Lenguajes Formales y Autómatas Maestría
Notas del Curso

José de Jesús Lavalle Martínez

Otoño de 2006

Índice

1. Marcar símbolos **1**

2. Desplazamiento **2**

3. La máquina de Turing como un procedimiento **3**

4. Modificaciones a la máquina de Turing **4**
4.1. Cinta infinita por ambos lados 4
4.2. Múltiples cintas 6

5. La clase de conjuntos recursivos **7**

6. Máquinas de Turing y gramáticas de tipo 0 **8**

7. Autómatas acotados linealmente y lenguajes sensibles al contexto **9**

1. Marcar símbolos

Considere una máquina de Turing $T = (K, \Sigma, \Gamma, \delta, q_0, F)$, donde:

$$
\begin{array}{rcl}
K & = & \{[q, d] | q = q_1, \ldots, q_9 \text{ y } d = a, b, B\}, \\
\Sigma & = & \{[B, d] | d = a, b, c\}, \\
\Gamma & = & \{[X, d] | X = B, \surd \text{ y } d = a, b, c, B\}, \\
q_0 & = & [q_1, B] \text{ y} \\
F & = & [q_9, B]
\end{array}
$$

La segunda componente del estado se usa para almacenar un símbolo de entrada. $[B, B]$ se toma como el símbolo blanco, $[B, a]$ como a, $[B, b]$ como b y

$[B, c]$ como c. La función de transición δ se difine como (en la descripción de la función de transición considere que $d = a, b$ y que $e = a, b$):

1. $\delta([q_1, B], [B, d]) = ([q_2, d], [\surd, d], R)$
2. $\delta([q_2, d], [B, e]) = ([q_2, d], [B, e], R)$
3. $\delta([q_2, d], [B, c]) = ([q_3, d], [B, c], R)$
4. $\delta([q_3, d], [\surd, e]) = ([q_3, d], [\surd, e], R)$
5. $\delta([q_3, d], [B, d]) = ([q_4, B], [\surd, d], L)$
6. $\delta([q_4, B], [\surd, d]) = ([q_4, B], [\surd, d], L)$
7. $\delta([q_4, B], [B, c]) = ([q_5, B], [B, c], L)$
8. $\delta([q_5, B], [B, d]) = ([q_6, B], [B, d], L)$
9. $\delta([q_6, B], [B, d]) = ([q_6, B], [B, d], L)$
10. $\delta([q_6, B], [\surd, d]) = ([q_1, B], [\surd, d], R)$
11. $\delta([q_5, B], [\surd, d]) = ([q_7, B], [\surd, d], R)$
12. $\delta([q_7, B], [B, c]) = ([q_8, B], [B, c], R)$
13. $\delta([q_8, B], [\surd, d]) = ([q_8, B], [\surd, d], R)$
14. $\delta([q_8, B], [B, B]) = ([q_9, B], [\surd, B], L)$

Analice la máquina de Turing anterior y diga qué lenguaje reconoce.

2. Desplazamiento

Suponga que tiene una máquina de Turing $T = (K, \Sigma, \Gamma, \delta, q_0, F)$ pero que no sabe lo que hace (como es usual), es decir, no tiene la descripción completa, sólo sabe que: K contiene estados de la forma $[q, A_1, A_2]$ para $q = q_1, q_2$; A_1 y A_2 están en Γ; B es el símbolo blanco; X es un símbolo especial que sólo usa T en este proceso; que T "inicia" este proceso en el estado $[q_1, B, B]$ y que las porciones relevantes, a este proceso, de la función δ son (considere que A, A_1, A_2 y A_3 están en $\Gamma - \{B, X\}$):

1. $\delta([q_1, B, B], A_1) = ([q_1, B, A_1], X, R)$
2. $\delta([q_1, B, A_1], A_2) = ([q_1, A_1, A_2], X, R)$
3. $\delta([q_1, A_1, A_2], A_3) = ([q_1, A_2, A_3], A_1, R)$
4. $\delta([q_1, A_1, A_2], B) = ([q_1, A_2, B], A_1, R)$
5. $\delta([q_1, A_1, B], B) = ([q_2, B, B], A_1, L)$

6. $\delta([q_2, B, B], A) = ([q_2, B, B], A, L)$

Analice la porción dada de la máquina T y diga el efecto que tiene sobre la cinta.

3. La máquina de Turing como un procedimiento

Hasta aquí hemos definido la máquina de Turing como un dispositivo de reconocimiento, pero también podemos considerar la máquina de Turing como un procedimiento. Por ejemplo, si deseamos definir un procedimiento para determinar si un número es primo, podriamos construir una máquina de Turing que acepte precisamente el conjunto de todos los primos. Note que, en este caso, cuando la máquina de Turing se piensa como un reconocedor o como un procedimiento es meramente una cuestión de preferencia.

