|  |  |  |  |
| --- | --- | --- | --- |
| **CARRERA** | **PLAN DE ESTUDIO** | **CLAVE ASIGNATURA** | **NOMBRE DE LA ASIGNATURA** |
| Ingeniería en Computación | 2003-1 | 5044 | Teoría de la Computación |

|  |  |  |  |
| --- | --- | --- | --- |
| **Práctica No.** | **LABORATORIO DE** | Teoría de la Computación | **DURACIÓN****(HORAS)** |
| 6 | **Nombre de la Práctica** | Lenguajes enumerables y decidibles | 2 |

|  |  |
| --- | --- |
| Elaboró:Christian Navarro Cota | Revisó:Odin Isaac Meling López |

**1.- INTRODUCCIÓN:**

La máquina de Turing es un modelo computacional introducido por Alan Turing en el cual se estudiaba la cuestión planteada por David Hilbert sobre si las matemáticas son decidibles, es decir, si hay un método definido que pueda aplicarse a cualquier sentencia matemática y que nos diga si esa sentencia es cierta o no.

**2.- OBJETIVO (COMPETENCIA):**

El alumno se adentrara a los lenguajes formales de tipo enumerable y decidible a través del modelo de la maquina de turing.

**3.- MARCO TEÓRICO:**

La máquina de Turing consta de un cabezal lector/escritor y una cinta infinita en la que el cabezal lee el contenido, borra el contenido anterior y escribe un nuevo valor. Las operaciones que se pueden realizar en esta máquina se limitan a:

Avanzar el cabezal lector/escritor hacia la derecha. avanzar el cabezal lector/escritor hacia la izquierda. El cómputo es determinado a partir de una tabla de estados de la forma:

(estado, valor) 🡪 (nuevo estado, nuevo valor, dirección)

Esta tabla toma como parámetros el estado actual de la máquina y el carácter leído de la cinta, dando la dirección para mover el cabezal, el nuevo estado de la máquina y el valor a ser escrito en la cinta.

Con este aparato extremadamente sencillo es posible realizar cualquier cómputo que un computador digital sea capaz de realizar.

Mediante este modelo teórico y el análisis de complejidad de algoritmos, fue posible la categorización de problemas computacionales de acuerdo a su comportamiento, apareciendo así, el conjunto de problemas denominados P y NP, cuyas soluciones en tiempo polinómico son encontradas según el determinismo y no determinismo respectivamente de la máquina de Turing.

Una máquina de Turing con una sola cinta puede ser definida como una 6-tupla M=(Q,T,s,b,F,S) , donde:

•*Q* es un conjunto finito de estados.

•*T* es un conjunto finito de símbolos de cinta, el alfabeto de cinta.

•*s* E Q es el estado inicial.

•*b* E L es un símbolo denominado blanco, y es el único símbolo que se puede repetir un número infinito de veces.

•*F* \_C Q es el conjunto de estados finales de aceptación.

•*S* : Q x L → Q x L x {L,R} es una función parcial denominada función de transición, donde L es un movimiento a la izquierda y R es el movimiento a la derecha.

Existen en la literatura un abundante número de definiciones alternativas, pero todas ellas tienen el mismo poder computacional, por ejemplo se puede añadir el símbolo S como símbolo de “no movimiento” en un paso de cómputo o el símbolo Σ para indicar el alfabeto de entrada.

**Ejemplo** Definimos una máquina de Turing sobre el alfabeto {0,1}, donde 0 representa el símbolo blanco. La máquina comenzará su proceso situada sobre un símbolo “1″ de una serie. La máquina de Turing copiará el número de símbolos “1″ que encuentre hasta el primer blanco detrás de dicho símbolo blanco. Es decir, situada sobre el 1 situado en el extremo izquierdo, doblará el número de símbolos 1, con un 0 en medio. Así, si tenemos la entrada “111″ devolverá “1110111″, con “1111″ devolverá “111101111″, y sucesivamente.

