Máquina de Turing

Teoría Computacional - ESCOM - IPN - Prof. Rafael Aguilar

Contenido

- Introducción
- Definición
- Ejemplos
- Ejercicios

Introducción

- Ahora veremos los autómatas que reconocen lenguajes generados por las gramáticas más generales que hay (tipo 0 y tipo 1): Autómatas Linealmente Acotados y Máquinas de Turing.

Gramáticas	Lenguajes	Máquinas	Problemas
			No computables
Tipo 0 (G. sin restricción)	Recursivamente enumerables	Máquinas de Turing	Computables
Tipo 1 (G. sensible al contexto)	Lenguajes sensibles al contexto	Autómatas lineales acotados	
Tipo 2 (G. Libre de contexto)	Lenguajes independientes del contexto	Autómatas de pila	
Tipo 3 (G. Regular)	Lenguajes regulares	Autómatas finitos deterministas	Expresiones regulares

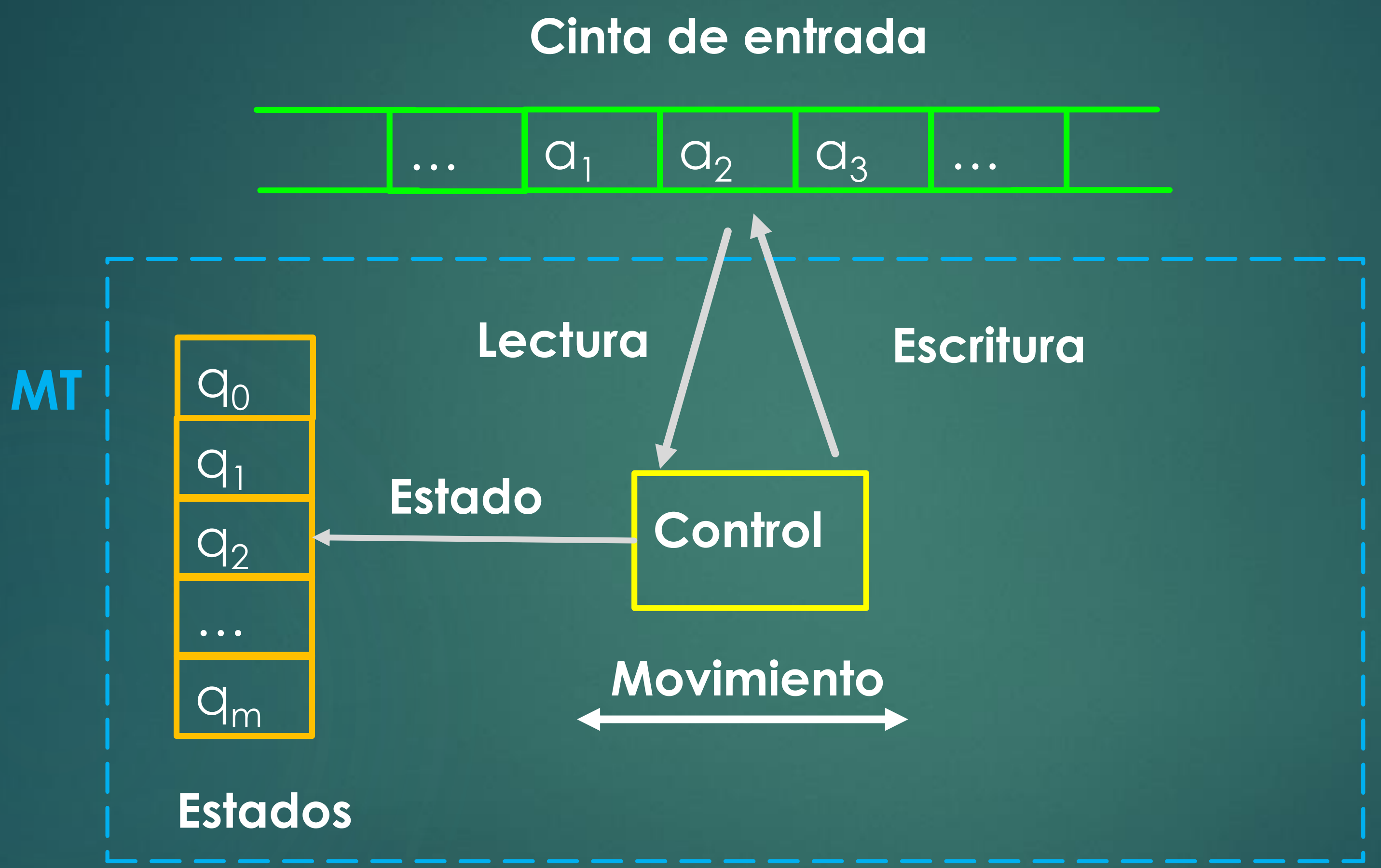
Introducción

- Estos dos tipos de autómata son menos restrictivos que los anteriores:
  - Permiten mover la cabeza de lectura en las dos direcciones: derecha e izquierda.
  - Permiten escribir en la cinta de entrada, ahora será de entrada/salida
- La Máquina de Turing dispone de una cinta teóricamente infinita
- Loa Autómatas Linealmente Acotados disponen de una cinta acotada a ambos lados.

Máquina de Turing

- Las Máquinas de Turing son el autómata más general, capaz de reconocer los lenguajes generados por las gramáticas menos restrictivas, las de tipo 0.
- Arquitectura de la Máquina de Turing:

MT – Concepto gráfico



Máquina de Turing - Definición

- Definimos formalmente una MT de la siguiente forma:

$$AP = \{ \Gamma, \sum, b, Q, q_0, f, F \}$$

donde :

Γ es el alfabeto de símbolos de la cinta

∑ es el alfabeto de la cadena de entrada.

