

Universidad de Oriente Núcleo Anzoátegui

Materia: Teoría de Autómatas y Lenguajes Formales

Profesor: José Bastardo

Sección: 01

Alumno: Luis Correa CI: xx.xxx.xxx

Departamento de Computación y Sistemas

Ingeniería en Computación

MAQUINAS DE TURING Y COMPLEJIDAD

Fecha 06 – 08 - 12

**INDICE**

|  |
| --- |
| [INTRODUCCIÓN 1](#_Toc331883507)  [1.- Defina formalmente una maquina de turing: 2](#_Toc331883508)  [2.- ¿Qué diferencia existe entre la memoria de una maquina de pila y la memoria de una maquina de turing?: 3](#_Toc331883509)  [3.- De ejemplo de un lenguaje l definido sobre ∑ = {0,1} que puede ser reconocido por una maquina de turing y no por una maquina de pila: 3](#_Toc331883510)  [4.- explique que es un problema np: 4](#_Toc331883511)  [5.- explique que es un problema np-completo: 4](#_Toc331883512)  [BIBLIOGRAFIA 5](#_Toc331883513) |
|  |
|  |
|  |

# INTRODUCCIÓN

Los autómatas son modelos para diseñar circuitos digitales, Analizadores léxicos de un compilador. Buscar por palabras claves en un archivo o en internet, como en protocolos de comunicación.

Los autómatas son muy importantes ya que sin ellos no podría saberse los cambios de estado de una maquina.

Por ejemplo tenemos la Tesis de Church: Todo lo que es computable se puede calcular con una Máquina de Turing. Existen problemas que no son computables.

Es decir, la Teoría de los Lenguajes Formales (y de los Autómatas) permite responder a preguntas esenciales de la Informática.

## 1.- Defina formalmente una maquina de turing:

Definición formal:

  Es un [modelo computacional](http://es.wikipedia.org/wiki/Modelo_computacional) que realiza una [lectura](http://es.wikipedia.org/wiki/Lectura)/[escritura](http://es.wikipedia.org/wiki/Escritura) de manera automática sobre una [entrada](http://es.wikipedia.org/wiki/Entrada) llamada cinta, generando una [salida](http://es.wikipedia.org/wiki/Salida_(inform%C3%A1tica)) en esta misma.

Este modelo está formado por un [alfabeto](http://es.wikipedia.org/wiki/Alfabeto) de entrada y uno de salida, un símbolo especial llamado blanco (normalmente *b*, *\Delta* o *0*), un conjunto de [estados](http://es.wikipedia.org/wiki/Estado_(inform%C3%A1tica)) finitos y un conjunto de transiciones entre dichos estados.

Su funcionamiento se basa en una [función de transición](http://es.wikipedia.org/wiki/Funci%C3%B3n_de_transici%C3%B3n), que recibe un *estado inicial* y una [cadena de caracteres](http://es.wikipedia.org/wiki/Cadena_de_caracteres) (la cinta, la cual puede ser infinita) pertenecientes al [alfabeto](http://es.wikipedia.org/wiki/Alfabeto) de entrada.

La máquina va leyendo una celda de la cinta en cada paso, borrando el símbolo en el que se encuentra posicionado su cabezal y escribiendo un nuevo símbolo perteneciente al alfabeto de salida, para luego desplazar el cabezal a la izquierda o a la derecha (solo una celda a la vez). Esto se repite según se indique en la [función de transición](http://es.wikipedia.org/wiki/Funci%C3%B3n_de_transici%C3%B3n), para finalmente detenerse en un estado final o de aceptación, representando así la salida.

Una máquina de Turing con una sola cinta puede definirse como una 7-[tupla](http://es.wikipedia.org/wiki/Tupla)

M=(Q, \Sigma, \Gamma, s, b, F, \delta),\!

Donde:

Q\! Es un conjunto finito de [estados](http://es.wikipedia.org/wiki/Estado_f%C3%ADsico).

\Sigma\! Es un conjunto finito de símbolos distinto del espacio en blanco, denominado alfabeto de máquina o de entrada.

