Boletín de Matemáticas
Volúmen X (1976). págs. 1 - 27

ALGORITMOS, NUMEROS DE FIBONACCI Y EL DECIMO PROBLEMA DE HILBERT(*)

XAVIER CAICEDO FERRER

1. Introducción.

Ciertos problemas de las matemáticas han sido fructíferos para su desarrollo, en el sentido de que los intentos para resolverlos han conducido a nuevas teorías y a un análisis mas refinado y profundo de los conceptos de que el problema trata. Este criterio se aplica a varios de los problemas que presentó David Hilbert, en el Segundo Congreso Internacional de Matemáticas que se celebró en París, en 1900. El décimo en la famosa lista era el siguiente : encontrar un algoritmo para decidir de cualquier ecuación diofántica en un número arbitrario de variables si ésta tiene soluciones enteras o nó.

Un algoritmo es un método cuya ejecución consiste en la aplicación, paso a paso, de ciertas reglas especificadas a priori. Estas reglas deben determinar el resultado final completamente. Las operaciones básicas de la aritmética : sumar, restar, multiplicar, dividir son algoritmos. Los cálculos que un piloto de navío ejecuta para determinar su posición geográfica se pueden considerar como un algoritmo. Un programa de computador es un algoritmo. Es claro que el concepto va más allá de sus aplicaciones numéricas. La palabra algoritmo viene del nombre

(*) Conferencia presentada por el autor en el V Coloquio Colombiano de Matemáticas; Medellín, 1975. *N. del E.*

del matemático árabe Al-Kwarizmi (siglo IX) en cuyos escritos aparecen muchos de los métodos que aún usamos hoy día en álgebra y aritmética(1).

Una ecuación *diofántica* (2) en las variables $x_1, \ldots, x_k$ es una ecuación que se puede escribir en la forma

$$P(x_1, \ldots, x_k) = 0$$

en donde $P(x_1, \ldots, x_k)$ es un polinomio con coeficientes enteros. Un ejemplo muy sencillo sería una ecuación de segundo grado, como $x^2 - 7x + 12 = 0$, cuya solución se aprende en la escuela secundaria ($x = 3$ ó $x = 4$ son soluciones). En cambio la ecuación : $x^2 - \sqrt{3}x + 17 = 0$ no sería diofántica pues tiene un coeficiente que no es entero. En general, una ecuación diofantica puede tener más de una variable, por ejemplo

$$5x_1 + 10x_1 x_2^3 - x_1 x_2 x_3 + 17 = 0$$

Una solución de enteros de esta ecuación sería $x_1 = -1$, $x_2 = 1$, $x_3 = -2$. Sin embargo, no toda ecuación diofántica tiene soluciónes enteras. La ecuación : $x^2 - 2x - 1 = 0$ tiene por únicas soluciones los números $1 + \sqrt{2}$, $1 - \sqrt{2}$, ninguno de los cuales es entero.

Hilbert no pedía un método para encontrar las soluciones de una ecuación diofanti a, sino algo más simple, un método para decidir si estas soluciones existen o nó. La solución a este problema, obtenida 70 años después, se basa en resultados combinados de Teoría de Números y Lógica Matemática. En particular, requirió una definición precisa del concepto de algoritmo. Los trabajos de Kurt Gödel, Martin Davis y Julia Robinson (entre otros) prepararon el camino para que el matemático ruso Yuri V. Matijasevich diera el último paso en la solución del problema en 1970 (tenía 22 años). La solución es muy sencilla : no hay tal algoritmo, *el décimo problema de Hilbert es insoluble* [5]. Este resul-

(1) El mismo origen tiene la palabra "guarismo" que usamos como sinónimo de "cifra".

(2) De Diofanto de Alejandría, matemático griego que estudió dichas ecuaciones (siglo III).

tado tiene implicaciones prácticas porque nos dice, por ejemplo, que dentro de la concepción actual de los computadores electrónicos, es imposible programar uno de éstos para que dada una ecuación diofantina (o la lista de sus coeficientes) decida si ésta tiene soluciones enteras.. Por otra parte es un ejemplo espléndido de las limitaciones y el poder de las matemáticas : por una parte se demuestra que el problema no se puede resolver por medio de ciertos métodos matemáticos ; por otra, esta demostración se hace usando métodos matemáticos ! Trataremos de describir en estas notas cómo se llegó a dicho descubrimiento, y algunas de sus consecuencias más interesantes.

2. Algoritmos.

Es posible dar un algoritmo para decidir si una ecuación diofántica *lineal*, es decir una de la forma :

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + a_{n+1} = 0$$

en donde $a_1, \ldots, a_n, a_{n+1}$ son enteros, tiene soluciones enteras. El algoritmo se basa en el siguiente teorema elemental de la Teoría de Números. Usamos $M(a_1, \ldots, a_n)$ para denotar el máximo común divisor de los números $a_1, \ldots, a_n$. La notación $n \mid m$ significa n divide a m.

Teorema. La ecuación diofántica $a_1 x_1 + \cdots + a_n x_n + a_{n+1} = 0$ tiene solución si y sólamente si $M(a_1, \ldots, a_n) \mid a_{n+1}$ (Vea [3] ó [9]).

El algoritmo de decisión consiste en los siguientes pasos que terminan en un resultado F al cual podemos dar el valor 1 (hay soluciones enteras) ó 0 (no las hay).

A. Calcule $b = M(a_1, \ldots, a_n)$

B. Si $b \mid a_{n+1}$, $F = 1$. Si $b \nmid a_{n+1}$, $F = 0$.

El cálculo de $M(a_1, \ldots, a_n)$ se puede realizar repitiendo varias veces la operación de tomar el máximo común divisor de dos números, pues :

$$M(a_1, a_2, a_3) = M(M(a_1, a_2), a_3)$$

$$M(a_1, a_2, a_3, a_4) = M(M(a_1, a_2, a_3), a_4)$$

etc.

$M(x,y)$ se calcula, a su vez, por medio de un algoritmo famoso, el Algoritmo de Euclides, que no es mas que la repetición iterada del algoritmo de división [9]. La verificación de si un número a divide o nó a un número b se puede reducir también al algoritmo de división, pues :

$$a \mid b \Longleftrightarrow \text{Residuo de } b \div a \text{ es } 0.$$

Si llamamos $R(x/y)$ al residuo de dividir x por y e introducimos la función $I(x)$ tal que $I(0) = 1$, $I(x) = 0$ si $x \neq 0$, vemos que el algoritmo para decidir la solubilidad de las ecuaciones diofantinas lineales se reduce a una función de sus coeficientes

$$F(a_1, \ldots, a_n, a_{n+1}) = I(R(a_{n+1} / M(a_1, \ldots, a_n)))$$

El ejemplo anterior ilustra dos hechos muy importantes : algoritmos son funciones (o métodos para calcular funciones) aún en el caso de algoritmo de decisión; y el cálculo de un algoritmo se puede reducir a la aplicación repetida de ciertas funciones básicas ($R(x/y)$, $I(x)$ serían suficientes en el ejemplo anterior) de acuerdo con ciertas instrucciones. Sin embargo debe hacerse una observación esencial : algoritmos son funciones *parciales* es decir, no necesariamente definidas en todas partes. Esto se debe a que los cálculos de un algoritmo pueden ser interminables. Considere por ejemplo, el algoritmo para obtener la raiz cuadrada de un número positivo. Veremos que en general *no* es posible decidir de antemano si un algoritmo termina o nó al aplicarlo a cierto dato.

Podría argumentarse que el ejemplo anterior es un algoritmo matemático, reducible a cálculos numéricos y por tanto no puede ilustrar el concepto mas general de algoritmo como un *proceso efectivamente realizable*, independiente de todo cálculo numérico. El hecho es que todo algoritmo es equivalente a uno que actúa sobre números. ¿Cuál es, en general, el dominio de aplicación de un algoritmo ?

Intuitivamente, los objetos a que se aplica un algoritmo son configuraciones finitas de ciertos objetos básicos, ya sea perforaciones en una tarjeta como en los algoritmos calculados por los computadores electrónicos, o sucesiones de símbolos como en el algoritmo de Euclides. Las sucesiones de símbolos sobre las que este último actúa son las expresiones decimales de los números, es decir, sucesiones finitas de símbolos tomados del alfabeto

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$$

en las cuales 0 no aparece como símbolo inicial. Un algoritmo para decidir la solubilidad de las ecuaciones diofánticas también actuaría esencialmente sobre sucesiones de símbolos, pues aun cuando la ecuación

$$125x_1^{\,2} - 10x_3^{\,9}x_1^{\,3}x_2 + 7 = 0$$

es una configuración bidimensional, podemos introducir símbolos $\uparrow$, $\downarrow$ y escribir :

$$125x\downarrow 1\uparrow 2 - 10x\downarrow 3\uparrow 9x\downarrow 1\uparrow 3x\downarrow 2 - 7 = 0.$$

Los nuevos símbolos indican sin ambiguedad los subíndices y los exponentes de las variables. El alfabeto en este caso sería

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9, x, \uparrow, \downarrow, -, +, =\}.$$

Lo que se ha dicho sobre el dominio de un algoritmo vale para su codominio, es decir, los objetos que resultan cuando éste termina, la expresión decimal de un número en el caso del algoritmo de decisión. No es difícil ver que toda configuración finita se puede "linealizar" es decir, codificar como una sucesión de símbolos, posiblemente con la ayuda de símbolos adicionales. Podemos suponer, pues, que todo algoritmo actúa en un espacio constituido por sucesiones finitas de símbolos de un alfabeto finito. Aun si el alfabeto fuera infinito : $\{s_1, s_2, s_3, \ldots\}$ lo podríamos identificar con las sucesiones $\{s1, s11, s111, \ldots\}$ reduciéndolo al alfabeto $\{s, 1\}$.

Dado $A = \{s_0, \ldots, s_n\}$, podemos identificar el símbolo s_i con el número natural i. De esta manera, toda sucesión de símbolos $s_{i_1} s_{i_2} \ldots s_{i_k}$ puede identificarse con la n-tupla de números $(i_1, i_2, \ldots, i_k)$, es decir, con un elemento del conjunto

$$\mathbb{N}^* = \bigcup_{k=1}^{\infty} \mathbb{N}^k \text{, en donde } \mathbb{N}^1 = \mathbb{N} = \{0, 1, 2, 3, \ldots\} \text{ y } \mathbb{N}^{k+1} = \mathbb{N}^k \times \mathbb{N}$$

Por otra parte, considere la sucesión creciente de los números primos : $p_1, p_2, p_3, \ldots$ La función $g: \mathbb{N}^* \to \mathbb{N}$ definida por

$$g(n_1, \ldots, n_k) = (p_1^{n_1 + 1} \ldots p_k^{n_k + 1}) - 2$$

es una biyección. La descomposición de todo natural mayor que 1 en factores primos garantiza que sea sobreyectiva. La unicidad de la descomposición implica que es inyectiva. Cada k-tupla queda codificada por un número natural, y hay algoritmos obvios para ir de una k-tupla a su código y viceversa. Esto significa que sin pérdida de generalidad podríamos reducirnos a estudiar algoritmos de $\mathbb{N}$ en $\mathbb{N}$. Por razones técnicas es mas conveniente considerar algoritmos de $\mathbb{N}^k$ en $\mathbb{N}$ $(k = 1, 2, 3, \ldots)$. Estos algoritmos se llaman *funciones recursivas* y se definen de una manera precisa en la sección siguiente.

3. Funciones recursivas.

Definición. Una función $f: \mathbb{N}^k \to \mathbb{N}$ es *recursiva* si consiste de una de las funciones básicas :

$$O(X) = 0$$

$$S(X) = x + 1$$

$$P_i^n(x_1, \ldots, x_n) = x_i \qquad (n = 1, 2, \ldots; \quad 1 \leq i \leq n)$$

o se puede obtener de otras funciones recursivas ya definidas por una de las reglas siguientes :

1. **Composición.** Si $g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)$ y $h(y_1, \ldots, y_k)$ son

recursivas, también lo es

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

II. Recurrencia. Si $g(x_1,\ldots,x_k)$ y $h(x_1,\ldots,x_k,z,w)$ son recursivas así lo es la función f definida por

$$f(x_1,\ldots,x_k,0) = g(x_1,\ldots,x_k)$$

$$f(x_1,\ldots,x_k,y+1) = h(x_1,\ldots,x_k,\ y\ ,\ f(x_1,\ldots,x_k,y))$$

III. Minimización. Si $g(x_1,\ldots,x_k,y)$ es recursiva y para todo $x_1,\ldots,x_k$ existe y tal que $g(x_1,\ldots,x_k,y) = 0$ entonces también es recursiva

$$f(x_1,\ldots,x_k) = \mu y\ (g(x_1,\ldots,x_k,y) = 0)$$

μy significa "el mínimo natural y tal que ..."