b es el espacio en blanco $b \in \Gamma$, pero $b \notin \sum$

Q es el conjunto finito de estado

$f : Q \times \Gamma \rightarrow Q \times \Gamma \times \{ I, D, P \}$: es la función de transición, donde I es izquierda, D es derecha, y P es parada.

$q_0 \in Q$ es el estado inicial

$F \subset Q$ es el conjunto de estados finales

Máquina de Turing -Características

- La cinta es infinita, por lo que a los dos lados de la información que parezca en la entrada, habrá espacios en blanco representados por **b**.
- Inicialmente contiene un número finito de símbolos de ∑ precedidos y seguidos de infinitos espacios en blanco (**b**).
- Al igual que el resto de los autómatas, se representan por la tabla de transición en la que los estados están en las filas, los símbolos de **Γ** están en las columnas, y en la posición **(q, a)** aparece **f(q, a)**.

Máquina de Turing - Movimiento

- Si la MT está en un estado **q ε Q** , lee **a ε Γ** de la cinta de entrada (el puntero de lectura de la cinta de entrada está sobre el símbolo **a**), y la función de transición para esta pareja estado-símbolo es:

  f(q, a) = (p, x, m) ó pxm donde **p ε Q**, **x ε Γ, m ε { I, D, P }**

  entonces la MT:

  1. Transita al estado p;
  2. Imprime el símbolo x en la cinta en donde estaba el a;
  3. Mueve la cabeza lectora de la cinta de entrada/salida hacia la izquierda (m = I), derecha (m = D), o se para (m = P).

Ejemplo 1. Duplicador de 1's

- Esta máquina convierte un número **n** de unos en la cinta de entrada en un número **2n** de unos, como se describirá a continuación.
- Se puede definir la siguiente MT_1:

$$MT = \{ \Gamma, \Sigma, b, Q, q_0, f, F \}$$

$$MT_1 = (\{ 0, 1, b \}, \{ 1 \}, b, \{ p, q, r, s \}, p, f, \{ s \})$$

f	1	0	b
p	q0D	p0I	rbD
q	q1D	q0D	p0I
r		rID	sbP
s			

Ejemplo 1. Palabra de entrada: 11
Salida de la MT: 1111

1

b	1	1	b

p

$f(p, 1) = (q, 0, D)$

2

b	0	1	b

q

2

b	0	1	b

q

$f(q, 1) = (q, 1, D)$

3

b	0	1	b

q

Simulación de la ejecución de la MT

3

b	0	1	b	b	b
			q		

4

b	0	1	0	b	b
			p		

5

b	0	0	0	b	b
			q		

6

b	0	0	0	b	b
				q	

7

b	0	0	0	0	b
			p		

8

b	0	0	0	0	b
		p			

9

b	0	0	0	0	b
	p				

10

b	0	0	0	0	b
p					

11

b	0	0	0	0	b
	r				

12

b	1	0	0	0	b
		r			

13

b	1	1	0	0	b
			r		

14

b	1	1	1	0	b
				r	

15

b	1	1	1	1	b
					r

16

b	1	1	1	1	b
					s

Ejemplo 2. Complemento binario

- Esta MT obtiene el complemento binario de un número binario almacenado en la cinta.

$$MT = \{ \Gamma, \sum, b, Q, q_0, f, F \}$$

$$MT_2 = (\{ 0, 1, b \}, \{ 0, 1 \}, b, \{ q_0, q_1 \}, q_0, f, \{ q_1 \})$$

f	0	1	b
q_0	$q_0 1D$	$q_0 0D$	$q_1 bI$

Ejercicio: realiza la simulación de la ejecución de la MT, proporciona una palabra de entrada y verifica que la salida sea correcta.

Ejemplo 3. Paridad

- Esta MT analiza la cadena de entrada: si el número de unos es par añade un 0, en otro caso, si es impar, añade un 1 a la cinta. EL 0 o el 1 se añaden en la cinta, al final de la cadena leída.

$$MT = \{ \Gamma, \sum, b, Q, q_0, f, F \}$$

$$MT_2 = (\{ 0, 1, b \}, \{ 0, 1 \}, b, \{ q_0, q_1, q_2 \}, q_0, f, \{ q_2 \})$$

f	0	1	b
q_0	$q_0 0D$	$q_1 1D$	$q_2 0I$
q_1	$q_1 0D$	$q_0 1D$	$q_2 1I$

Ejercicio: realiza la simulación de la ejecución de la MT, proporciona una palabra de entrada y verifica que la salida sea correcta.

Ejercicio 1. Imagen espejo

- Diseña una MT que lea una palabra y devuelva su imagen espejo, por ejemplo:

  Entrada: 100 Salida: 100001

  Entrada: 001 Salida: 001100

Punto a Realizar:

- Obtener la MT
- Hacer la simulación de la ejecución de la MT: proporciona una palabra de entrada y verifica que la salida sea correcta.

Ejercicio 2. Verificador de paréntesis anidados

- Diseña una MT que lea una cadena que incluya paréntesis, y que haga la indicación si es correcta (1) o errónea (0), por ejemplo:

  Entrada: ((())) Salida: 1

  Entrada: (()))(Salida: 0

Nota: puedes idear algún otro tipo de salida que indique si la entrada es o no es correcta

Punto a Realizar:

- Obtener la MT
- Hacer la simulación de la ejecución de la MT: proporciona una palabra de entrada y verifica que la salida sea correcta.

Ejercicio 3. Verificador de palíndromos

- Diseña una MT que lea una cadena y que haga la indicación si es un palíndromo (1) o no lo es (0), por ejemplo:

  Entrada: 1001   Salida: 1

  Entrada: 10100   Salida: 0

Nota: puedes idear algún otro tipo de salida que indique si la entrada es o no es correcta

Punto a Realizar:

- Obtener la MT
- Hacer la simulación de la ejecución de la MT: proporciona una palabra de entrada y verifica que la salida sea correcta.

Restricciones a la MT

- La MT que se ha definido anteriormente es la más genérica posible. Sin embargo, veremos que se le pueden imponer restricciones a la definición sin que esto afecte a la potencia computacional de la máquina. Las restricciones se aplicarán a:
- El alfabeto
- La estructura de la cinta
- La capacidad de la máquina para realizar diferentes operaciones (escribir, desplazarse o cambiar d estado) en una sola operación.

