

Sobre las conferencias de HC

CARLOS RAFAEL ORTEGA LEZCANO

ERIC MARTIN GARCIA

Grupo C511

CONFERENCIA 1

Definiremos la noción intuitiva de algoritmo como un conjunto de acciones finitas aplicadas finitamente a la información de un problema transformando esta en una solución para el problema. Un ejemplo es la definición de función matemática donde f transforma un valor $x \in A$ en un valor que pertenece a B mediante $f(x)$.

Problema de Decisión: Un problema se considera de decisión si existe un algoritmo que de manera excluyente suministra una salida binaria para cada entrada del problema, esta salida debe ser *SI* o *NO*.

Algunos ejemplos de problemas de decisión serían, dada una fórmula de la lógica proposicional determinar si es verdadera o falsa, dado el problema del viajante vendedor saber si existe un recorrido de viaje que genere ganancias.

Por tanto podemos decir que las definiciones de decidible y computable son equivalentes, además la definición intuitiva de computable no incluye la complejidad del algoritmo, por ejemplo los dos ejemplos anteriores son computables pero según su complejidad son NP-Complejos

Dado un problema debe distinguirse entre clase del problema, las subclases de este problema y ejemplos específicos de la clase, por tanto a la hora de definir algoritmos podemos encontrar algunos para solucionar cualquier subclase de un problema o ejemplo de la clase, otros que solo resuelven un conjunto de subclases del problema y es posible encontrar problemas para los cuales no podemos escribir un algoritmo que resuelva todas las subclases del problema.

Por lo anterior es posible pensar si existen problemas los cuales no son decidibles, el matemático David Hilbert, presentó una lista de problemas matemáticos aun sin resolver por entonces. Entre estos se encontraba el LPO, el cual es un problema semidecidible ya que existe un procedimiento que si la formula es verdadera, para dando una respuesta afirmativa y si la fórmula es falsa, puede que no pare dando una respuesta negativa, este no es muy efectivo ya que no se puede realizar satisfactoriamente la partición del conjunto de entrada. Por tanto esto nos lleva a pensar que el LPO no es decidible, en 1936 Alan Turing demostró esto planteando una definición rigurosa de computabilidad, ya que si se quiere demostrar que un problema no es decidible, se tiene que demostrar que no existe algoritmo que de modo general para cualquier ejemplo del problema termine dando una respuesta afirmativa o negativa.

Para su demostración Turing planteó una definición rigurosa de algoritmo basada en una máquina abstracta que definía el concepto de computabilidad, convertida en una fase posterior de la demostración en una máquina que empleaba otras máquinas, terminando en lo que hoy conocemos como las computadoras físicas.

La Tesis de Church-Turing: Alonzo Church, tutor de la tesis de doctorado de Alan Turing propuso el Cálculo Lambda como un formalismo para el cálculo efectivo de funciones, este daría con posterioridad al surgimiento del primer lenguaje de programación, LISP. La definición de la tesis planteada por ambos es: *Una función es Turing-computable si existe una Máquina de*

Turing que realiza el cálculo de efectivo de la función, por tanto "Toda función efectivamente calculable es Turing-computable"

Además del CL y la MT se han ofrecido otras definiciones todas equivalentes:

1. Las Funciones recursivas de Herbrand- Gödel-Kleene
2. Sistemas de Post
3. Algoritmos de Markov

Sugerencias: Cuando se menciona que los conceptos de computable y decidible son equivalentes se podría profundizar en probar porque se puede hacer esta afirmación. Se podría a la hora de hablar del LPO suministrar algún ejemplo que permita ver la existencia de semidecidibilidad del problema.

CONFERENCIA 2

Decimos que un número es Turing-computable si existe una Máquina de Turing que lo genera y recordando la tesis de Church-Turing será computable. El conjunto de los números naturales es computable, el procedimiento para probarlo es el empleado por Dedekind-Peano, que consiste en lo siguiente $0 \in \mathbf{N}$, luego si $n \in \mathbf{N}$ entonces $\text{suc}(n) \in \mathbf{N}$, para construir una máquina de turing que genere cualquier número natural se puede emplear la representación unaria, de la siguiente forma, el único símbolo de nuestro sistema unario será 1, luego la regla de formación sería la siguiente:

1. $\bar{0} = 1$
2. $\overline{n+1} = \bar{n} 1$

Con esta representación queda de forma intuitiva la creación de la función sucesor en la MT. Adicionalmente los conjuntos \mathbf{Z} y \mathbf{Q} tienen definiciones constructivas a partir de \mathbf{N}

El conjunto de los números reales está compuesto por racionales e irracionales, entre las alternativas definitorias podemos encontrar las cortaduras de Dedekind, las clases de equivalencia de sucesiones de Cauchy y la axiomática que expone que los reales son un Cuerpo Ordenado Completo. En el caso de los irracionales los cuales no presentan un período se desea saber si son computables ya que estos definen la computabilidad del conjunto de los números reales.

Para definir el concepto de enumerable primeramente es necesario definir que es la correspondencia biunívoca de pares de conjuntos. Si dos conjuntos pueden ser puestos en correspondencia 1 a 1 mediante una función biyectiva (es posible encontrar mas de una función biyectiva para la correspondencia entre los conjuntos), entonces tienen la misma cardinalidad de elementos.