Definición. Si $g(x_1,\ldots,x_k,y)$ es recursiva, la función parcial

$$f(x_1,\ldots,x_k) = \mu y\ (g(x_1,\ldots,x_k,y) = 0)$$

definida solamente para aquellos $x_1,\ldots,x_k$ tales que existe y con $g(x_1,\ldots,x_k,y) = 0$, es una función *recursiva parcial.*

Las funciones básicas son trivialmente calculables y las reglas I, II, III se pueden considerar como instrucciones para calcular nuevas funciones. En particular, la regla III pide calcular consecutivamente los valores $g(\vec{x},0), g(\vec{x},1),\ldots$ ($\vec{x}$ denota la k-tupla $x_1,\ldots,x_k$.)) hasta obtener la primera n tal que $g(\vec{x},n) = 0$, y hacer $f(\vec{x}) = n$; si tal n no existe el cálculo no termina y la función queda indefinida en $\vec{x}$. En este caso f no es recursiva sino parcial.

Ejemplos.

i) $x+y$ se define tomando $g(x) = P_1^1(x) = x$, $h(x,z,w) = S(P_3^3(x,z,w)) = S(w)$

(regla I) y aplicando la regla II :

$$x + 0 = g(x) = x$$

$$x + S(y) = h(x, y, (x + y)) = S(x + y)$$

Similarmente para la multiplicación

$$x.0 = 0$$

$$x.S(y) = x + x.y$$

ii) Cada función constante $C_n(x) = n$ es recursiva pues $C_0(x) = O(x)$ y $C_n(x) = S(S(\ldots S(x)\ldots))$ n veces, para $n > 0$.

iii) La resta truncada : $x \dot{-} y = \begin{cases} x-y & \text{si } x \geq y \\ 0 & \text{si } x < y \end{cases}$ se define :

$$(A) \quad \begin{cases} h(0) = 0 \\ h(S(x)) = x \end{cases} \qquad (B) \quad \begin{cases} x \dot{-} 0 = x \\ x \dot{-} S(y) = h(x \dot{-} y) \end{cases}$$

iv) La función $|x-y| = (x \dot{-} y) + (y \dot{-} x)$ es tal que $|x-y| = 0 \Leftrightarrow x = y$.

v) $I(x) = \mu \tilde{y}(|x.y - x|) = 0$ es tal que $I(0) = 0$, $I(x) = 1$ si $x \neq 0$.

vi) Si $P(x_1, \ldots, x_k)$ es un polinomio con coeficientes enteros (incluyendo negativos) entonces

$$P(x_1, \ldots, x_k) = P_1(x_1, \ldots, x_k) - P_2(x_1, \ldots, x_k)$$

en donde P_1 y P_2 sólo tienen coeficientes no negativos. Es claro que P_1 y P_2 determinan funciones recursivas de $\mathbb{N}^k$ en $\mathbb{N}$ y por lo tanto

$$|P(x_1, \ldots, x_k)| = |P_1(x_1, \ldots, x_k) - P_2(x_1, \ldots, x_k)|$$

$$P(x_1, \ldots, x_k)^2 = |P(x_1, \ldots, x_k)| \, |P(x_1, \ldots, x_k)|$$

son funciones recursivas.

Las razones para identificar las funciones calculables de $\boldsymbol{N}$ con las funciones recursivas se basan en el hecho de que todos los intentos para definir la noción de algoritmo (Post, Turing, Gödel, Church, Markov) han llevado siempre al mismo conjunto de funciones, las funciones recursivas totales o parciales arriba definidas. La afirmación de que toda función calculable es recursiva se conoce con el nombre de *Tesis de Chruch*.

Si introducimos el símbolo RC y usamos la notación $f(\vec{x}) = RC(g(\vec{x}), h(\vec{x},z,w))$ para indicar que f se ha definido recursivamente de g y h, de acuerdo con II, es claro que toda función recursiva estará representada por una sucesión finita que contiene los símbolos "O", "S", "RC", "P_i^n", "μn", variables y paréntesis; la cual indica cómo ha sido definida a partir de las funciones básicas. Por ejemplo:

$$x + y = RC(P_1^1(x),\ S(P_3^3(x, z, w))).$$

Esto significa que podremos codificar toda la información sobre f por medio de una sucesión finita de símbolos, y por lo tanto por un solo número, de la manera explicada en la sección 2. Esto muestra, en primer lugar, que sólo hay enumerables funciones recursivas. Además, para cada $k \geq 1$ se puede construir una función recursiva $V_k(x)$ tal que

$$V_k(n) = \begin{cases} 1 & \text{si } n \text{ codifica una función de } k \text{ variables} \\ 0 & \text{en caso contrario.} \end{cases}$$

En caso que n_f codifique la función $f(\vec{x})$ de las variables $\vec{x}=(x_1, \ldots, x_k)$, n_f contiene información suficiente para calcular $f(\vec{x})$ paso a paso. Podemos imaginar cada paso como la aplicación de una función básica a valores ya obtenidos, o la comprobación que un valor ya obtenido es igual a cero. En todo caso, si suponemos que la sucesión de valores calculados hasta el paso m se codifica como un solo número, es posible definir una función recursiva $B_k(y,w,\vec{x})$ tal que para cualquier función recursiva f, de k variables, número m y argumentos $\vec{x}$, tenemos que si n_f codifica a f entonces

$$B_k(n_f,m,\vec{x}) = \textit{código}$$

(Valores obtenidos hasta el paso m en el cálculo de $f(\vec{x})$) (Véase [1]).

Obviamente, la función f da un resultado para x si y solamente si el algoritmo se detiene en cierto paso k, es decir, deja de calcular nuevos valores y por tanto

$$B_k(n_f, k+1, \vec{x}) = B_k(n_f, k, \vec{x}).$$

Hay una función recursiva C que extrae el último valor de la sucesión codificada por un número. Por ejemplo, si $n = p_1^2 p_2^1 \ldots p_k^8 \cdot 2$ tenemos $C(n) = 7$. Entonces la función :

$$A(n_f, \vec{x}) = C(\mu k(|B(n_f, k+1, \vec{x}) - B(n_f, k, \vec{x})| = 0))$$

es tal que

$$A(n_f, \vec{x}) = f(\vec{x}) .$$

A es una función recursiva *universal* tal que dado el código n_f de f y algún dato $\vec{x}$, calcula $f(\vec{x})$, o queda indefinida si $f(\vec{x})$ no está definido.

4. Conjuntos recursivamente enumerables, Decidibles, Diofantinos.

Definición. Un conjunto S de $\mathbf{N}$ es *recursivamente enumerable* (R.E.) si $S = \phi$ o existe una función recursiva $f: \mathbf{N} \to S$ sobreyectiva.