MT con alfabeto binario

- Cualquier MT es equivalente (realiza la misma tarea) a una MT con un alfabeto binario **Γ= {0, 1, b}**. Para conseguirlo, será necesario codificar en binario los caracteres del alfabeto original.
- Cada transición original se desglosará en varias transiciones de la máquina con alfabeto binario ocn el consiguiente incremento de estados intermedios, por ejemplo:

MT con alfabeto binario

Sea Z una M.T. definida sobre el alfabeto $\Sigma = \{x, y, z, w\}$. Supongamos que Z' es la M.T. definida sobre un alfabeto binario, equivalente a Z, que pretendemos construir. Codificamos los símbolos de Σ de la siguiente forma:

$x = 00 \qquad y = 10 \qquad z = 01 \qquad w = 11$

Supongamos que $f(p, x) = (q, y, D)$ es una de las transiciones de Z. En la máquina Z' esta transición se desglosaría en las siguientes:

$f'(p, 0) = (p_0, 0, D)$ reconoce el primer 0 de x
$f'(p_0, 0) = (p_x, 0, I)$ reconoce el segundo 0 de x
$f'(p_x, 0) = (p_{x0}, 1, D)$ cambia 00 por 10 (x por y)
$f'(p_{x0}, 0) = (q, 0, D)$ se desplaza a la derecha y pasa al estado q

Para conseguir en Z' transiciones equivalentes a la de Z ha sido necesario utilizar tres nuevos estados p_0, p_x, p_{x0}.

MT con la cinta limitada en un sentido

- Dada una MT con una cinta infinita en ambos sentidos, siempre existe una máquina equivalente cuya cinta está limitada por un extremo pero es infinita por el otro.
- Sea Z una MT con una cinta infinita en ambos sentidos. Se pueden numerar las casillas de la siguiente forma:

-2	-1	1	2	

Es posible construir una M.T. equivalente Z' cuya cinta tendrá la siguiente estructura:

*				
0	1	-1	2	-2

- En Z' existe una nueva casilla, numerada con el 0, que contiene un símbolo nuevo (*). Además de esta nueva casilla, se ha redistribuido la información de Z, de manera que cada casilla de Z' contiene la misma información que su correspondiente casilla en Z.

MT con restricciones en cuanto a las operaciones que realiza simultáneamente

- En el modelo original de MT, se realizan tres operaciones en cada transición:
  - Escritura de un símbolo
  - Cambio de estado
  - Movimiento de la cabeza de lectura/escritura

MT con restricciones en cuanto a las operaciones que realiza simultáneamente

Veamos, con diferentes ejemplos, como, aumentando el número de estados, se puede restringir el número de operaciones que se realizarán simultáneamente.

Imposibilidad para escribir y cambiar de estado simultáneamente

La transición $f(p, x) = (q, y, D)$ se convierte en:

$$f'(p, x) = (p_{xD}, x, P)$$
$$f'(p_{xD}, x) = (p_{xD}, y, P)$$
$$f'(p_{xD}, y) = (q, y, D)$$

Imposibilidad para escribir y desplazarse simultáneamente

La transición $f(p, x) = (q, y, D)$ se convierte en:

$$f'(p, x) = (p_D, y, P)$$
$$f'(p_D, y) = (q, y, D)$$

De forma análoga, dada una M.T. es posible construir otra equivalente a ella que ejecute una sola operación en cada transición.

Modificaciones de la MT

- Una de las razones de la gran aceptación de la MT como modelo general de computación es que el modelo estándar definido anteriormente es equivalente a otras versiones del MT que, sin aumentar el poder de cómputo del dispositivo, permiten resolver con más facilidad determinados problemas.

Modificaciones de la MT – Almacenamiento de información en el control finito.

- Es posible utilizar el estado de control (perteneciente a un conjunto finito) para almacenar una cantidad finita de información. Para ello cada estado se presenta como un par ordenado donde el primer elemento representa realmente al estado y el segundo a la información que se pretende almacenar.

Modificaciones de la MT – Almacenamiento de información en el control finito.

Veamos su utilidad con un ejemplo en el que se define una M.T. que reconoce el lenguaje $L = 01^* + 10^*$. Como el primer símbolo de la cadena no puede volver a aparecer, se almacena con el estado de control. Las transiciones no definidas son situaciones de error, es decir, la cadena no ha sido reconocida.