El conjunto de estados es {s1,s2,s3,s4,s5} y el estado inicial es s1. La tabla que describe la función de transición es la siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESTADO | S.LEIDO | S.ESCRITO | MOV. | ESTADO SIG. |
| s1 | 1 | 0 | R | s2 |
| s2 | 1 | 1 | R | s2 |
| s2 | 0 | 0 | R | s3 |
| s3 | 0 | 1 | L | s4 |
| s3 | 1 | 1 | R | s3 |
| s4 | 1 | 1 | L | s4 |
| s4 | 0 | 0 | L | s5 |
| s5 | 1 | 1 | L | s5 |
| s5 | 0 | 1 | R | s1 |

El funcionamiento de una computación de esta máquina se puede mostrar con el siguiente ejemplo (en negrita se resalta la posición de la cabeza lectora/escritora):

|  |  |  |
| --- | --- | --- |
| PASO | ESTADO | CINTA |
| 1 | s1 | 11 |
| 2 | s2 | 01 |
| 3 | s2 | 010 |
| 4 | s3 | 0100 |
| 5 | s4 | 0101 |
| 6 | s5 | 0101 |
| 7 | s5 | 0101 |
| 8 | s1 | 1101 |
| 9 | s2 | 1001 |
| 10 | s3 | 1001 |
| 11 | s3 | 10010 |
| 12 | s4 | 10011 |
| 13 | s4 | 10011 |
| 14 | s5 | 10011 |
| 15 | s1 | 11011 |
| PARADA |  |  |

La máquina realiza su proceso por medio de un bucle, en el estado inicial s1, reemplaza el primer 1 con un 0, y pasa al estado s2, con el que avanza hasta la derecha, saltando los símbolos 1 hasta un 0 (que debe existir), cuando lo encuentra pasa a ser s3, con este estado avanza saltando los 1 hasta encontrar otro 0 (la primera vez no habría ningún 1). Una vez en el extremo derecho, añade un 1. Después comienza el proceso de retorno; con s4 vuelve a la izquierda saltando los 1, cuando encuentra un 0 (en el medio de la secuencia), pasa a s5 que continúa a la izquierda saltando los 1 hasta el 0 que se escribió al principio. Se reemplaza de nuevo este 0 por 1, y pasa al símbolo siguiente, si es un 1, se pasa a otra iteración del bucle, pasando al estado s1 de nuevo. Si es un símbolo 0, será el símbolo central, con lo que la máquina se detiene al haber finalizado su cómputo.

**Máquinas de Turing deterministas y no deterministas**

La entrada de una máquina de Turing viene determinada por el estado actual y el símbolo leído, un par [estado, símbolo], siendo el cambio de estado, la escritura de un nuevo símbolo y el movimiento las acciones a tomar en función de una entrada. En el caso de que para cada par estado y símbolo posible exista a lo sumo una posibilidad de ejecución, se dirá que es una máquina de Turing determinista, mientras que en el caso de que exista al menos un par [estado, símbolo] con más de una posible combinación de actuaciones se dirá que se trata de una máquina de Turing no determinista.

La función de transición δ en el caso no determinista, queda definida como sigue:

*S*: *Q* X T→ *P*(*Q*X T X {*L*,*R*})

¿Cómo sabe una máquina no determinista cuál de las varias actuaciones tomar? Hay dos formas de verlo: una es decir que la máquina es “el mejor adivino posible”, esto es, que siempre elige la transición que eventualmente la llevará a un estado final de aceptación. La otra es imaginarse que la máquina se “clona”, bifurcándose en varias copias, cada una de las cuales sigue una de las posibles transiciones. Mientras que una máquina determinista sigue un solo “camino computacional”, una máquina no determinista tiene un “árbol computacional”. Si cualquiera de las ramas del árbol finaliza en un estado de aceptación, se dice que la máquina acepta la entrada.

La capacidad de cómputo de ambas versiones es equivalente; se puede demostrar que dada una máquina de Turing no determinista existe otra máquina de Turing determinista equivalente, en el sentido de que reconoce el mismo lenguaje, y viceversa. No obstante, la velocidad de ejecución de ambos formalismos no es la misma, pues si una máquina no determinista M reconoce una cierta palabra de tamaño n en un tiempo O(t(n)), la máquina determinista equivalente reconocerá la palabra en un tiempo O(2t(n)). Es decir, el no determinismo permitirá reducir la complejidad de la solución de los problemas, permitiendo resolver, por ejemplo, problemas de complejidad exponencial en un tiempo polinómico.

**4.- DESCRIPCIÓN**

**A) PROCEDIMIENTO Y DURACIÓN DE LA PRÁCTICA:**

**Investigue cada uno de los tipos de maquinas de turing que existen y realice un resumen de ello.**

**B) CÁLCULOS Y REPORTE:**

Se presentara el resumen de acuerdo a los criterios del maestro.

**C) RESULTADOS Y CONCLUSIONES:**

El trabajo se presentara frente a la clase y se debatirá el como es que funciona, así como sus ventajas y desventajas.

**5.- ANEXOS:**

* Introducción a la teoría de autómatas, lenguajes y computación.

Hopcroft, J. E.; Motwani, R.; Ullman, J. D.

* Introduction to the theory of computation, Michael Sipser, PWS Publishing Company, 1997