En general, si uno desea considerar un procedimiento para manipular cadenas de símbolos, uno puede convertir tal procedimiento a un problema de reconocimiento construyendo una nueva máquina de Turing que acepte pares de cadenas separadas por un símbolo especial. La nueva máquina de Turing acepta un par dado precisamente cuando el procedimiento convierta la primera cadena del par en la segunda cadena del par y luego pare.

Ejercicio 1 Demuestre que dada cualquier máquina de Turing que funcione como un procedimiento se puede construir una máquina de Turing que funcione como un reconocedor y viceversa.

La máquina de Turing que reconoce el lenguaje $L = \{0^n 1^n | n \geq 1\}$ se usa como un reconocedor. Note que, no importando cual sea la entrada, la máquina de Turing eventualmente alcanzará una condición en la que δ esté indefinida, para un estado del control finito y un símbolo leído por la cabeza de la cinta, en tal condición se dice que la máquina de turing para y que no hay más movimientos posibles. Si un lenguaje es aceptado por una máquina de Turing que para ante todas las entradas posibles se dice que el lenguaje es *recursivo*.

Se debe enfatizar que existen lenguajes que son aceptados por alguna máquina de Turing, pero que la máquina de Turing no para ante algunas entradas que no están en el lenguaje, esto se debe entender como la imposibilidad de construir una máquina de Turing que reconozca tales lenguajes y que pare ante todas las entradas posibles, no como el mero hecho de que algunas personas no la puedan construir. Un lenguaje que pueda ser reconocido por una máquina de Turing (aunque no pare ante todas la entradas posibles) se llama lenguaje *enumerable recursivamente*.

Cuando se considera la máquina de Turing como un procedimiento decimos que el procedimiento es un *algoritmo* si la máquina de Turing para ante cualquier entrada. Existen procedimientos para los cuales no existe un algoritmo correspondiente (justifique). Un ejemplo es un procedimiento para determinar si una gramática sensible al contexto (gsc) genera al menos una cadena terminal. Se puede construir una máquina de Turing que, dada la especificación de la gsc, genere todas las posibles cadenas terminales en algún orden lexicográfico. Para cada palabra la máquina de Turing aplica el algoritmo que construye su árbol de derivación (para ver si la palabra es generada por la gramática). Si la gramática genera al menos una palabra la máquina de Turing la encontrará (ya que habrá construido su árbol de derivación) y parará en un estado final. No obstante, si el lenguaje generado por la gsc es vacío la máquina de Turing seguirá generando palabras (tratando de construir su árbol de derivación) por siempre.

Además de considerar la máquina de Turing como un dispositivo reconocedor o como un procedimiento, podemos considerarla como que define una función. Sea $f(n)$ una función que mapea los enteros positivos en los enteros positivos. Sea una máquina de Turing $T = (K, \Sigma, \Gamma, \delta, q_0, F)$. Si, para todo entero n, $(q_0, 1^n, 1) \vdash (p, 1^{f(n)}, 1)$ para algún $p \in F$, entonces decimos que T *calcula* la función $f(n)$. Si alguna máquina de Turing T calcula $f(n)$ para cada n se dice que $f(n)$ es una función *recursiva*. Si $f(n)$ no está definida para todo n entonces se dice que es una función *parcial*. Si alguna máquina de Turing T calcula $f(n)$ siempre que $f(n)$ esté definida, pero no para para aquellos n para los que $f(n)$ esté indefinida, entonces se dice que $f(n)$ es una función *recursiva parcial*.

4. Modificaciones a la máquina de Turing

Una razón para aceptar a la máquina de Turing como un modelo general de cálculo es que el modelo que hemos estado trabajando es invariante a modificaciones que, a primera vista, podrían parecer incrementan su poder de cálculo. Por lo que en este apartado estudiaremos algunas de estas equivalencias.

4.1. Cinta infinita por ambos lados

Uma máquina de Turing con una cinta infinita por ambos lados se denota mediante $T = (K, \Sigma, \Gamma, \delta, q_0, F)$, como en el modelo original. Como su nombre implica, la cinta es infinita tanto a la izquierda como a la derecha. Denotamos una configuración de tal dispositivo mediante (q, α, i), donde q es el estado, α es la cadena de símbolos no blancos sobre la cinta, i es la posición de la cabeza de la cinta relativa al primer símbolo de α. Esto es, $i = 1$ si T está leyendo el símbolo más izquierdo de α, $i = 2$ si T está leyendo el segundo símbolo de α, etc. Imaginaremos, no obstante, que existen una infinidad de celdas blancas tanto a la izquierda como a la derecha de α. Así, es posible que $i = 0$, en cuyo caso T estará leyendo el blanco que está inmediatamente a la izquierda de α.

La relación $\vdash_T$, que relaciona dos configuraciones siempre que la configuración de la derecha se obtenga de la configuración de la izquierda mediante un solo movimiento, se define como en el modelo original con las siguientes excepciones para $i \leq 1$.