Observe que f debe ser *total*, es decir, definida en todo $\mathbf{N}$. Podemos considerar a f como una forma de enumerar o generar los elementos de S

$$f(0), f(1), f(2), \ldots$$

No todo subconjunto de $\mathbf{N}$ es R. E., puesto que hay enumerables funciones recursivas y $\mathbf{N}$ tiene una cantidad no enumerable de subconjuntos (teorema de Cantor). Suponga que S se puede generar por medio de una función de más de una variable $f: \mathbf{N}^k \to S$, entonces S es R. E. . Para ver esto considere las siguientes funciones :

$$g_i: \mathbf{N} \to \mathbf{N} \quad , \quad i = 1,2,3, \ldots, k \quad ; \quad g_i(0) = 0$$

$g_i(n)$ = (exponente del primo p_i en la descomposición de n) $(n > 0)$. Se puede demostrar que estas funciones son recursivas. Por lo tanto $h: \mathbf{N} \to \mathbf{N}$ en donde $h(n) = f(g_1(n), \ldots, g_k(n))$ es recursiva, y enumera a S, puesto que la correspondencia

$$n \mapsto (g_1(n), \ldots, g_k(n))$$

es sobreyectiva.

Definición. Un subconjunto S de $\mathbf{N}$ se llama *diofántico* si es una proyección del conjunto solución de una ecuación diofantina, es decir existe un polinomio con coeficientes enteros $P(y_1, \ldots, y_k, x)$ tal que

$$S = \{ n \mid \exists y_1 \ldots \exists y_k (P(y_1, \ldots, y_k, n) = 0) \}$$

Observe que la existencia de soluciones enteras $y_1, \ldots, y_k$ para un polinomio $Q(y_1, \ldots, y_k)$ es equivalente a la existencia de soluciones *naturales* para uno de los polinomios $Q(\pm y_1, \ldots, \pm y_k)$ que resultan de anteponer signos negativos a algunas de las variables. Por tanto si definimos

$$Q^*(y_1, \ldots, y_k) = \prod_{(s_1, \ldots, s_k)} Q(s_1 y_1, \ldots, s_k y_k)$$

en donde $(s_1, \ldots, s_k)$ recorre todas las posibles sucesiones de $1, -1$, tendremos que

$$\exists y_1 \ldots \exists y_k (Q(y_1, \ldots, y_k) = 0) \iff \exists y_1 > \ldots \exists y_k > 0 \; (Q^*(y_1, \ldots, y_k) = 0)$$

Esto significa, en primer lugar, que el problema de Hilbert se puede reducir al de la existencia de soluciones naturales. Además, todo conjunto diofántico toma la forma

$$S = \{ n \mid \exists y_1 > 0 \ldots \exists y_k > 0 \; (P(y_1, \ldots, y_k, n) = 0) \}$$

Por lo tanto, *todo conjunto diofántico es* $R.E.$, pues es generado por la función

$$f(\vec{y},x) = x.(1 \dot{-} |P(\vec{y},x)|) + i_o.(1 \dot{-} (1 \dot{-} |P(\vec{y},x)|))$$

en donde i_o es algún elemento fijo de S. Una función $f(\vec{x})$ es diofántica si su gráfico es la proyección del conjunto solución de una ecuación diofántica; por las mismas razones anteriores tenemos :

$$m = f(n) \iff \exists y_i \geq 0 \ldots \exists y_k \geq 0 \ (P(y_1, \ldots, y_k, n, m) = 0)$$

Por lo tanto, f es recursiva, pues

$$f(n) = g_{k+1}(\mu s(|P(g_1(s), \ldots, g_k(s), n, g_{k+1}(s))| = 0))$$

Definición. Un subconjunto S de $\mathbb{N}$ es *decidible* si existe una función recursiva $f: S \to \mathbb{N}$ tal que $f(n) = 1$ si $n \in S$, $f(n) = 0$ si $n \notin S$. (Es decir, la función característica de S es recursiva).

Teorema A. Todo conjunto decidible es R. E.

Demostración. Si $S = \phi$ no hay nada que demostrar. Sea $S \neq \phi$ y sea $f: S \to \mathbb{N}$ su función característica. Como f es recursiva podemos definir una función recursiva g :

$$g(0) = \mu k(f(k) = 1)$$

$$g(n+1) = \begin{cases} n+1 & \text{si } f(n+1) = 1 \\ g(n) & \text{si } f(n+1) = 0 \end{cases} \qquad (1)$$

Entonces, $g(0) \in S$ y si $n \in S$, $g(n) = n$ de modo que g genera todos los elementos de S. Además, suponga, por hipótesis de inducción que $g(n) \in S$. Si $n+1 \in S$ entonces $g(n+1) = n+1 \in S$; si $n+1 \notin S$ entonces $g(n+1) = g(n) \in S$. Esto demuestra que g sólo genera elementos de S. Observe que (1) en la definición de g se puede expresar

$$g(n+1) = (n+1).f(n+1) + g(n).(1 \dot{-} f(n+1))$$

El recíproco de este resultado no es cierto.

Teorema B. Existen conjuntos R.E. que no son decidibles.

Demostración. Sean $V_1(x)$ y $B_1(x,w,y)$ las funciones recursivas definidas en la sección anterior. Dado un número natural n que codifica cierta función f llame $[n]$ a f. Defina

$S = \{n \mid n$ codifica una función de una variable y $[n]$ está definida en $n\}$.

Obviamente,

$$S = \{n \mid V_1(n) = 1 \wedge \exists k(\,|\, B_1(n,k+1,n) - B_1(n,k,n) \,|\, = 0\,)\}.$$

S es R. E. pues está generado por la siguiente función recursiva de dos variables

$$g(n,k) = \begin{cases} n \text{ si } V_1(n) = 1 \text{ y } |B_1(n,k+1,n) - B_1(n,k,n)| = 0 \\ i_o \text{ si } |B_1(n,k+1,n) - B_1(n,k,n)| \neq 0 \end{cases},$$

i_o es el código de cualquier función recursiva total de una variable[1] Pero S no es decidible. De lo contrario, la siguiente función sería recursiva

$$b(n) = \begin{cases} [n]\,(n) + 1 & \text{si } n \in S \\ 0 & \text{si } n \notin S \end{cases}$$

Pues,

$$b(n) = (A_1(n,n) + 1) \cdot f(n)$$

en donde $A_1(x,y)$ es la función universal definida en la sección anterior y $f(x)$ es la función característoca de S. Entonces $b = [\ell]$ para algún ℓ. Como b es total por definición, estaría definida en todas partes, en particular en ℓ, así que $\ell \in S$ y

---

(1) g es explícitamente definible de V_1 y B por la fórmula $g(n,k) = n \cdot V_1(n) \cdot (1 \dot{-} |\, B(n,k+1,n) - B(n,k,n) \,|\,) + i_o \cdot (1 \dot{-} (1 \dot{-} |\, B(n,k+1,n) - B(n,k,n) \,|\,))$

$[\ell\ ]\ (\ell) = b(1) = [\ell\ ]\ (\ell\ ) + 1$. Contradicción.

