



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Departamento de Estadística, I.O. y Computación
Teoría de Automatas y Lenguajes Formales

PRACTICA 10: Máquinas de Turing

10.1. Introducción

La clase de autómatas que hoy conocemos como máquinas de Turing fue propuesta en 1936 por el matemático Alan Turing. La pretensión de Turing era el estudio de los procesos algorítmicos usando un modelo computacional. Podemos considerar las máquinas de Turing como una generalización de los autómatas que hemos estudiado en temas anteriores. Las máquinas de Turing (TM, por Turing Machine) son similares a los autómatas finitos en tanto en cuanto poseen un mecanismo de control y un flujo de entrada que se representa como una cinta (ver Figura 10.1). La diferencia fundamental es que una TM puede mover su cabeza de lectura hacia delante y hacia atrás sobre la cinta de entrada, y puede tanto leer como escribir en dicha cinta.

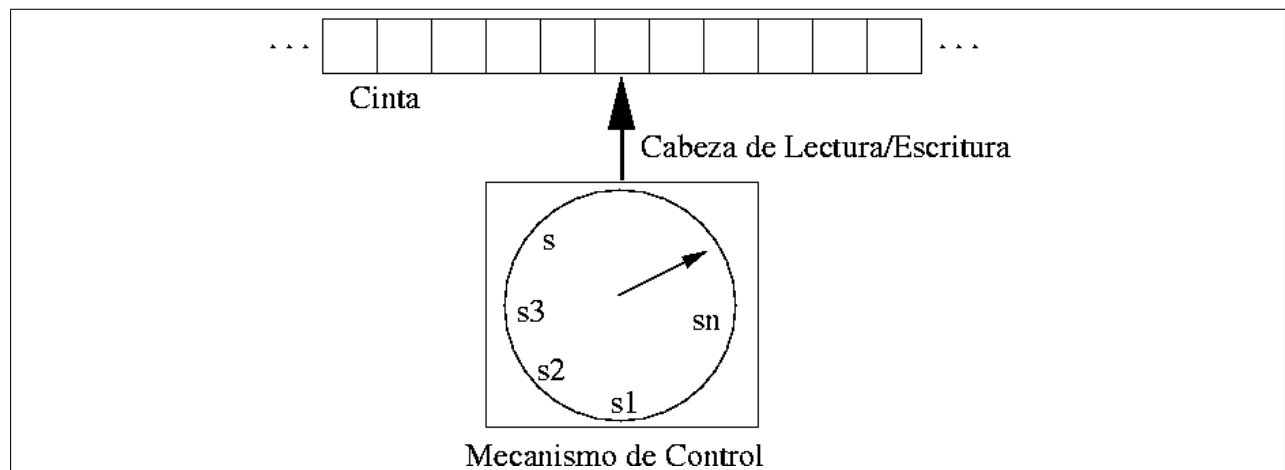


Figura 10.1: Conceptualización de una Máquina de Turing

Una TM contiene un mecanismo de control que en cualquier momento puede encontrarse en uno de entre un número finito de estados. Uno de estos estados se denomina estado inicial y representa el estado en que la máquina comienza sus cálculos. Otro estado distinguido es el estado de parada. Una vez que la TM alcanza ese estado, finalizan todos los cálculos. El estado inicial y el de parada han de ser distintos, y por tanto, toda TM ha de tener al menos dos estados.

La TM dispone de una cabeza que puede leer y escribir símbolos en la cinta de la máquina. Esta cinta se supone que se extiende indefinidamente hacia derecha e izquier-

da. Una TM utiliza su cinta como dispositivo de almacenamiento auxiliar y puede no sólo insertar o extraer símbolos sino también rastrear los datos de la cinta y modificar las celdas que desee sin alterar el resto.

