TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INTEGRANTE:

PEREZ ARMAS FAUSTO ISAAC

181080037

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 15

RESUMEN TEMA 8.1

El propósito de la teoría de los problemas indecidibles no es sólo establecer la existencia de tales problemas (una idea excitante por sí misma desde el punto de vista intelectual) sino proporcionar también una guía a los programadores sobre lo que se puede o no conseguir a través de la programación. La teoría también tiene un gran impacto práctico, como veremos en el Capítulo 10, al tratar problemas que aunque sean decidibles, requieren mucho tiempo para ser resueltos. Estos problemas, conocidos como “problemas intratables”, suelen plantear una mayor dificultad al programador y al diseñador de sistemas que los problemas indecidibles. La razón de ello es que mientras que los problemas indecidibles normalmente suelen resultar obvios y habitualmente no se intentan resolver, los problemas intratables se presentan continuamente. Además, a menudo dan lugar a pequeñas modificaciones de los requisitos o a soluciones heurísticas. Por tanto, el diseñador se enfrenta con frecuencia a tener que decidir si un problema es o no intratable, y qué hacer si lo es. Necesitamos herramientas que nos permitan determinar cuestiones acerca de la indecidibilidad o intratabilidad todos los días. La tecnología presentada en la Sección 8.1 resulta útil para cuestiones que tratan con programas, pero no se puede trasladar fácilmente a problemas en otros dominios no relacionados.

Una ventaja de la máquina de Turing sobre los programas como representación de lo que se puede calcular es que la máquina de Turing es lo suficientemente simple como para que podamos representar su configuración de manera precisa, utilizando una notación sencilla muy similar a las descripciones instantáneas de un autómata a pila. En cambio, aunque los programas en C tienen un estado, que implica a todas las variables en cualquier secuencia de llamadas a función que se realice, la notación para describir estos estados es demasiado compleja como para poder realizar demostraciones formales comprensibles. Con la notación de la máquina de Turing, demostraremos que ciertos problemas, que aparantemente no están relacionados con la programación, son indecidibles.

Notación para la máquina de Turing Podemos visualizar una máquina de Turing. La máquina consta de una unidad de control, que puede encontrarse en cualquiera de un conjunto finito de estados. Hay una cinta dividida en cuadrados o casillas y cada casilla puede contener un símbolo de entre un número finito de símbolos

Inicialmente, la entrada, que es una cadena de símbolos de longitud finita elegidos del alfabeto de entrada, se coloca en la cinta. Las restantes casillas de la cinta, que se extiende infinitamente hacia la izquierda y la derecha, inicialmente almacenan un símbolo especial denominado espacio en blanco.

El espacio en blanco es un símbolo de cinta, pero no un símbolo de entrada, y pueden existir también otros símbolos de cinta además de los símbolos de entrada y del espacio en blanco. Existe una cabeza de la cinta que siempre está situada en una de las casillas de la cinta. Se dice que la máquina de Turing señala dicha casilla. Inicialmente, la cabeza de la cinta está en la casilla más a la izquierda que contiene la entrada. Un movimiento de la máquina de Turing es una función del estado de la unidad de control y el símbolo de cinta al que señala la cabeza. En un movimiento, la máquina de Turing:

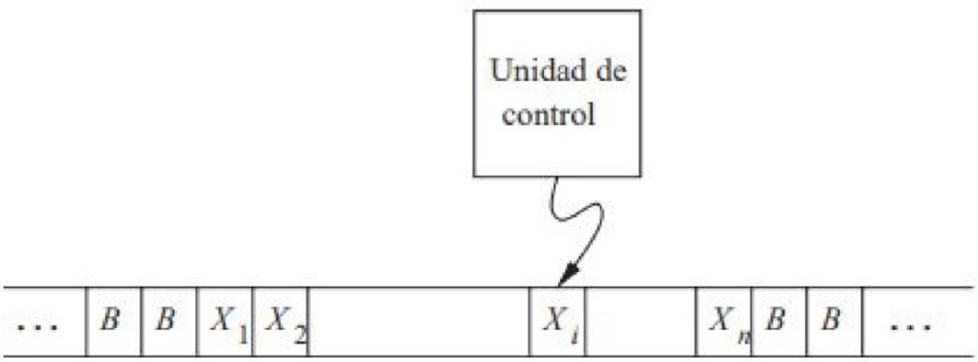

Figura 8.8. Una máquina de Turing.