.UBAfiuba
FACULTAD DE INGENIERÍA

# Tesis de Church-Turing

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

# Los 23 problemas de Hilbert

## David Hilbert

Presentó 23 problemas matemáticos a resolver en el próximo siglo

En una conferencia del congreso internacional de matemáticos en Paris en el año 1900.

## El 10mo problema

Encontrar un algoritmo para determinar si una ecuación polinómica diofántica (con 2 o más incógnitas) con coeficientes enteros, tiene una solución entera.

solución entera: Si la evaluación del polinomio con una asignación de sus variables con valores enteros da como resultado cero (raíz entera)

David Hilbert: Matemático alemán (1862 – 1943)

# 10mo problema

## Hilbert

Presenta el problema pensando que se iba a poder resolver.

## No se planteaba

La imposibilidad del cálculo

(similar era el pensamiento hegemónico contemporáneo)

## Sería cuestión

de encontrar el algoritmo que indique paso a paso como hallar la solución

# Ejemplo

**La siguiente ecuación polinómica diofántica**

$6x^3yz^2 + 3xy^2 - x^3 - 10$

**Corresponde a un polinomio**

De 4 términos con 3 variables

**Buscamos la existencia de una raíz entera**

una asignación de sus variables con valores enteros que da como resultado cero

**Expresamos entonces:**

$6x^3yz^2 + 3xy^2 - x^3 - 10 = 0$

**Y podemos ver que la siguiente asignación responde afirmativamente**

$x = 5, y = 3$ y $z = 0 \rightarrow 6x^3yz^2 + 3xy^2 - x^3 - 10 = 0 + 3*5*9 - 125 - 10 = 135 - 125 - 10 = 0$

# Sobre el algoritmo

## Hilbert no utilizó el término algoritmo

Sino, “un proceso según el cual se puede determinar en un número finito de operaciones”

## El concepto de algoritmo es antiguo

Utilizado desde la antigüedad para procesos matemáticos (griegos, egipcios, babilonios)

## Pero no había sido definido formalmente aún

Para responder afirmativamente bastaba con encontrar el procedimiento

Para responder negativamente, se requería determinar los límites de los algoritmos

# Entscheidungsproblem

## El problema de decisión

Fue un desafio lanzado por David Hilbert y Wilhelm Ackermann en 1928

## Implica

encontrar un algoritmo general que decidiese si una fórmula del cálculo de primer orden es un teorema

## En este desafío es en el que trabajo Turing

Y su resultado fue la publicacion “ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM” del año 1936

# Entscheidungsproblem (cont.)

**Turing propone**

Su maquina automática (Maquina de Turing) como herramienta de cálculo.

**Paralelamente (y aventajándolo por unos meses),**

Alonzo Church desarrolla su cálculo lambda (λ-calculus) para el mismo motivo.

**Turing demuestra que su**

Máquina y el λ-calculus son equivalentes.

Alonzo Church (1903-1995): Matemático Estadounidense

# Tesis de Church Turing

## Con el trabajo de Church y turing

Se dió un marco para definir que es un algoritmo

## Coloquialmente:

Todo algoritmo existe si es equivalente a una maquina de Turing

## Todos los modelos matemáticos de computo posibles

Tienen igual o menor poder computacional que las máquinas de turing.

(con posibles queremos decir construibles en el mundo real, por ejemplo: la imposibilidad de realizar una cantidad de operaciones infinita en un tiempo finito)

## Importante! Es una tesis, por lo tanto no está probado

Pero es aceptado casi universalmente

# 10mo problema: Un problema más sencillo

## Si restringimos el 10mo problema

A determinar si un polinomio de 1 variable tiene raíces enteras

## Expresamos el polinomio como

$f(x) = c_1x^k + c_2x^{k-1}+\ldots+c_{k+1}$

## Podemos ir probando valores de x

0, 1, -1, 2, -2, 3, -3,...

# Resolución

## Podemos construir una TM

para computar el problema

## Definimos el lenguaje para nuestra TM

P={p/ p polinomio con variable x con una raiz entera}

## Nuestra TM M será

“Ante el input (p): donde p polinomio con variable x

Evaluar p estableciendo x según la secuencia 0, 1, −1, 2, −2, 3, -3 …

Si en algun momento p se evalua en 0 , aceptar”

# Análisis de la TM

**El algoritmo encontrará eventualmente si el polinómio tiene una raíz entera.**

Pero si no tiene, loopeará eternamente

**Por lo tanto**

la TM anterior reconoce el lenguaje P

**¿Se puede evitar el posible loopeo?**

Si! podemos acotar el rango donde se puede encontrar la raiz entera.

# Rango de la raíz entera

## Sea

$$f(x_0) = c_1 x_0^k + c_2 x_0^{k-1} + \ldots + c_{k+1} = 0$$

## Podemos operar algebraicamente

$$c_1 x_0^k + c_2 x_0^{k-1} + \ldots + c_{k+1} = 0 \Rightarrow c_1 x_0^k = -\left(c_2 x_0^{k-1} + \ldots + c_{k+1}\right) \Rightarrow$$

$$|c_1 x_0^k| = |-\left(c_2 x_0^{k-1} + \ldots + c_{k+1}\right)| \Rightarrow |c_1||x_0^k| = |c_2 x_0^{k-1} + \ldots + c_{k+1}| \Rightarrow$$

$$|c_1||x_0^k| \leq |c_2 x_0^{k-1}| + \ldots + |c_{k+1}|$$

# Rango de la raíz entera (cont.)

## Sea $C_{max}$ al coeficiente con mayor numero absoluto

$$|c_1||x_0^k| \leq |c_{max}\, x_0^{k-1}| + \ldots + |c_{max}| \Rightarrow |c_1||x_0^k| \leq |c_{max}| * (|x_0^{k-1}| + \ldots + 1) \Rightarrow$$

$$|c_1||x_0^k| \leq |c_{max}| * (k|x_0^{k-1}|) \Rightarrow |x_0| \leq \frac{|c_{max}| * k}{|c_1|}$$

## Entonces

puedo restringir la búsqueda de la raíz entera entre valores $\pm \frac{|c_{max}| * k}{|c_1|}$

Y mi TM se convierte en un decididor

## Como colorario

P={p/ p polinomio con variable x con una raiz entera} es TURING DECIDIBLE

# Regresando a la ecuación polinómica diofántica

## Podemos construir una TM

para computar el problema

## Definimos el lenguaje para nuestra TM

D={p/ p polinomio con 2 o más variables con una raiz entera}

## Podemos ir probando los valores

(0,0,…,0),(1,0,…,0),(1,1,…,0),…,(1,1,…,1),…,(0,0,…,-1),…. todas las combinaciones posibles

## Por lo tanto

D={p/ p polinomio con 2 o más variables con una raiz entera} es TURING RECONOCIBLE

# Regresando... (cont.)

## Se puede crear una TM

para el problema que sea decididor?

## En 1971

Yuri Matijasevich demostró que no es posible

(utilizó el andamiaje creado por Church y Turing)

## Por lo tanto

D={p/ p polinomio con 2 o más variables con una raiz entera} NO ES TURING DECIDIBLE

.UBAfiuba

FACULTAD DE INGENIERÍA

Presentación realizada en Julio de 2020