Es conveniente que la TM utilice marcas especiales para distinguir partes de la cinta. Con este fin, la TM utiliza símbolos que no aparecen en los datos de entrada, es decir, se distingue entre el conjunto finito de símbolos llamado el alfabeto de la máquina, en el que deben estar codificados los datos de entrada iniciales y otro conjunto también finito de símbolos de cinta posiblemente mayor que el anterior llamado símbolos de cinta, que son los símbolos que la máquina puede leer y escribir. De este modo los símbolos de cinta de una TM pueden incluir marcas especiales que no sean símbolos del alfabeto de la máquina. El espacio en blanco es uno de los símbolos de esta clase: se supone que está en cualquier celda de la cinta que no esté ocupada. Por ejemplo, si una TM leyera más a la derecha de los símbolos de entrada de su cinta, encontraría y leería las celdas en blanco que allí encontraría. También puede ser necesario que una TM borre una celda de la cinta, escribiendo en ella un espacio en blanco. Por estos motivos se considera que el espacio en blanco forma parte de los símbolos de cinta, pero no del alfabeto de la máquina. Para que el símbolo de espacio en blanco no nos ocasione problemas en las representaciones adoptaremos el símbolo \$ para representar el espacio en blanco. Este convenio elimina la ambigüedad entre las cadenas $a\ b$ y $a\ b$ porque representaremos la primera como $a\$b$ y la segunda como $a\$\b .

En cada paso de un cómputo una TM lee un símbolo de la cinta de entrada, lo reemplaza por un nuevo símbolo (que puede ser el que encontró originalmente) y a continuación cambia su estado (el nuevo estado puede ser el mismo en que se encontraba antes de la operación) y realiza un movimiento de la cabeza de lectura/escritura hacia la izquierda o a la derecha. La acción que en un determinado momento del cómputo realizará la TM dependerá del símbolo (símbolo actual) que esté visible en ese momento en la celda (celda actual) en que está posicionada la cabeza de lectura/escritura, y del estado actual del mecanismo de control de la máquina.

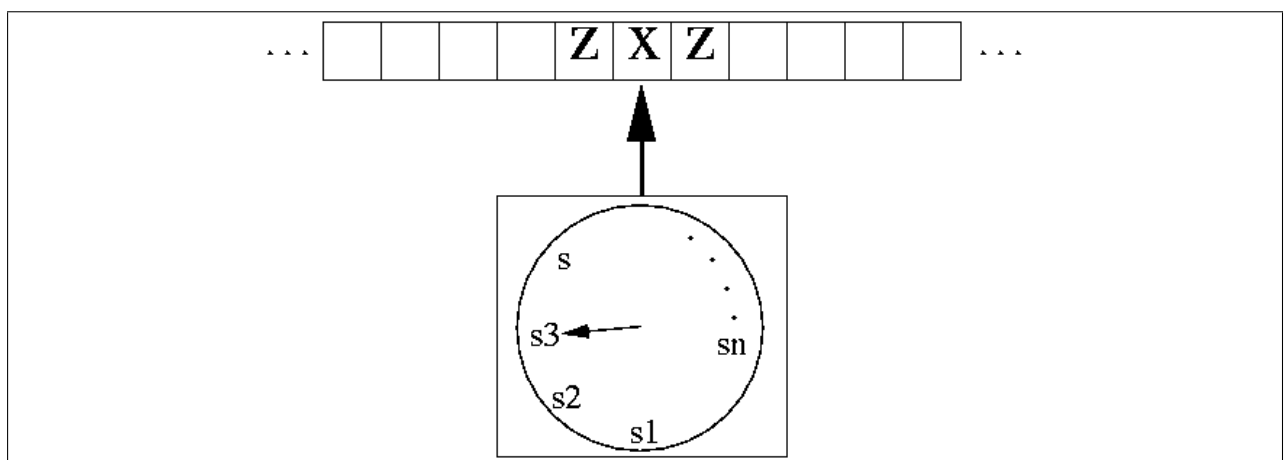


Figura 10.2: Un estado de una TM

Una TM se puede definir formalmente como una 7-upla $TM \equiv (Q, \Sigma, \Gamma, s, \$, F, \delta)$

donde el significado de cada uno de los elementos es:

- Q es el conjunto finito de estados de la máquina
- Σ es el alfabeto de entrada
- Γ es el alfabeto de la cinta
- $s \in Q$ es el estado inicial
- $\$ \in \Gamma$ es el símbolo “espacio en blanco”. $\$ \notin \Sigma$
- $F \subseteq Q$ es el conjunto de estados finales o de aceptación
- δ Es la función (parcial) de transición $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