$Q = \{q_0, q_1\} * \{0, 1, b\}$ estado inicial $= [q_0, b]$ $F = \{[q_1, b]\}$

$\Sigma = \{0, 1\}$ $\Gamma = \{0, 1, b\}$

	0	1	b	
$[q_0, b]$	$[q_1, 0]0D$	$[q_1, 1]1D$		
$[q_1, 0]$		$[q_1, 0]1D$	$[q_1, b]0P$	este estado indica que la cadena comienza pc
$[q_1, 1]$	$[q_1, 1]0D$		$[q_1, b]0P$	este estado indica que la cadena comienza pc

Es evidente que no se aumenta la potencia computacional del modelo ya que simplemente se ha utilizado una notación diferente para designar a los estados.

Modificaciones de la MT – Pistas múltiples

- En este caso se considera que la cinta está dividida en un número k de pistas, de manera que los símbolos de la cinta se representan como k-tuplas.
- Tampoco en este caso se aumenta la potencia del modelo ya que sólo hay un cambio en la representación de los símbolos de la MT.
- Esta variación de la MT, concretamente con tres pistas, puede resultar muy útil para resolver operaciones aritméticas con dos datos. Cada dato se almacena en una pista y el resultado se almacena en la tercera (que inicialmente está vacía).
- Por ejemplo, una MT que sume números binarios comenzaría con la siguiente información en la cinta:

Modificaciones de la MT – Pistas múltiples

	b	1	1	0	b
	b	1	1	1	b
	b	b	b	b	b

y finalmente la cinta quedaría así:

	b	1	1	0	b
	b	1	1	1	b
b	1	1	0	1	b

Ejemplos de transiciones para esta máquina:

$$f(q_0, \begin{bmatrix} 0 \\ 1 \\ b \end{bmatrix}) = (q_0, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, I) \qquad f(q_0, \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix}) = (q_0, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, I)$$

Modificaciones de la MT – Máquinas multicinta

- En este caso la MT dispone de k cintas y k cabezas de lectura/escritura.
- Cada cabeza trabaja (lee/escribe) sobre su cinta y se mueve de forma independiente a las demás.
- Para comprobar que este tipo de máquinas no aumenta el poder computacional de la MT estándar se puede construir una máquina equivalente a ella utilizando varias pistas y símbolos de chequeo (para llevar la pista de los símbolos procesados) .
- Supongamos que la máquina Z dispone de tres cintas y una situación concreta puede ser representada como se indica en el siguiente esquema:

		A_i			
		$\uparrow p$			
				B_j	
				$\uparrow q$	
			C_k		
			$\uparrow r$		

Es posible construir una nueva máquina *multipista* que sea equivalente a ella. Esta nueva máquina tendrá un número de pistas igual al doble de las cintas de la máquina original, en este caso seis. Las pistas pares almacenan un símbolo de chequeo que sirve para indicar qué celda de la pista inmediatamente superior se está procesando en ese momento. Las pistas impares tienen la misma función que las correspondientes cintas de la máquina original. Evidentemente, esta nueva M.T. tendrá más estados porque tiene que ir procesando secuencialmente cada pareja de pistas buscado el símbolo de chequeo correspondiente.

		A_i			
		*			
				B_j	
				*	
			C_k		
			*		

MT Universal

- El concepto de MT es tan general y potente que es posible construir una MT que sea capaz de simular el comportamiento de cualquier otra MT, a esta máquina se la llama **Máquina de Turing Universal**, y su forma de trabajar puede compararse con la de un ordenador que se comporta de una u otra forma dependiendo del programa que ejecuta en cada momento.
- Veremos cómo se podría representar toda la información necesaria para simular el comportamiento de una MTU llamada M.
- Es necesario conocer: f, q0, contenido inicial de la cinta y la posición de la cabeza de lectura/escritura.
- Sin pérdida de generalidad supondremos que M trabaja con un alfabeto binario y que necesitamos m bits para codificar cada uno de sus estados. El movimiento de la cabeza se puede codificar con un único bit (0=derecha;1=izquierda)

MT Universal

- Así, cada transición f(p, a) = (q, b, D) puede codificarse con 2m+3 símbolos binarios de la forma:

$\underbrace{xx\ldots x}_{m}\,a\,\underbrace{yy\ldots y}_{m}\,b0$, donde $xx\ldots x$ es una codificación binaria del estado p y $yy\ldots y$ representa al estado q.