5. Definición Aritmética de las Funciones Recursivas.

Una fórmula *aritmética* es una formada por medio de los conectivos lógicos , $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, \exists, \forall$, a partir de fórmulas atómicas de la forma $\tau_1 = \tau_2$: en donde τ_1 y τ_2 son *términos* construidos de los símbolos de operación "+", "." aplicados a las variables y a las constantes 0, 1. Si identificamos el término $((1 + 1) + 1) + \cdots + 1)$ que contiene n "unos" con el numeral n, podemos suponer que el lenguaje contiene también símbolos para todos los naturales. Las relaciones elementales de las aritmética son expresables por fórmulas aritméti - cas si suponemos que las variables solo denotan números naturales :

$x \leq y \Leftrightarrow \exists z\,(x + z = y)$

$x < y \Leftrightarrow \exists z(x + z + 1 = y)$

"x es primo" $\Leftrightarrow \forall z\ \forall w(x = zw \Rightarrow z = 1 \vee z = x)$

$y \equiv x\ (m\acute{o}d\ z) \Leftrightarrow \exists k(y = kz + x \vee x = kz + y)$

Nótese que una ecuación polinomial

$$P(x_1, \ldots, x_k) = 0 \qquad (1)$$

es expresable en esta forma, aún si tiene coeficientes negativos, pues es equivalente a una ecuación :

$$P_1(x_1, \ldots, x_k) = P_2(x_1, \ldots, x_k) \qquad (2)$$

en donde P_1 y P_2 solo tienen coeficientes no negativos. Usaremos ecuaciones del tipo (1) como abreviaciones de ecuaciones del tipo (2).

Dado un subconjunto S de $\mathbb{N}$ se dice que es *aritmético* o *definible en la aritmética* si existe una fórmula aritmética $\varphi(x)$ tal que

$S = \{ n \mid \varphi(n) \}$ cuando las variables de φ varían en $\mathbb{N}$.

Por ejemplo, el conjunto de los números primos. Una función $f: \mathbb{N}^k \to \mathbb{N}$ se dice aritmética si su gráfica es definible en la aritmética, es decir,

$$m = f(n_1, \ldots, n_k) \iff \varphi(n_1, \ldots, n_k, m) \quad \text{vale en } \mathbb{N}$$

para alguna fórmula φ. Todo conjunto y función diofánticos son obviamente aritméticos. Las funciones $f(x,y) = x + y$, $g(x,y) = x.y$ se definen por las fórmulas "$x+y = z$", "$x.y = z$" respectivamente. ¿ Podemos decir que la función $f(x,y) = x^y$ es aritmética ? O la función $f(x) = x!$? Esta no es una pregunta trivial. La definición intuitiva.

$$m = n^k \iff m = \underbrace{n.n \ldots n}_{k \text{ veces}}$$

no da una fórmula válida para todo m,n,k, sino una fórmula diferente para cada valor de k, es decir, la familia de fórmulas "$y = x$", "$y = x.x$", "$y = x.x.x$" que definen las funciones de una variable

$$f_2(x) = x^2, \quad f_3(x) = x^3, \ldots$$

Otras funciones, aparentemente más complicadas tienen una definición aritmética muy clara ; por ejemplo $f(x)$ = "mínimo número primo mayor que x".

$$m = f(n) \iff \text{"}m \text{ es primo"} \wedge n < m \wedge \forall z(\text{"}z \text{ es primo"} \wedge n < z \Rightarrow m < z)$$

La función $f(x,y) = x^y$ es, sin embargo, recursiva y demostraremos que toda función recursiva es definible en la aritmética. La fórmula que define x^y en términos de "+", ".", y los conectivos lógicos se podría obtener analizando la prueba del siguiente teorema.

Teorema C. Toda función recursiva es definible por medio de una fórmula aritmética.

Demostración. Inducción en la complejidad de la definición de la función recursiva. La complejidad se mide por el número de veces que se han aplicado las reglas de formación I, II, III a las funciones básicas. Para las funciones básicas tenemos :

$$y = O(x) \iff x = x \ \wedge \ y = 0$$

$$y = S(x) \iff y = x + 1$$

$$y = P_i^n(x_1, \ldots, x_n) \iff x_1 = x_1 \wedge \ldots \wedge x_n = x_n \wedge y = x_i$$

Regla I.

Supongamos que el teorema es cierto para las funciones $h(z_1, \ldots, z_k)$, $g_1(\vec{x})$, $\ldots$, $g_k(\vec{x})$ y h es definida por composición

$$f(x) = h(g_1(\vec{x}), \ldots, g_k(\vec{x}))$$

Entonces,

$$y = f(\vec{x}) \iff \exists z_1 \ldots \exists z_k \begin{cases} y = h(z_1, \ldots, z_k) \\ \wedge \ z_1 = g_1(\vec{x}) \\ \vdots \\ \wedge \ z_k = g_k(\vec{x}) \end{cases}$$

substituyendo "$y = h(z_1, \ldots, z_k)$", "$z_1 = g_1(\vec{x})$" etc., por sus definiciones aritméticas, obtenemos una definición aritmética para $y = f(\vec{x})$

Regla III.

Sea $f(\vec{x})$ definida de la función $h(\vec{x}, y)$ por

$$f(\vec{x}) = \mu y \, (h(\vec{x}, y) = 0)$$

en donde "$z = h(\vec{x}, y)$" es aritmética.

Entonces $y = f(\vec{x}) \iff h(\vec{x}, y) = 0 \ \wedge \forall z < y \, (\neg h(\vec{x}, z) = 0))$.

Hemos dejado la regla de recurrencia para el final porque esta es la que ofrece

mayor dificultad.

Regla II.