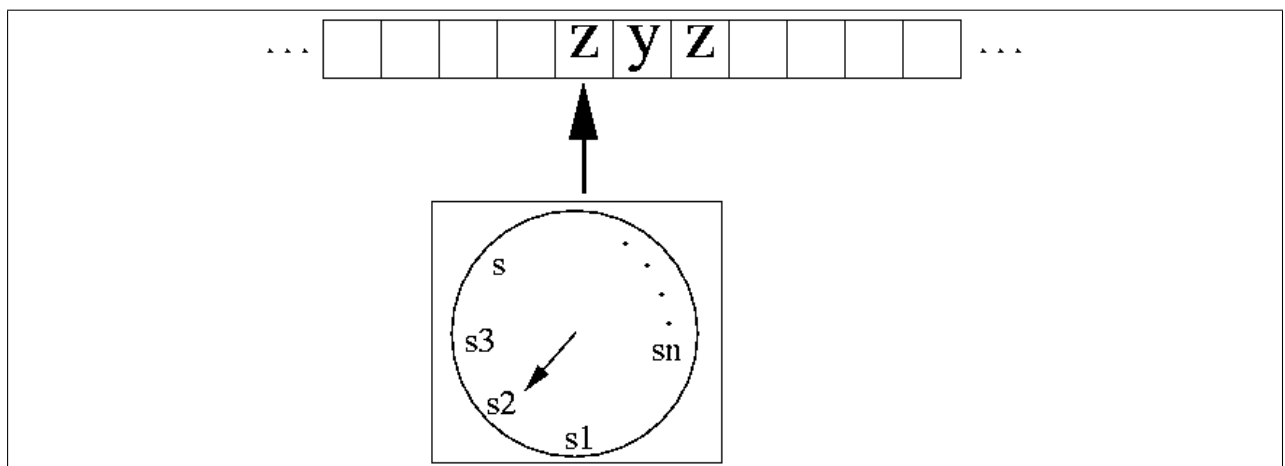


Figura 10.3: El nuevo estado después de la transición $\delta(s3, x) = (s2, y, L)$

Generalmente se permite que $\Sigma \subseteq \Gamma - \{\$\}$, es decir que el alfabeto de entrada sea un subconjunto del de pila, pero el símbolo blanco no puede formar parte del alfabeto de entrada. La función de transición transforma pares (q, σ) formados por el estado actual, q y el símbolo σ en que está posicionada la cabeza de lectura/escritura en triples (p, t, X) donde p es el estado siguiente, $t \in \Gamma$ es el símbolo que se escribe en la cinta y $X \in \{L, R\}$ es un movimiento de la cabeza de lectura escritura que puede ser a la izquierda (L) o a la derecha (R). Por ejemplo, una transición como $\delta(s3, x) = (s2, y, L)$ provoca que la TM pase de una configuración como la que se muestra en la Figura 10.2 a la que presenta la Figura 10.3.

La función de transición, δ de una TM es una función parcial, lo cual quiere decir que no tiene porqué estar definida para todos los pares (q, σ) del conjunto de partida.

Nos interesará disponer de una notación concisa que represente la configuración de una TM, incluyendo el contenido de las celdas de la cinta, la posición de la cabeza de lectura/escritura y el estado de la máquina. Para representar las configuraciones utilizaremos pares de la forma $(q_i, w_1 \sigma w_2)$ donde q_i es el estado en que se encuentra la máquina, σ es el símbolo presente en la celda en que está posicionada la cabeza de lectura

escritura, y w_1 y w_2 son respectivamente las cadenas de símbolos que anteceden y siguen a σ sobre la cinta. A estas configuraciones se les llama descripciones instantáneas (DI). El paso de una DI a otra lo denotaremos con el símbolo \vdash , y los símbolos \vdash^* y \vdash^+ tienen el significado habitual de “cero o más” y “uno o más” pasos respectivamente.

10.2. Ejemplo 1

Consideremos como primer ejemplo la TM definida por:

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, \$\}$
- $s = q_1$
- $F = \{q_2\}$

y con la siguiente función de transición δ :

- $\delta(q_1, a) = (q_1, a, R)$
- $\delta(q_1, b) = (q_1, a, R)$
- $\delta(q_1, \$) = (q_2, \$, L)$

La máquina comienza a funcionar estando en el estado q_1 . Supongamos que la cabeza de lectura/escritura se encuentra situada sobre un símbolo 'a'. En ese caso, la transición que se aplicaría es: $\delta(q_1, a) = (q_1, a, R)$; es decir, la máquina sobrescribirá el símbolo que encuentra sobre la cinta, se moverá una posición a la derecha y permanecerá en el mismo estado. Esto es lo que ocurre tanto con la primera 'a' como con todas las sucesivas que la máquina encuentre sobre la cinta. Si eventualmente se encuentra una 'b', el comportamiento es análogo, con la diferencia que la máquina sobrescribe la 'b' con una 'a'. Si la máquina encuentra un blanco, se moverá hacia la izquierda y alcanzará el estado final q_2 . Dado que no hay transiciones desde el estado final, la máquina parará una vez llega a ese estado.