La M.T. **M** puede representarse de la siguiente forma:

$$dd\ldots * \ldots d < q\ldots qd > xx\ldots xayy\ldots yb0 > xx\ldots xayy\ldots yb0 > \ldots$$

Los símbolos $<$ y $>$ permiten delimitar los diferentes bloques que constituyen esta cadena. El primer bloque($dd\ldots * \ldots d$) representa el contenido de la cinta de la M.T. **M**, y el asterisco se coloca inmediatamente a la izquierda del símbolo al que apunta la cabeza de lectura/escritura. Esta información va cambiando a lo largo del proceso representando en cada momento el contenido de la cinta de **M**. El segundo bloque representa el estado actual de **M** ($q\ldots q$) y el símbolo al que apunta la cabeza (d), que es el que está a la derecha del asterisco en el bloque anterior. Esta información tiene longitud $m+1$ y también cambia a lo largo del proceso. A partir de este punto, el resto de los bloques (de longitud $2m+3$) representan a los diferentes valores de la función de transición.

MT Universal

$$dd\ldots*\ldots d < q\ldots qd > xx\ldots xayy\ldots yb0 > xx\ldots xayy\ldots yb0 > \ldots$$

Cada transición puede considerarse como un registro formado por dos partes: la primera ($xx\ldots xa$) es una *etiqueta* que identifica a la transición, la segunda ($yy\ldots yb0$) puede ser considerada como el *dato* de la transición. Por lo tanto, esta última parte de la cadena ($> xx\ldots xayy\ldots yb0 > \ldots$) puede ser vista como una colección de registros formados cada uno de ellos por una *etiqueta* y un *dato*. Con estas consideraciones, será necesario construir una M.T. `localizadora de información` cuya misión sea buscar en esta colección de registros aquél que tenga una etiqueta determinada. Es evidente que, conociendo el significado de los datos que hay en la cinta, el segundo bloque de información ($< q\ldots qd >$) constituye la etiqueta que es necesario buscar. Una vez localizado el registro (transición) adecuado será necesario copiar el dato ($yy\ldots yb$) en el segundo bloque de la cinta, y modificar adecuadamente el primer bloque. Básicamente, la M.T. Universal está constituida por una M.T. que localiza información en la cinta y otra que copia información de una parte a otra de la cinta.

La tesis Church-Turing

Podemos decir que una M.T. es un modelo general de computación y esto es equivalente a decir que cualquier *procedimiento algorítmico* que sea ejecutable (por una persona o una máquina) puede ser desarrollado por una M.T.

"La noción de procedimiento algorítmico que actúa sobre una secuencia de símbolos es idéntica al concepto de un proceso que puede ser ejecutado por una M.T." Esta afirmación la formuló el lógico Alonzo Church a principios de la década de los 30 y suele denominarse **tesis de Church** o **tesis de Church-Turing**. No es una afirmación matemática exacta, ya que carecemos de una definición precisa para el término *procedimiento algorítmico* y, por lo tanto, no es algo que pueda comprobarse. Sin embargo esta tesis es aceptada de forma general, debido a diferentes motivos:

La tesis Church-Turing

1. No se ha planteado ningún tipo de computo que pueda incluirse en la categoría de *procedimiento algorítmico* y que no pueda ejecutarse en una M.T.

2. Se han propuesto mejoras al diseño de la M.T. original y en todos los casos ha sido posible demostrar que el poder computacional de las M.T.'s no se veía modificado.

3. Se han propuesto otros modelos teóricos de cómputo, que siempre son equivalentes al de las M.T.'s

La tesis de Church-Turing no puede probarse de manera precisa debido justamente a la imprecisión del término *proceso algorítmico*. Sin embargo, una vez adoptada esta tesis, ya podemos darle un significado preciso al término: "un algoritmo es un procedimiento que puede ser ejecutado por una M.T." Esta definición nos proporciona un punto de partida para analizar problemas que pueden o no resolverse con una M.T. como veremos en los siguientes temas.