Sea $f(\vec{x}, y)$ definida de la función $g(\vec{x})$ y la iterada $h(\vec{x}, z, w)$ por

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

en donde $g(\vec{x})$ y $f(\vec{x}, z, w)$ son aritméticas. Obviamente tenemos :

$$f(\vec{x}, n) = m \iff \exists a_1 \exists a_2 \cdots \exists a_{n+1} \begin{cases} a_1 = g(\vec{x}) \\ \wedge\, a_2 = h(\vec{x}, 1, a_1) \\ \wedge\, a_3 = h(\vec{x}, 2, a_2) \\ \vdots \\ \wedge\, a_{n+1} = h(\vec{x}, n, a_n) \\ \wedge\, m = a_{n+1} \end{cases} \qquad (A)$$

Desgraciadamente, esta no es *una* fórmula que pueda definir h, pues la fórmula depende del argumento n. En particular el número de cuantificadores depende de n. ¿ Será posible encontrar *una* fórmula equivalente a (A) en donde aparezca n como valor de una variable y no afecte la estructura de la fórmula ? La respuesta es que sí es posible.

Gödel descubrió una manera de generar todas las sucesiones $a_0, \ldots, a_n$, de cualquier longitud, por medio de una sola función representable de tres variables.

Definición.

$G(a,b,k)$ = mínima solución no negativa de la congruencia $y \equiv a \ mod\ (1 + k\,b)$

Obviamente, $G(a,b,k)$ es representable :

$$G(a,b,k) = y \iff [\, y \equiv a \ mód\ (1 + k\,b)\,] \wedge [\, 0 \leq a < 1 + k\,b\,]$$

El siguiente lema se basa en el teorema chino del residuo [3] .

Lema D. Dados números no negativos $a_1, \ldots, a_n$, existen números (no negativos) a, b tales que :

$$G(a,b,1) = a_1$$

$$G(a,b,2) = a_2$$

$$\vdots$$

$$G(a,b,n) = a_n$$

Demostración. Dados $a_1, \ldots, a_n$, defina $b = n!\,(a_1+1)(a_2+1)\ldots(a_n+1)$. Entonces los números $1+b,\ 1+2b, \ldots,\ 1+nb$ son relativamente primos por pares . Pues si es un número primo tal que $p \mid (1+kb)$, $p \mid (1+k'b)$ con $k' < k \leq n$, entonces $p \mid ((1+kb)-(1+k'b)) = (k-k')b$, por lo tanto $p \mid b$ ó $p \mid (k-k')$. Pero $k-k' < n$, de manera que el caso $p \mid (k-k')$ implica $p \mid n! \mid b$. En todo caso $p \mid b$, que dá una contradicción, pues $p \mid (1+kb)$ y por lo tanto tendríamos $p \mid ((1+kb)-kb) = 1$. Por el teorema chino del residuo existe una solución a al sistema de congruencias

$$x \equiv a_1 \quad mód\ (1+b)$$

$$x \equiv a_2 \quad mód\ (1+2b)$$

$$\vdots$$

$$x \equiv a_n \quad mód\ (1+nb)$$

puesto que los módulos son relativamente primos por pares. Además en la congruencia

$$a \equiv a_k \quad mód\ (1+kb)$$

es obvio que a_k constituye la mínima solución no negativa de la congruencia $a \equiv x \ mód\ (1+kb)$ pues, por escogencia de b, $0 \leq a_k < (a_1+1)(a_2+1)\ldots(a_n+1) \leq b$.∎

Ahora podemos completar la demostración del teorema. Como consecuencia del lema, la expresión (A) es equivalente a :

$$\exists a \exists b \begin{cases} G(a,b,1) = f(\vec{x}) \\ \forall k \leq n \ [k \geq 1 \Rightarrow G(a,b,k+1) = g(\vec{x},k,G(a,b,k))] \\ m = G(a,b,n+1) \end{cases}$$

Como G es representable y f,g son representables por hipótesis, la expresión arriba indicada es equivalente a una fórmula aritmética : $\exists a \exists b \varphi(a,b,\vec{x},n,m)$ y tenemos :

$$b(\vec{x},n) = m \iff \exists a \exists b \varphi(a,b,\vec{x},n,m)$$

para cualquier $\vec{x},n,m$. ∎

Corolario E. Todo conjunto R.E. es definible.

Demostración. Sea $f : \mathbb{N} \to \mathbb{N}$ una función que enumera recursivamente a S. Entonces,

$$n \in S \iff \exists x \ (f(x) = n)$$

pero $f(x) = n$ es equivalente a una fórmula $\psi(x,n)$, de modo que

$$S = \{ n \mid \exists x \psi(x,n) \}$$

Esta representación se puede refinar si hacemos uso del siguiente resultado :

Teorema F. Toda fórmula aritmética $\varphi(x_1,\ldots,x_k)$ es equivalente a una de la forma :

$$Qy_1 Qy_2 \cdots Qy_s (P(y_1,\ldots,y_s,x_1,\ldots,x_k) = 0) \qquad (*)$$

en donde Q es uno de los cuantificadores $\exists$ ó $\forall$, y $P(y_1,\ldots,y_s,x_1,\ldots,x_k)$

es un polinomio.

Demostración. Hacemos la demostración por inducción en la complejidad de las fórmulas, es decir en el número M de conectivos lógicos. Supongamos que estos se reducen a $\wedge$, $\neg$, $\exists$. (Qy denota una lista arbitraria de cuantificadores.

$M=0$, fórmulas atómicas. Deben ser de la forma $x=y$, $x+y=z$, $x.y=z$, en donde x,y,z son variables o numerales; y se pueden expresar como $x-y=0$, $x+y-z=0$ y $xy-z=0$ respectivamente.

Supongamos la hipótesis cierta para complejidad $N < M$. Sea $\varphi(\vec{x})$ de complejidad M. Entonces φ es de la forma $\delta \wedge \rho$, $\neg \delta$, o $\exists w \ \delta(m,x)$ en donde δ , ρ tienen complejidad menor que M. Por hipótesis de inducción tenemos

$$\delta(\vec{x}) \Longleftrightarrow \vec{Q}y \ (P(\vec{y},\vec{x}) = 0)$$

$$\rho(\vec{x}) \Longleftrightarrow \vec{Q}z \ (R(\vec{z},\vec{x}) = 0)$$

Pero $A=0 \wedge B=0 \Longleftrightarrow A^2 + B^2 = 0$ en los enteros, y $\vec{Q}y \ \varphi(\vec{y}) \wedge \vec{Q}z \ \psi(\vec{z})$ $\Longleftrightarrow \vec{Q}y \ \vec{Q}z(\varphi(\vec{y}) \wedge \psi(\vec{z}))$ vale siempre que las listas $\vec{y},\vec{z}$ sean disyuntas ; entonces