Veamos un ejemplo de cómputo de esta máquina para la frase de entrada *abba*:

$(q_1, \underline{abba}) \vdash (q_1, \underline{abba}) \vdash (q_1, \underline{aaba}) \vdash (q_1, \underline{aaaa}) \vdash (q_1, \underline{aaaa\$}) \vdash (q_2, \underline{aaaa})$

Como vemos, esta TM transforma su entrada que es una cadena de aes y bes de un determinado tamaño en otra cadena del mismo tamaño formada sólo por símbolos 'a'.

Cuando $\delta(q, \sigma)$ no está definida y la configuración de la TM es $(q, w_1 \underline{\sigma} w_2)$ es imposible que pase a otra configuración. En este caso se dice que la TM está parada. En este caso puede ocurrir que q sea un estado final ($q \in F$) o que no. En cualquier caso interesa dotar de significado a la parada en un estado final. Para simplificar, supondremos que no se definirá ninguna transición para los estados de F , por lo que la TM siempre parará al alcanzar un estado final.

Se llama computación a la secuencia de movimientos que conducen a una configuración de parada.

10.3. Ejemplo 2

Consideremos ahora la TM definida por:

- $Q = \{q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, \$\}$
- $s = q_1$
- $F = \emptyset$

y δ definida como :

- $\delta(q_1, a) = (q_2, a, R)$
- $\delta(q_1, b) = (q_2, b, R)$
- $\delta(q_1, \$) = (q_2, \$, R)$
- $\delta(q_2, a) = (q_1, a, L)$
- $\delta(q_2, b) = (q_1, b, L)$
- $\delta(q_2, \$) = (q_1, \$, L)$

Si la máquina comienza a funcionar con la cabeza situada en el primer símbolo (la a) de una cadena de la forma abw realizará la siguiente secuencia de movimientos:

$(q_1, \underline{a}bw) \vdash (q_1, a\underline{b}w) \vdash (q_1, ab\underline{w}) \vdash (q_1, abw\underline{}) \vdash \dots$

y se moverá por tiempo indefinido desplazando la cabeza de lectura/escritura de izquierda a derecha alternativamente. Este es un ejemplo de TM que nunca parará. Podríamos decir que la máquina entra en un “bucle infinito”. Esta situación es fundamental en la teoría de las TM y la representaremos $(q_1, w_1\sigma w_2) \vdash^* \infty$ indicando que la máquina, partiendo de la configuración inicial $(q_1, w_1\underline{\sigma}w_2)$ nunca parará.

10.4. Ejemplo 3

Consideremos a continuación una TM que acepte el lenguaje $L = \{0^n 1^n \mid n \geq 1\}$. Inicialmente la máquina contendrá una secuencia de símbolos $0^n 1^n$ seguida por un número infinito de $\$$. Sucesivamente, la máquina reemplazará el 0 más a la izquierda en la cinta por un símbolo X, se moverá hacia la derecha hasta el 1 situado más a la izquierda en

la cinta reemplazándolo por un símbolo Y, se moverá hacia la izquierda hasta encontrar la X situada más a la derecha, a continuación se mueve una celda a la derecha para alcanzar el 0 situado más a la izquierda y repetirá el ciclo. Si la máquina encuentra un \$ cuando busca un 1, la máquina parará sin aceptar la cadena. Si después de cambiar un 1 por una Y la máquina no encuentra más ceros, comprueba que no hay más unos, aceptando la cadena si no los hay.

El alfabeto de entrada será $\Sigma = \{0, 1\}$, el de cinta $\Gamma = \{0, 1, X, Y, \$\}$ y el conjunto de estados de la máquina será $Q = \{q_0, q_1, q_2, q_3, q_4\}$. El estado q_0 es el inicial y también se transita a él antes de reemplazar un cero situado más a la izquierda por una X. El estado q_1 se utiliza para buscar hacia la derecha ignorando ceros y símbolos Y, hasta encontrar el 1 situado más a la izquierda. Si la TM halla un 1, lo cambia por Y pasando al estado q_2 . El estado q_2 se usa para buscar un símbolo X hacia la izquierda. Cuando lo encuentra, la máquina pasa al estado q_0 y se mueve a la derecha para colocarse sobre el 0 situado más a la izquierda. Cuando la máquina busca hacia la derecha en el estado q_1 rechaza la entrada si encuentra un \$ o una X antes que un 1: en este caso o bien hay demasiados ceros o bien la entrada no es de la forma 0^*1^* .

