



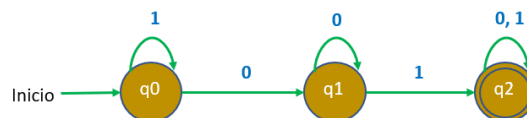
CC184 - Complejidad Algorítmica

Tema: Computabilidad – Modelo Formal: Máquina de Turing
Formato: Esquema de Aprendizaje
Elaborado por: Robert Zubieta
Fuente: Propia

Modelo Formal: Máquina de Turing

I. Autómata Finito Determinista

Autómatas Finitos Deterministas



Definiciones

Estados del Autómata: $Q = \{q_0, q_1, q_2\}$

Alfabeto: $\Sigma = \{0, 1\}$

Estado Inicial: q_0

Estados de Aceptación: $F = \{q_2\}$

Función de Transición: δ

Transiciones: Nos permiten ir de un estado a otro: Flechas

Estados de Transición:

$\delta(q_0, 1) = q_0$
 $\delta(q_0, 0) = q_1$

$\delta(q_1, 0) = q_1$
 $\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2$
 $\delta(q_2, 1) = q_2$

Automata Determinista:

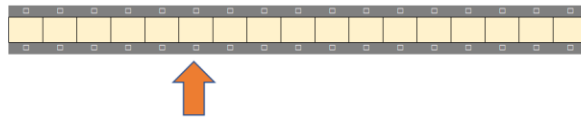
Para cualquier valor de entrada existe un único estado al que puede ir el autómata.

Autómata Finito Determinista

$A = (Q, \Sigma, \delta, q_0, F)$

II. Máquina de Turing

Máquina de Turing



DEFINICIÓN INFORMAL

Una **cinta** que se divide en celdas, una al lado de la otra. Cada celda contiene un símbolo de algún alfabeto finito.
 Un **cabezal** que puede leer y escribir símbolos en la cinta y mover la cinta a la izquierda y a la derecha **una celda a la vez**.
 Un **registro de estado** que almacena el estado de la máquina de Turing, uno de los estados finitos.
 Una **tabla** finita de instrucciones o reglas.

DEFINICIÓN FORMAL

Una máquina de Turing⁴⁸ es un **modelo computacional** que realiza una **lectura/escritura** de manera automática sobre una **entrada** llamada cinta, generando una **salida** en esta misma.

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

donde:
 Q es un conjunto finito de **estados**.
 Σ es un conjunto finito de símbolos distinto del espacio en blanco, denominado **alfabeto de máquina** o de entrada.
 Γ 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** denominada **función de transición**, donde L es un movimiento a la izquierda y R es el movimiento a la derecha.

$M = (Q, \Sigma, \Gamma, \delta, q_1, B, F)$
 Q : Conjunto finito de estados
 Σ : Alfabeto de entrada
 Γ : Alfabeto de cinta
 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times Mov$ donde $Mov \in \{L, R\}$
 $q_1 \in Q$: estado inicial
 $B \in \Gamma$: símbolo de blanco
 $F \subseteq Q$: estados de aceptación

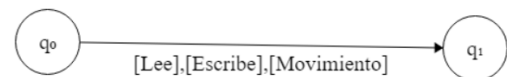
III. Función de Transición

Máquina de Turing

Regla de Transición

$\delta(q_0, Lee) = (q_1, Escribe, Movimiento)$

Diagrama de Estado



IV. Caso 01 – Diagrama

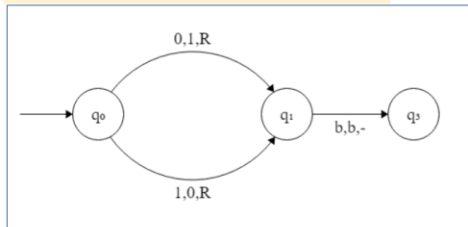
Máquina de Turing

Ejemplo 1:

Complemento a 2 en Binario: En una cadena, convierte 0 en 1 y 1 en 0

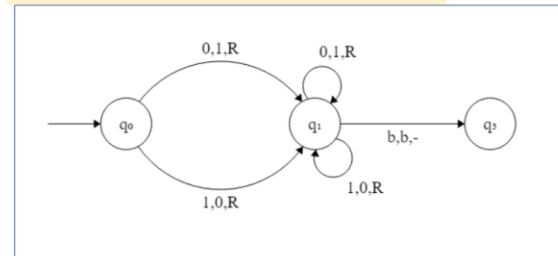
Diseño de la Máquina de Estado Finito

Convierte 0 en 1 o 1 en 0, para cadenas de longitud uno.



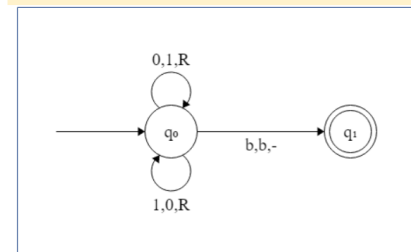
Diseño de la Máquina de Estado Finito

Convierte 0 en 1 o 1 en 0, para cadenas de longitud n: **Recursivo**



Diseño de la Máquina de Estado Finito (Simplificado)