$$\delta(\vec{x}) \wedge \rho(\vec{x}) \Longleftrightarrow \vec{Q}y\vec{Q}z \ (P(\vec{y},\vec{x}) = 0 \wedge R(\vec{z},\vec{x}) = 0) \Longleftrightarrow \vec{Q}y \ \vec{Q}z(P(\vec{y},\vec{x})^2 + R(\vec{z},\vec{y})^2 = 0$$

$$\neg \delta(x) \Longleftrightarrow \neg \vec{Q}y \ (P(\vec{y},\vec{x}) = 0) \Longleftrightarrow \vec{Q^*}y \ (P(\vec{y},\vec{x}) \neq 0)$$

en donde Q^* resulta de intercambiar "$\exists$" y "$\forall$". Pero $A \neq 0 \Longleftrightarrow A^2 > 0$ $\Longleftrightarrow \exists z(A^2 = z+1)$. Entonces,

$$\neg \delta(\vec{x}) \Longleftrightarrow \vec{Q^*}y \ \exists z \ (P^2(\vec{y},\vec{x}) - z - 1 = 0)$$

Finalmente, si $\delta(w,\vec{x}) \Longleftrightarrow \vec{Q}y \ (P(\vec{y},w,\vec{x}) = 0)$,

$$\exists w \ \delta(w,\vec{x}) \Longleftrightarrow \exists w \ \vec{Q}y \ (P(\vec{y},w,\vec{x}) = 0) \ . \blacksquare$$

Del teorema anterior obtenemos que si S es recursivamente enumerable, S

es definible por una fórmula de la forma (*). Sin embargo, si se analiza la prueba de la definibilidad aritmética de las funciones recursivas se descubrirá que todo conjunto R.E. se puede definir por una fórmula de la forma (*) en donde todos los cuantificadores universales son *acotados*, es decir, de la forma $\forall x \leq y$. M. Davis demostró que el número de estos cuantificadores se podía reducir a uno solo.

Teorema G. Todo conjunto R. E. es definible por medio de una fórmula $\varphi(x)$ de la forma

$$\exists u \forall y_1 \leq u \exists y_2 \exists y_3 \cdots \exists y_s (P(u, y_1, \ldots, y_s, x) = 0 \quad \text{(Vea [8])}.$$

De manera que a cada conjunto R.E., S, se puede asociar un polinomio $P(u, y_1, \ldots, y_s, x)$ tal que $n \in S$ si y sólo si existe un número M tal que el sistema de M ecuaciones :

$$\begin{aligned} &P(M, 1, y_2, \ldots, y_s, n) = 0 \\ &P(M, 2, y_2, \ldots, y_s, n) = 0 \\ &\quad\vdots \\ &P(M, M, y_2, \ldots, y_s, n) = 0 \end{aligned}$$

tiene una solución $(y_2, \ldots, y_s)$. Como el número M de ecuaciones depende de n, no podemos reducir el sistema a una sola ecuación tomando las sumas de los cuadrados de los polinomios, pues nos daría una ecuación diferente para cada n. ¿Es posible eliminar el cuantificador universal de la definición de S? Es decir, es todo conjunto R.E. diofántico? M. Davis y J. Robinson llegaron a demostrar el siguiente resultado.

Teorema H. Basta que exista una función recursiva de crecimiento exponencial que sea diofántica para que toda función recursiva y por tanto todo conjunto R.E., sea diofántico (Vea [8]).

$f(x)$ crece exponencialmente si existen L, M, $L > 1$ tales que para toda n $L^n \leq f(n) \leq M^n$.

6. Números de Fibonacci, Teorema de Matijasevich.

Fibonacci, o Leonardo de Pisa (siglo XIII), introdujo la sucesión de números

$$0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

cada número resulta de sumar los dos anteriores, es decir

$$F(0) = 0, \quad F(1) = 1, \quad F(n+1) = F(n) + F(n-1).$$

Es posible dar una fórmula analítica para esta sucesión. Considere la ecuación

$$x_{n+1} = x_n + x_{n-1} \qquad (1)$$

y suponga que $x_n = r^n$. Entonces

$$r^{n+1} = r^n + r^{n-1}$$

que es equivalente a $r^2 = r + 1$. Esta es una ecuación de segundo grado cuyas soluciones son

$$a = \frac{1 + \sqrt{5}}{2} \qquad b = \frac{1 - \sqrt{5}}{2}$$

Por lo tanto $x_n = a^n$, $x_n = b^n$ son soluciones de (1). Es claro, debido a la linealidad de la ecuación, que cualquier combinación

$$F(n) = \alpha a^n + \beta b^n$$

también es solución. Basta, pues, determinar α, β de manera que $F(0) = 0$ $F(1) = 1$. Esto da

$$\alpha + \beta = 0 \quad ; \quad \alpha a + \beta b = 1$$

cuya solución es

$$\alpha = \frac{1}{a-b} = \frac{1}{\sqrt{5}} \quad , \quad \beta = -\alpha = -\frac{1}{\sqrt{5}}$$

Tenemos, entonces

$$F(n) = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right\}$$

En particular :

$$F(2n) \leq \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^{2n} = \frac{1}{\sqrt{5}} \left(\frac{3+\sqrt{5}}{2}\right)^n \leq 3^n$$

Además, la sucesión es creciente y

$$F(2n) \leq F(2n-1) = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2}\right)^{2n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{2n-1} \right\} = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2}\right)^{2n-1} + \left(\frac{\sqrt{5}-1}{2}\right)^{2n-1} \right\} \geq \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^{2n-1}$$

Pero

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^{2n-1} = \frac{1}{\sqrt{5}} \frac{2}{1+\sqrt{5}} \left(\frac{3+\sqrt{5}}{2}\right)^n \geq \frac{1}{4} 2^n \geq (\sqrt{2})^n \qquad \text{para } n \geq 4 .$$

Entonces,

$$(\sqrt{2})^n \leq F(2n) \leq 3^n \qquad n \geq 4$$

y la función $H(n) = F(2n)$ es de crecimiento exponencial. La función $H(n) = F(2n)$ es obviamente recursiva. El trabajo de Matijasevich consistió en demostrar que esta función es diofántica [5]. El demostró que existe un polinomio (de 16 variables) tal que para todo n, m ,

$$m = H(n) \iff \exists x_1 \ldots \exists x_{14} \left(P(x_1, \ldots, x_{14}, n, m) = 0\right)$$

el encontrar este polinomio es un trabajo de teoría de números. Combinando esto con el resultado de Davis y Robinson, tenemos :

Teorema 1. (Matijasevich). Todo conjunto R.E. es diofantino.