Sabemos que $\mathbf{N} \subset \mathbf{Z} \subset \mathbf{Q}$, por tanto podríamos suponer que a medida que avanzamos en la pertenencia de conjunto aumenta la cantidad de elementos, tal como planteo Euclides, "el todo es mayor que cualquiera de sus partes propias", pero en 1638 nota que los cuadrados de los enteros positivos pueden ser puesto en correspondencia con todos los números positivos, más tarde Bolzano considera la equivalencia geométrica entre el espacio unidimensional y el espacio n-dimensional.

Para definir si un conjunto es enumerable tomaremos el conjunto \mathbf{N} , entonces cualquier conjunto que pueda ser puesto en correspondencia 1 a 1 con el el conjunto \mathbf{N} será enumerable, por tanto el propio \mathbf{N} es enumerable y también cualquier conjunto finito será enumerable, ejemplos de conjuntos infinitos son los enteros y los cuadrados de los enteros positivos.

Otro conjunto de importancia es el conjunto de los números algebraicos, son aquellos números que son raíces de las ecuaciones algebraicas en una variable x con coeficientes enteros, cuya forma

general es: $a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = 0$ con $a_0 \neq 0$. Este conjunto es enumerable debido a que si es posible enumerar el conjunto de todas las ecuaciones algebraicas entonces es posible enumerar sus soluciones, para ello se reemplaza las ecuaciones por sus soluciones las cuales son iguales al grado de la ecuación, para cada una contamos con un conjunto finito numerable, por tanto se obtiene una enumeración con repeticiones de los números algebraicos las cuales pueden ser eliminadas.

La existencia de conjuntos no enumerables fue probada por Cantor quien utilizó un método conocido como *diagonalización*, para ello empleo la tabla infinita de funciones de una variable $f : \mathbf{N} \rightarrow \mathbf{N}$, en la cual al analizar la diagonal nos percatamos que esta también es una función de una variable, al notar la diagonal observamos que su ecuación es: $f(a) = f_a(a) + 1$, luego por reducción al absurdo asumamos que el índice i es con el cual la función diagonal fue enumerada, f_i es la función de la diagonal por tanto para i resultaría $f(i) = f_i(i) = f_i(i) + 1$ obteniendo que no es posible enumerar la función de la diagonal. Otro ejemplo de conjunto no enumerable demostrado empleando este método de Cantor es: el conjunto de todos los subconjuntos (enumerables) de \mathbf{N} .

Empleando los números algebraicos se encontró que números que eran solución de ecuaciones algebraicas eran irracionales, por ejemplo $\sqrt{2}$ es solución de $x^2 - 2 = 0$, el cual es un número computable, ahora ¿será posible obtener todos los números irracionales mediante soluciones de ecuaciones algebraicas?, como sabemos no, estos números que no son raíces se denominan *trascendentes*, el ejemplo que tenemos es π y también $2^{\sqrt{2}}$ el cual se demostró que no es algebraico. No existe un algoritmo para determinar si un número real es trascendental o no, ni lo puede haber según demostró Turing.

Según lo anterior podemos concluir que si no es posible determinar si un número es trascendental entonces no es posible enumerar el conjunto de los números reales, por tanto se puede decir que los conceptos de enumerable y computable son equivalentes, ya que no es posible determinar mediante un proceso si un número es trascendental, por tanto no es posible enumerarlo.

CONFERENCIA 3

La *Universal Turing Machine* es una máquina que construye y que es capaz de simular el proceso que realiza cualquier otra MT. Para realizar tal simulación, la MUT (por sus siglas en español) recibe como entrada en su cinta además de los datos de un problema, el programa (las instrucciones) de cualquier MT y simula su ejecución dejando sobre su cinta el resultado o salida que produce la MT simulada.

La idea fundamental detrás de esta es que en vez de tener sólo máquinas (MTs) que resuelven problemas particulares, tener una máquina universal (MUT) capaz de simular la computación que realiza cualquier MT resolviendo un problema particular. Considerando las instrucciones de cualquier MT como un programa, la MUT almacena programas y los ejecuta, surgiendo así por primera vez el concepto de máquina con programa almacenado, fundamento de las computadoras modernas.

Las instrucciones de la MUT son bastante complicadas e incluyen el uso de subrutinas (Turing utilizó por primera vez el concepto de subrutina). Partimos de la representación de un problema en código binario. Los datos que en la configuración inicial recibe la MUT son una sucesión binaria. Por lo tanto, podemos reducir el problema de la decidibilidad al problema de determinar si para el conjunto de sucesiones binarias (instrucciones-datos) que constituyen el problema existe una MUT (algoritmo) que finaliza en cualquier caso ya sea dando una respuesta afirmativa y proporcionando una solución al problema o bien diciendo que para los datos introducidos no existe solución computable.

Una MT se denomina *circular* si siempre imprime un número finito de dígitos binarios.

Una MT se denomina *no-circular* si imprime un número infinito de dígitos binarios.

Una secuencia de dígitos binarios se denomina *secuencia computable* si es una secuencia computada por alguna máquina *no-circular*. De esta forma se definen como computables secuencias de dígitos binarios que son infinitas. Turing pasa a demostrar que no toda secuencia infinita de dígitos binarios es *computable*.

Al quedar demostrado que no toda secuencia infinita de dígitos binarios es *computable* sabemos que no existe una MUT que de modo general determine si una máquina es *no-circular*. El resultado mas inmediato para la Ciencia de la Computación tiene que ver con un problema esencial de los programas, su terminación.

Aplicando el resultado de Turing: No hay un programa de computadora que pueda determinar en general si cualquier programa de computadora termina o no.