\Gamma\! Es un conjunto finito de símbolos de cinta, denominado alfabeto de cinta (\Sigma \subseteq\Gamma).

s \in Q Es el estado inicial.

b \in \Gamma Es un símbolo denominado blanco, y es el único símbolo que se puede repetir un número infinito de veces.

F \subseteq Q Es el conjunto de estados finales de aceptación.

\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\} Es una [función parcial](http://es.wikipedia.org/wiki/Funci%C3%B3n_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.

## 2.- ¿Qué diferencia existe entre la memoria de una maquina de pila y la memoria de una maquina de turing?:

* Autómata de pila:

Solo se cuenta con la información suministrada por la sima de la pila.

* Maquina de turing:

Además de almacenar datos por las cadenas recibidas sin necesidad de desapilar nada es posible realizar consultas a los datos que se encuentran bajo la sima.

## 3.- De ejemplo de un lenguaje l definido sobre ∑ = {0,1} que puede ser reconocido por una maquina de turing y no por una maquina de pila:

Maquina que acepta cadenas que tengan el mismo número de a’s que b’s definimos la máquina de turing como una 8 tupla, la maquina acepta esta cadena

Alfabeto de cadena

Todos los estados

Estado inicial

Estado final

Símbolo con el cual sustituimos símbolos de la cadena

Símbolo final de la cadena

Símbolo inicial de la cadena

, ,

, ,

, ,

, ,

,

} Transiciones de los estados, D significa mover apuntador a la derecha, E moverlo a la izquierda

## 4.- explique que es un problema np:

Es el conjunto de los problemas de decisión que pueden ser resueltos por una máquina no determinista en tiempo polinómico. Esta clase contiene muchos problemas que se desean resolver en la práctica, incluyendo el [problema de satisfacibilidad booleana](http://es.wikipedia.org/wiki/Problema_de_satisfacibilidad_booleana) y el [problema del viajante](http://es.wikipedia.org/wiki/Problema_del_viajante), un camino Hamiltoniano para recorrer todos los vértices una sola vez.

## 5.- explique que es un problema np-completo:

En [teoría de la complejidad computacional](http://es.wikipedia.org/wiki/Complejidad_computacional), la [clase de complejidad](http://es.wikipedia.org/wiki/Clase_de_complejidad) NP-completo es el subconjunto de los [problemas de decisión](http://es.wikipedia.org/wiki/Problema_de_decisi%C3%B3n) en [NP](http://es.wikipedia.org/wiki/NP_(Complejidad_computacional)) tal que todo problema en NP se puede [reducir](http://es.wikipedia.org/wiki/Transformaci%C3%B3n_polin%C3%B3mica) en cada uno de los problemas de NP-completo.

Se puede decir que los problemas de NP-completo son los problemas más difíciles de NP y muy probablemente no formen parte de la clase de complejidad [P](http://es.wikipedia.org/wiki/Tiempo_polin%C3%B3mico).

La razón es que de tenerse una solución polinómica para un problema NP-completo, todos los problemas de NP tendrían también una solución en tiempo polinómico.

Si se demostrase que un problema NP-completo, llamémoslo *A*, no se pudiese resolver en tiempo polinómico, el resto de los problemas NP-completos tampoco se podrían resolver en tiempo polinómico.

Esto se debe a que si uno de los problemas NP-completos distintos de *A*, digamos *X*, se pudiese resolver en tiempo polinómico, entonces *A* se podría resolver en tiempo polinómico, por definición de NP-completo. Ahora, pueden existir problemas en *NP* y que no sean NP-completos para los cuales exista solución polinómica aún no existiendo solución para *A*.

# BIBLIOGRAFIA

* [http://www.sc.ehu.es/jiwhehum2/TC/temas/[2]turing.pdf](http://www.sc.ehu.es/jiwhehum2/TC/temas/%5b2%5dturing.pdf)
* <http://www.mitecnologico.com/Main/LaMaquinaDeTuring>
* [http://www.mitecnologico.com](http://www.mitecnologico.com/)
* <http://www.wikiteka.com/trabajos/algoritmo-4/>