Corolario J. El décimo problema de Hilbert es insoluble.

Demostración. Sea S un conjunto R.E. que no sea decidible. Entonces existe un polinomio $P(\vec{y},x)$ tal que

$$S = \{ n \mid \exists \vec{y} \, (P(\vec{y},n) = 0 \}$$

Supongamos que hay un algoritmo $\mathcal{H}$ para el problema de Hilbert. Entonces podemos construir el siguiente algoritmo de los números naturales en $\{0,1\}$.

A. Dado n como dato, halle la forma canónica del polinomio $Q(\vec{y}) = P(\vec{y},n)$ (esto se reduce a realizar ciertas multiplicaciones).

B. Aplique $\mathcal{H}$ a la ecuación "$Q(y) = 0$" y de como resultado el valor de $\mathcal{H}$.

Obviamente,

$$n \in S \iff \exists \vec{y} \, P(\vec{y},n) \iff \exists \vec{y} \, Q(\vec{y}) \iff \mathcal{H}(\text{"}Q(\vec{y}) = 0\text{"}) = 1$$

de manera que tendríamos un algoritmo de decisión para S. Contradicción. ∎

En la demostración anterior hemos obtenido algo más fuerte. No hay un algoritmo para decidir la solubilidad de cada una de la familia infinita de ecuaciones :

$$P(y_1, \ldots, y_k, 0) = 0$$
$$P(y_1, \ldots, y_k, 1) = 0$$
$$P(y_1, \ldots, y_k, 2) = 0$$
$$\vdots$$

Puesto que el grado y número de variables de $P(\vec{y},x)$ está determinado, por ejemplo se puede dar un conjunto R.E. no decidible y una ecuación que lo defina de grado 50 y con 50 variables; obtenemos que no existe un algoritmo para decidir si una ecuación diofantina de grado 50 con 50 variables tiene soluciones enteras.

Otra consecuencia muy interesante debida a H. Putnam es la siguiente.

Corolario K. Todo conjunto R.E. de números naturales coincide con los valores no negativos de cierta función polinomial $Q(x_1,\ldots,x_t)$, para $x_1,\ldots,x_t$ naturales.

Demostración. Sea $S = \{n \mid \exists y_1,\ldots,y_k\,(P(y_1,\ldots,y_k,n) = 0)\}$.

Tome como $Q(y_1,\ldots,y_k,x)$ el polinomio

$$(x+1)\,(1-P^2(y_1,\ldots,y_k,x)) - 1$$

Sea $x \geq 0$

$x \in S \Rightarrow P^2(y_1,\ldots,y_k,x) = 0$ para algunos $y_1,\ldots,y_k \geq 0$

$\Rightarrow Q(y_1,\ldots,y_k,x) = (x+1) - 1 = x$.

Si $x = Q(y_1,\ldots,y_k,z) = (z+1)(1-P^2(y_1,\ldots,y_k,z) - 1$ con $y_1,\ldots,y_k,z \geq 0$ tenemos $P^2(y_1,\ldots,y_k,z) = 0$, de lo contrario la expresión sería negativa. Por tanto, $x = (z+1) - 1 = z$ y tenemos $P(y_1,\ldots,y_k,x) = 0$ que implica $x \in S$.

En particular, debe existir un polinomio que genera los números primos. Se han construido polinomios con esa propiedad de 26 variables [4]. En contraste, en teoría elemental de los números se demuestra que no puede existir un polinomio de una variable que genere solamente números primos. Finalmente podemos dar una versión del famoso teorema de incompletitud de Godel.

Teorema L. No existe un sistema axiomático consistente del cual se pue-

dan deducir todas las verdades de la aritmética.

Demostración. Supongamos que existe un tal sistema A y usemos la notación $A \vdash \varphi$ para indicar que la fórmula φ de la aritmética se deduce de A. El concepto de deducción formal es algorítmico, en el sentido de que dada una sucesión de fórmulas es posible decidir si es o no es una deducción correcta. Es posible entonces dar un algoritmo que enumere todas las deducciones correctas. Basta enumerar todas las sucesiones y tomar aquellas que son deducciones correctas. Ahora, dada una ecuación diofantina "$P(x) = 0$" sabemos que se puede poner en la forma $P_1(\vec{x}) = P_2(\vec{x})$ en donde "$P_1(\vec{x}) = P_2(\vec{x})$" pertenece al lenguaje de la aritmética. Considere la fórmula

$$\varphi : \exists \vec{x} \, (P_1(\vec{x}) = P_2(\vec{x}))$$

Esta debe ser verdadera o falsa. Si φ es verdadera, $A \vdash \varphi$ por hipótesis. Si φ es falsa entonces $\neg \varphi$ es verdadera y $A \vdash \neg \varphi$. Además de A no se puede deducir φ y $\neg \varphi$ pues A es consistente. El siguiente algoritmo nos daría un algoritmo de decisión para el problema de Hilbert. Enumere todas las deducciones correctas y examine la última fórmula; detenga el proceso en el momento en que aparezcan φ ó $\neg \varphi$. El proceso termina pues una de las dos fórmulas es deducible; debe haber pues una deducción que termina en la primera o en la segunda. Si es φ la que aparece la ecuación $P(x) = 0$ es soluble. Si es $\neg \varphi$, la ecuación es insoluble

Bibliografía

[1] Hans Hermes, *Enumerability, Dedidability, Computability*. Springer-Verlag, Berlín, 1969.

[2] Martin Davis, *Hilbert's tenth problem is Unsolvable*. American Mathematical Monthly, Vol. 80, Nº 3, (1973).

[3] Burton W. Jones, *Teoría de los Números*, Ed. F. Trillas, México, 1969.

[4] James P. Jones, *Diophantine representation of the set of prime numbers*. Notices of the A.M.S. , Vol. 22 Nº 2 (1975).

[5] Yu V. Matijasevich, *Enumerable sets are diophantine*. Soviet Mathematics , Vol. II, Nº 2 (1970).

[6] ———————, *Diophantine Representation of recurssively enumerable predicates. Proceedings of the second Scandinavian Logic Symposium*, Nort Holland, Amsterdam, 1971.

[7] Hilary Putnam. *An unsolvable problem in number theory*. The Journal of Symbolic Logic, Vol. 25, Nº 3 (1960).

[8] Julia Robinson, *Diophantine Decision problems*. Studies in Mathematics , Vol. 6, The Mathematical Association of America, 1969.

[9] William J. Le Veque, *Teoría Elemental de los números* (traducción al español), Herrero Hermanos, S.A., México, 1968.