

Notas de computabilidad.

En un congreso internacional de matemáticas el matemático *Hilbert* enunció 23 famosos problemas no resueltos hasta ese entonces.

Entre ellos se contaba con la pregunta que hasta ese momento sólo había sido formulada de manera filosófica de la completez de las matemáticas.

Esto es, dado un conjunto de axiomas, con este y las técnicas de toda la matemática conocida hasta entonces, era posible demostrar cualquier nuevo resultado?

Algún tiempo después *Goedel* respondió a la pregunta de manera negativa, demostrando formalmente que un sistema axiomático si es consistente no puede ser completo y su consistencia no puede ser demostrada dentro del sistema.

Esto vino a revolucionar la forma de pensar a las matemáticas, base de todas las así llamadas "ciencias puras", ya que implicaba que habrían resultados ciertos pero que no se podrían demostrar.

Siguiendo estos resultados, *Alan Turing* trabajó en la idea de la *máquina automática*, actualmente llamada **máquina de Turing**.

Esta consiste en una *cinta* que se extiende de manera infinita hacia la izquierda y hacia la derecha, dividida transversalmente en *celdas* o *cuadros*, en cada celda se puede escribir un solo *símbolo* dentro de un *alfabeto* finito de símbolos **S**.

La cinta es recorrida por una cabeza que *lee* y puede *modificar* el símbolo en cada celda, basándose en una *tabla finita de reglas o tuplas* divididas en una serie de *estados* en los que la máquina puede *caer*.

La cabeza puede recibir las instrucciones **I** o **D**, para *moverse* exactamente una casilla a la izquierda o a la derecha respectivamente. Una variación es usar **N**, para *no moverse*.

Cada estado describe las posibles *acciones* a realizar por la cabeza, de encontrarse en la celda actual con alguno de los símbolos del alfabeto.

Dentro del alfabeto se encuentra el símbolo **b**, llamado *blanco* que es con el que inicialmente se encuentran pobladas todas las celdas de la cinta (a veces en vez de **b** usaremos **0**).

Dentro de los estados, hay uno que es el *primero* y otro especial llamado **H** (del inglés *halt* = *detenerse*)

El símbolo blanco es el único que se permite repetirlo de manera infinita en la cinta.

La cinta puede comenzar vacía (sólo blancos), o con cierta *información inicial* y la cabeza se sitúa por lo general en el elemento no blanco de más a la izquierda, de existir.

Las tuplas se dividen en cinco campos que originalmente representan en orden: *Estado actual*, *Valor leído*, *Valor a escribir*, *Movimiento* y *Estado siguiente*.

Un ejemplo (*el castor ocupado*) es el siguiente: la cinta empieza vacía (todas las celdas en blanco), el alfabeto es $\{0, 1\}$; $(b=0)$ y las tuplas son

$(A, 0, 1, D, B)$; $(A, 1, 1, I, C)$; $(B, 0, 1, I, A)$; $(B, 1, 1, D, B)$; $(C, 0, 1, I, B)$; $(C, 1, 1, D, H)$.

Esta construye un conjunto (finito) de números uno, y luego de detiene.

Ejercicios, deducir que hacen las tuplas anteriores; escribir máquinas de Turing que escriban n números consecutivos, para $n=1, 2, \dots, 10$.

Pregunta: puede una máquina de Turing almacenar información?

Algo que observó Turing es que en la cinta de su máquina podía representar las reglas mismas, codificando en dicha cinta una máquina de Turing completa, con lo cual inventó el concepto de programa almacenado. (esto sería lo que ahora llamamos *programa* o *app*)

Con esto, ideó el concepto de **máquina universal de Turing**, las cuales son máquinas normales de Turing pero cuya entrada (programa almacenado) es otra máquina de Turing. Este es el principio de las computadoras actuales.

Alan Turing inventó su máquina para demostrar que existen problemas que *no son computables*, es decir, que no tienen una solución *sistemática* o *algorítmica* diríamos actualmente.

Esto lo logró con ayuda de las máquinas universales, probando que el problema de demostrar que **siempre se puede detectar que una máquina de Turing se detiene**, es no computable. (problema **HALT**)

Para ello definió $H(M, I)$ como la máquina de Turing universal que tenía las entradas M e I , que devolvía *si* , si la máquina de Turing M ejecutada con la entrada I terminaba, y devolvía *no* , si M no terminaba.

Luego definió otra máquina universal de Turing $H'(M)$ cuya única entrada era la máquina M y su resultado era *si* en caso de que $H(M, M)$ devolviera *no* , y devolvía un ciclo infinito (no terminaba) en caso de que $H(M, M)$ devolviera *si* .

Tomó entonces $H'(H')$, que es también una máquina de Turing, pero su misma existencia es una contradicción, pues no puede dar ningún valor. Esto demuestra que la hipótesis de que existía una máquina de Turing H , no es cierta.

A continuación veremos algunos ejemplos de "programas" que se pueden generar para la máquina de Turing.

En los siguientes ejemplos la cabeza empieza en el 1 más a la izquierda, y el alfabeto es {0,1} ...

Ej. Rebota entre dos posiciones (ciclo infinito):

Entrada=(1)

Tuplas: A00IA, A11DA

Ej. Recorre la serie de unos de la entrada de izq a der y de der a izq

Entrada=(1111111)

Tuplas: A00IB, A11DA, B00DA, B11IB

Ej. Escribir 10101010101...

Entrada=(0) "cinta vacía"

Tuplas: A01DB, B00DA

Ej. Escribir 110110110...

Entrada=(0)

Tuplas: