Complejidad Computacional

Problemas, Modelos y Recursos

Autor: Fausto David Hernández Jasso

August 28, 2022

1 Introducción

Los problemas son tan diferentes entre sí y tienen su origen en distintas variedades, algunos problemas son fáciles de resolver como puede ser **ordenar los monedas por lo grande de su diámetro** o pueden ser sumamente difíciles como **tener un buen horario en el semestre**. Del razonamiento anterior, surge la siguiente pregunta: **¿Qué hace díficil a un problema?**
A lo largo de la primera mitad del siglo pasado, prominentes matemáticos como **Kurt Gödel, Alan Turing** y **Alonzo Church** hicieron un trabajo excepcional al descubrir que ciertos problemas no podían ser resultos por las computadoras, una de las consecuencias de éste trabajo es que en particular **Alan Turing** en su artículo *"On Computable Numbers, with an Application to the Entscheidungsproblem"* demostró que no era posible determinar sí un premisa matemática es verdadera o falsa *(Alonzo Church demostró lo mismo a través del cálculo lambda, en su famoso)* **Teorema de Church**. Lo anterior implica que no existe un algoritmo que pueda realizar ésta tarea.
Así, a través de la computabilidad podemos dar una clasificación a los problemas sí éstos pueden ser resultos por un módelo de cómputo o no, a diferencia de la complejidad que ésta nos permite clasificar los problemas en sí éstos son difíciles o fáciles.

2 Desarrollo

La teoría de la complejidad es el enfoque matemático del estudio de las cosas computables *(es decir, aquellas que pueden ser calculados a través de un algoritmo ejecutado por algún modelo de cómputo)*, y de los recursos que se requieren utilizar para realizar el cómputo. A lo largo de la historia de la humanidad, las personas se han encargado de calcular ciertas cosas por su propia mano, ejemplificando, cuando una persona iba a comprar su comida a algún local, a inicios del siglo pasado, la persona que le atendía, calculaba el precio total sin ayuda de una computadora, sino simplemente lo hacía siguiendo el algoritmo para sumar, sin embargo, ahora ya no sucede así, ya que podemos realizar éste y muchos más cálculos con ayuda de una computadora. Pero ¿qué es calculable?, ¿para qué cosas podemos escribir un algoritmo que sea ejecutado por una computadora y evitar calcularlo a mano?, éstas preguntas se responden a través de la definición de computabilidad, en esencia y no siendo formal estrictamente, algo es computable sí existe un algoritmo que puede calcularlo.
En el primer parráfo de éste texto hablamos acerca de los problemas, utilizaremos la siguiente definición para referirnos a un problema:
Definimos a un problema P como el problema de decisión para un lenguaje formal L, esto es, dada una cadena de entrada s, determinaremos sí s pertenece al lenguaje L o no.
Hacemos la observación que pensar a los problemas como lenguajes formales, está lleno de ventajas, ya que podemos utilizar todo el trabajo creado para los **lenguajes recursivos** y **recursivamente enumarables** para resolver los problemas, desde ahora nuestro estudio estará concentrado en la **clase compleja** *(complexity class)*, que la definimos como el conjunto de lenguajes que pueden ser solucionados bajo una restricción de recursos.
¿Cómo saber cuando un problema (lenguaje) es más complejo que otro?, esencialmente definiremos la complejidad de un lenguaje tomando en cuenta el recursos gastados para resolver el problema de decisión relacionado con dicho lenguaje.
Por lo explicado en el parráfo, daremos la siguiente definición puramente matemática:
La complejidad de un lenguaje L en relación al uso de recursos que requiere para resolver el problema de decisión de L, está dado por la función $f(n)$, ésta función es la cantidad máxima de recursos usados por L, donde n es el tamaño de la cadena s de entrada. Nuestro enfoque será cuando $f(n)$ va creciendo conforme n se va haciendo terriblemente grande. Debido a ésta definición cualquier **algoritmo de decisión** creado para determinar si una cadena s pertenece o no a un lenguaje L tendrá dos cotas, una superior y una inferior, la cuales, naturalmente, serán definidas como lo máximo de recursos que vamos a usar y lo que al menos tenemos que usar para resolver el problema de decisión, respectivamente.
Solamente estamos interesados en cotas polinomiales ya que para entradas bastante grandes, se comportan de manera decente.
Los modelos de cómputo que utilizaremos son: máquinas y circuitos. A partir de éste punto, hablaremos solamente de las máquinas como modelo de computo. Una máquina es una abstracción de la computadora

actual, y ésta consistirá en leer una entrada, realizar ciertos cálculos sobre su memoria interna y devolver una salida.

3 Conclusión

Las medidas de recursos para las máquinas será el tiempo y el espacio. De la medida del tiempo sale una clase extremadamente importante como lo es a clase P, que es la clase que contiene al conjunto de lenguajes en los que se puede resolver el problema de decisión en tiempo polinomial. De la medida del espacio emanan 3 clases:

- **DSPACE(1)**.
- **DSPACE**$(\log n)$
- **PSPACE**.

Gracias a lo anterior tenemos una clasificación concreta y precisa de los problemas, ya sea tomando como referencia la medida del recurso tiempo o el recurso espacio.