Convierte 0 en 1 o 1 en 0, para cadenas de longitud n: **Recursivo**

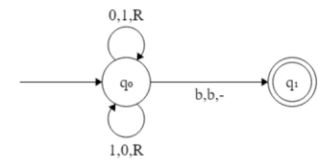
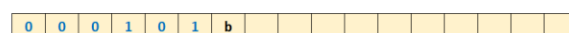
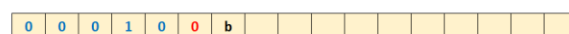
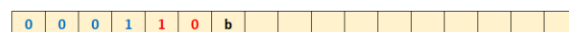
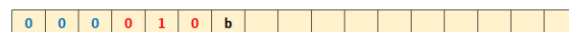
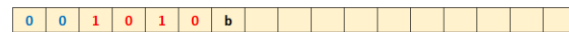
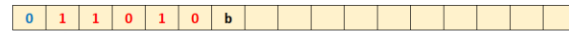
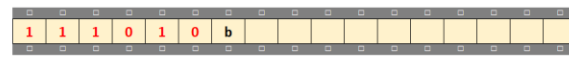


V. Caso 01 – Máquina de Turing

Máquina de Turing

Ejemplo 1:

Complemento a 2 en Binario: En una cadena, convierte 0 en 1 y 1 en 0



Reglas de Transición

$\delta(q_0, Lee) = (q_1, Escribe, Movimiento)$

$\delta(q_0, 0) = (q_0, 1, R)$

$\delta(q_0, 1) = (q_0, 0, R)$

$\delta(q_0, b) = (q_1, b, -)$

VI. Caso 01 – Simulador de Máquina de Turing

Turing Machine Simulator ([Click here for info and instructions](#))

Machine name = "Untitled"; max_state = 25; symbols = "xyzabc01S@"

0 1 1 1 0 1 0

Rule Editor

Old State	Old Symbol	New Symbol	New State	Move
0	#	#	h	S
0	0	1	0	R
0	1	0	0	R

Run Speed: moderate

Run Stop Step Step Back Reset State to Zero

Reglas de Transición

$\delta(q_0, \text{Lee}) = (q_1, \text{Escribe}, \text{Movimiento})$

$\delta(q_0, 0) = (q_0, 1, R)$

$\delta(q_0, 1) = (q_0, 0, R)$

$\delta(q_0, b) = (q_1, b, -)$

Ejemplo 1:
Complemento a 2 en Binario: En una cadena, convierte 0 en 1 y 1 en 0

URL: <https://math.hws.edu/eck/js/turing-machine/TM.html>

Simulador - Procedimiento

0. Click en New Turing Machine.
1. Ingresar las Reglas o Funciones de Transición, usando Rule Editor.
2. Ingresar el dato de entrada en formato binario.
3. Ubicar el cabezal al inicio del dato de entrada (a la izquierda, en el primer caracter).
4. Ejecutar con "Step" o con "Run".

Entrada: 111010

Salida: 000101

En el Simulador:

Old Symbol: Lo que se Lee
New Symbol: Lo que se Escribe
h: Halt: Estado de Aceptación.
#: caracter en blanco

VII. Caso 02 – Diagrama y Reglas

Máquina de Turing

Ejemplo 2:
Agrega 1 a un número binario

Reglas de Transición

$\delta(q_0, \text{Lee}) = (q_1, \text{Escribe}, \text{Movimiento})$

$\delta(q_0, b) = (q_1, 1, R)$

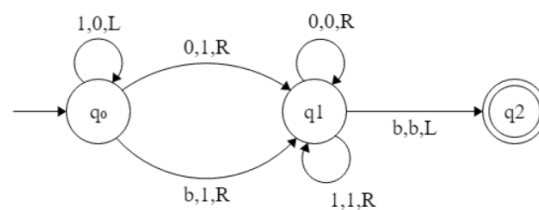
$\delta(q_0, 0) = (q_1, 1, R)$

$\delta(q_0, 1) = (q_0, 0, L)$

$\delta(q_1, b) = (q_2, b, L)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, 1) = (q_1, 1, R)$



VIII. Caso 02 – Simulador de Máquina de Turing

Turing Machine Simulator ([Click here](#) for info and instructions)

The Binary Increment Turing machine will add one to a binary number and halt. It must be started on the right end of the number that is to be incremented.

Machine name = "Binary Increment"; max_state = 25; symbols = "xyzabc01S@"

0

1 0 0 1 1 1 1 1

Rule Editor

OldState: 0 OldSymbol: 1 NewSymbol: 0 NewState: 0 Move: L

Run Stop Run Speed: moderate Step Step Back Reset State to Zero Install Example: Binary Increment

Old State	Old Symbol	New Symbol	New State	Move
0	#	1	1	R
0	0	1	1	R
0	1	0	0	L
1	#	#	h	L
1	0	0	1	R
1	1	1	1	R

Ejemplo 2:

Agrega 1 a un número binario

URL: <https://math.hws.edu/eck/js/turing-machine/TM.html>

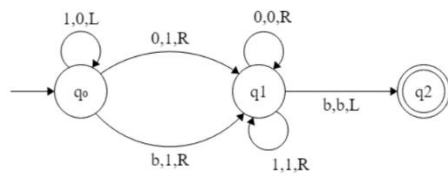
Simulador - Procedimiento

1. Ingresar las Reglas o Funciones de Transición, usando Rule Editor.
2. Ingresar el dato de entrada en formato binario.
3. Ubicar el cabezal al inicio del dato de entrada (a la derecha, en el primer carácter)
4. Ejecutar con "Step" o con "Run"

Entrada: 159 = 1001 1111

+ 1 =

Salida: 160 = 1010 0000



Reglas de Transición

 $\delta(q_0, \text{Lee}) = (q_1, \text{Escribe, Movimiento})$ $\delta(q_0, b) = (q_1, 1, R)$ $\delta(q_0, 0) = (q_1, 1, R)$ $\delta(q_0, 1) = (q_0, 0, L)$ $\delta(q_1, b) = (q_2, b, L)$ $\delta(q_1, 0) = (q_1, 0, R)$ $\delta(q_1, 1) = (q_1, 1, R)$