ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Departamento de Estadística, I.O. y Computación

Teoría de Autómatas y Lenguajes Formales

Cardinalidad

Suponga que tenemos un programa listo para ejecutarse en un ordenador. El programa está diseñado para aceptar cierta entrada (quizás una lista de nombres que debe clasificarse) y producir determinada salida (como sería la lista ordenada). Sin importar cuáles sean la entrada y la salida, desde el punto de vista de la máquina no son más que una cadena de bits. Por lo tanto, podemos considerar que tanto la entrada como la salida son números naturales representados en notación binaria (podemos imaginar un 1 adicional colocado en el extremo izquierdo de estas cadenas, de modo que los 0 iniciales de las cadenas originales sean significativos). Desde esta perspectiva, el programa acepta un número de entrada y en base al valor de ese número, produce otro número como salida. Es decir, la acción del programa es calcular una función de $\mathbb{N}$ a $\mathbb{N}$. De forma análoga, cada función de $\mathbb{N}$ a $\mathbb{N}$ puede representar un programa que en alguna ocasión podríamos escribir.

¿Cuántas de éstas funciones existen? Obviamente, no hay más funciones de $\mathbb{N}$ a $\{0, 1\}$ que de $\mathbb{N}$ a $\mathbb{N}$, ya que éstas últimas proporcionan más valores de salida potenciales, Sin embargo, hay tantas funciones de $\mathbb{N}$ a $\{0, 1\}$ como elementos en el conjunto potencia de $\mathbb{N}$. Incluso, para cada subconjunto X de $\mathbb{N}$ hay una función llamada función característica de X, de $\mathbb{N}$ a $\{0, 1\}$, definida por:

$$f(n) = \begin{cases} 1 & \text{si } n \in X \\ 0 & \text{si } n \notin X \end{cases}$$

Concluimos que existen por lo menos tantas funciones de $\mathbb{N}$ a $\mathbb{N}$ como elementos de $\wp(\mathbb{N})$. Por lo tanto, existe una cantidad incontable de funciones para las cuales podríamos escribir programas.

Ahora considere su lenguaje de programación preferido. Existe un número finito de símbolos a partir de los cuales se construye cualquier programa en dicho lenguaje. Establezcamos un orden “alfabético” para estos símbolos. Luego podríamos generar una lista de todos los programas en el lenguaje que tienen longitud de un símbolo (si acaso existe alguno) en orden alfabético; después podríamos presentar en orden alfabético los programas con longitud dos; a continuación, los programas de longitud tres en orden alfabético, etcétera. Esta lista contendría cualquier programa que pudiera escribirse en este lenguaje. Por consiguiente, este procedimiento de generación de la lista sugiere que el conjunto de los programas que pueden escribirse en el lenguaje es contable.

¿Qué hemos demostrado con esto? Sólo existe una cantidad contable de programas que pueden escribirse en su lenguaje de programación favorito, pero hay una cantidad incontable de funciones que podrían calcularse con un programa. Puesto que cada programa calcula sólo una función, se quedará corto. Por ende, existen problemas que se quisieran resolver utilizando un computador, ¡pero para los cuales no puede escribirse un programa en su lenguaje de programación preferido!

¿Cuáles son algunos de estos problemas? ¿Es posible resolver alguno de ellos cambiando a otro lenguaje de programación? ¿Podrían resolverse estos problemas si construimos un computador más grande? Incluso si usted puede producir un programa para resolver un problema, ¿sería capaz una máquina actual de ejecutarlo con la rapidez suficiente para producir una respuesta mientras usted esté vivo? ¿Será capaz el computador del futuro de producir una respuesta durante la vida de sus hijos? Estas son algunas de las preguntas que directa o indirectamente trataremos en esta asignatura.