	Símbolo				
Estado	0	1	X	Y	\$
q_0	(q_1, X, R)	-	-	(q_3, Y, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-

Cuadro 10.1: La función de transición de la máquina que reconoce 0^n1^n

El estado q_0 juega también otro papel: si después de que en el estado q_2 se encuentre la X situada más a la derecha hay una Y situada inmediatamente a la derecha, entonces es que los ceros se han acabado. En este caso (estando en q_0 y viendo una Y) se transita al estado q_3 para leer todas las Y y comprobar que no queda ningún 1. Si los símbolos Y están seguidos por un \$ se pasa al estado de aceptación q_4 . En otro caso, la cadena es rechazada. La Figura 10.1 muestra la función de transición para esta máquina.

Veamos un ejemplo de computación con esta máquina, para la cadena de entrada 0011:

$(q_0, \underline{0011}\$) \vdash (q_1, X\underline{011}\$) \vdash (q_1, X0\underline{11}\$) \vdash (q_2, X\underline{0Y1}\$) \vdash (q_2, X\underline{0Y1}\$) \vdash (q_0, X\underline{0Y1}\$) \vdash$
 $(q_1, X\underline{XY1}\$) \vdash (q_1, X\underline{XY1}\$) \vdash (q_2, X\underline{XY\underline{Y}}\$) \vdash (q_2, X\underline{XY\underline{Y}}\$) \vdash (q_0, X\underline{XY\underline{Y}}\$) \vdash (q_3, X\underline{XY\underline{Y}}\$) \vdash$
 $(q_3, X\underline{XY\underline{Y}}\$) \vdash (q_4, X\underline{XY\underline{Y}}\$\$)$

10.5. Simulador de Máquina de Turing

El objetivo de la práctica es diseñar en C++ un programa simulador de máquinas de Turing. El programa deberá ser capaz de leer la tabla de transición de una máquina, así como el contenido inicial de la cinta de entrada, y simular el comportamiento de cada una de las acciones que efectúe dicha máquina.

La tabla de transición de la máquina a simular y su cinta de entrada deben estar en dos ficheros diferentes, y sus nombres se capturarán en la línea de comandos, mostrando un texto explicativo de cómo se han de introducir los parámetros en caso de que el usuario no los introduzca de forma correcta. Los ficheros de transición para las diferentes máquinas tendrán la extensión `.tm`, (Turing Machine) mientras que los ficheros de cintas de entrada tendrán extensión `.tmt` (Turing Machine Tape). El simulador ignorará de los ficheros de definición de máquinas (TM) todas las líneas que comiencen con los caracteres `//`. Esto permitirá colocar comentarios a dichos ficheros, explicando la máquina de Turing que representan.

Los símbolos del alfabeto podrán ser cualquier carácter alfanumérico. Los estados son enteros comenzando con estado inicial igual a cero.

10.5.1. Ficheros de especificación de máquinas de Turing

El fichero que contiene a la tabla de transición de la máquina debe tener la extensión `.tm` y seguir el siguiente formato:

Posiblemente una serie de líneas que comienzan con `//` y que serán ignoradas por el simulador a la hora de leer la máquina. Después de estas líneas (que son opcionales) vendría la siguiente información:

- Línea 1: Entero indicando el número de estados
- Línea 2: Entero indicando el primer estado final
- Líneas sucesivas: una tupla en cada línea

Los estados con número mayor o igual que el estado final se consideran también estados finales.

El formato de las tuplas es el siguiente:

`estado$entrada$escribe$movimiento$siguiente_estado`

- `$`: Espacio en blanco (' ').
- `estado`: Entero que representa el estado
- `entrada`: Símbolo del alfabeto de entrada
- `escribe`: Elemento del alfabeto de cinta a escribir en la cinta para dicho estado y entrada.
- `movimiento`: Movimiento a realizar por la cabeza de lectura:
 - L - Izquierda
 - R - Derecha
 - S - Parar
- `siguiente_estado`: Siguiendo estado al que transita la máquina.

En la sección de descargas de las páginas de la asignatura pueden obtenerse ejemplos de ficheros de definición de máquinas de Turing, así como algunos ficheros con entradas para dichas máquinas.