

1. Sobre el trabajo

El presente trabajo trata sobre la equivalencia de distintos tipos de modelos de cómputo; más concretamente se centra en demostrar que la potencia de cómputo de los lenguajes actuales, las máquinas de Turing y las funciones recursivas es igual, esto es, que cualquier algoritmo que se puede expresar en cualquiera de estos modelos se puede expresar también en los otros dos. En toda la explicación se da por hecho que se conoce la teoría sobre las máquinas tal y como las definió originalmente Turing ([1]) y las reglas para definir funciones recursivas presentadas por S. C. Kleene en [5].

1.1. Justificación y explicación

Las máquinas de Turing y el λ -cálculo inventado por Alonzo Church, así como la teoría sobre las funciones recursivas, nacieron en el siglo XX en el marco de los estudios de lógica y el límite de lo resoluble. La tesis Church-Turing nos dice que cualquier problema para el que exista un algoritmo que lo solucione puede resolverse tanto con una máquina de Turing como usando λ -cálculo.

La tesis no se puede demostrar pero tampoco ha sido refutada, con lo que sabemos que no hay método conocido más potente que estos dos para expresar soluciones a problemas. Cuando un lenguaje tiene la misma capacidad que ellos para definir algoritmos se dice que es Turing-completo, esto es, que puede definirse con él una solución para cualquier problema que pueda resolver una máquina de Turing. Sin embargo a la mayoría de personas que empieza a estudiar computación la duda sobre la capacidad de cómputo le surge en sentido contrario: ¿cómo es posible con un método tan simple como una máquina de Turing resolver lo que resuelve un ordenador actual?

Este trabajo no contiene, pues, algo novedoso ni especialmente interesante; simplemente pretende responder a la pregunta anterior dando una demostración más, desde un punto de vista práctico, de algo que aunque sabido no es obvio. Algo así como dar un modo de hacer recetas de Thermomix en una cocina de posguerra; se necesita más tiempo y se mancha más, pero es posible hacerlo.

1.2. Estructura del trabajo

La mayoría de demostraciones sobre sistemas de cómputo que se pueden encontrar se basan en conceptos teóricos. Las demostraciones dadas aquí son más pedestres por así decirlo; se hacen presentando soluciones completas. Se empezará con la motivación primera del trabajo, que es la demostración de que para cualquier algoritmo, por complejo que sea, que se escriba en un lenguaje de programación de los usados hoy, existe una máquina de Turing que ejecuta u