales. Entonces, dado x podríamos usar dicho algoritmo para decidir si $P(x, y_1, ..., y_n)$ tiene solución en $y_1, ..., y_n$. Es decir, tendríamos un algoritmo para decidir si x pertenece o no a U. Por ende, U es decidible$(\rightarrow\leftarrow)$