- Si $\delta(q, X) = (p, Y, L)$ entonces $(q, X\alpha, 1) \vdash_T (p, Y\alpha, 0)$,

- Si $\delta(q, B) = (p, Y, R)$ entonces $(q, \alpha, 0) \vdash_T (p, Y\alpha, 2)$,
- Si $\delta(q, B) = (p, Y, L)$ entonces $(q, \alpha, 0) \vdash_T (p, Y\alpha, 0)$

Aquí B es el blanco. Por supuesto $Y \neq B$. La configuración inicial es $(q_0, w, 1)$. A diferencia del modelo original, no existe final izquierdo para la máquina de Turing por lo cual se puede ir hacia la izquierda tanto como se desee. La relación $\vdash_T^*$, como es usual, relaciona dos configuraciones si la del lado derecho se puede obtener de la del lado izquierdo mediante cero o más movimientos.

Teorema 1 Si L es reconocido por una máquina de Turing con una cinta infinita por ambos lados entonces en reconocido por una máquina de Turing con una cinta infinita por la derecha (modelo original).

Prueba. Sea $T_2 = (K_2, \Sigma_2, \Gamma_2, \delta_2, q_{0_2}, F_2)$ una máquina de Turing con una cinta infinita por ambos lados. Construiremos una máquina de Turing T_1 que simula a T_2 y que sólo es infinita por la derecha. T_1 tendrá dos pistas, una para representar las celdas de la cinta de T_2 que están a la derecha de la celda leida inicialmente (incluyendola), la otra para representar, en orden inverso, las celdas a la izquierda de la celda inicial. Si numeramos con 0 la celda de T_2 leida inicialmente, con $1, 2, \ldots,$ las celdas a su derecha y con $-1, -2, \ldots,$ las celdas a su izquierda, entonces la relación entre las cintas T_2 y T_1 es:

$$\begin{gathered} \ldots, A_{-5}, A_{-4}, A_{-3}, A_{-2}, A_{-1}, A_0, A_1, A_2, A_3, A_4, A_5, \ldots \rightsquigarrow \\ A_0/¢, A_1/A_{-1}, A_2/A_{-2}, A_3/A_{-3}, A_4/A_{-4}, A_5/A_{-5}, \ldots \end{gathered}$$

La primer celda de la cinta de T_1 mantendrá, en la pista inferior, el símbolo ¢ para indicar que es la celda más izquierda. El control finito de T_1 mantendrá información (pista superior o pista inferior) para simular el símbolo que T_2 estaría leyendo.

Se puede construir T_1 para simular a T_2 en el sentido de que mientras T_2 está a la derecha de la posición inicial de la cabeza lectora, T_1 trabaja en la pista superior. Cuando T_2 está a la izquierda de la posición inicial de la cabeza lectora, T_1 trabaja en la pista inferior moviéndose en dirección opuesta a la que se mueve T_2. Los símbolos de entrada de T_1 son símbolos con un blanco en la pista inferior y un símbolo de entrada de T_2 en la pista superior.

La construcción formal de $T_1 = (K_1, \Sigma_1, \Gamma_1, \delta_1, q_{0_1}, F_1)$ es como sigue. Los estados K_1 de T_1 son objetos de la forma $[q, U]$ o $[q, L]$, donde $q \in K_2$ además del símbolo q_{0_1}. Note que, como ya se mencionó, la segunda componente indica que T_1 está trabajando en la pista superior o inferior (U por *upper*, L por *lower*). Los símbolos de entrada en Γ_1 son todos los objetos de la forma $[X, Y]$, donde X y Y están en Γ_2. Además, Y pudiera ser ¢, un símbolo que no está en Γ_2. Si B es el blanco de T_2, $[B, B]$ es el blanco de T_1. Σ_1 contiene todos los símbolos de la forma $[a, B]$, donde $a \in \Sigma_2$. $F_1 = \{[q, D] | q \in F_2, D = U \text{o} L\}$. Definimos δ_1 como sigue:

1. Para cada $a \in \Sigma_2$

$$\delta_1(q_{0_1}, [a, B]) = ([q, U], [X, \text{¢}], R) \text{ si } \delta_2(q_{0_2}, a) = (q, X, R)$$

2. Para cada $a \in \Sigma_2$

$$\delta_1(q_{0_1}, [a, B]) = ([q, L], [X, \text{¢}], R) \text{ si } \delta_2(q_{0_2}, a) = (q, X, L)$$

3. Para cada $[X, Y] \in \Gamma_1$, con $Y \neq \text{¢}$ y $D = L$ o R

$$\delta_1([q, U], [X, Y]) = ([p, U], [Z, Y], D) \text{ si } \delta_2(q, X) = (p, Z, D)$$

4. Para cada $[X, Y] \in \Gamma_1$, con $Y \neq \text{¢}$. Si D es L, $\bar{D}$ es R, si D es R, $\bar{D}$ es L

$$\delta_1([q, L], [X, Y]) = ([p, L], [X, Z], D) \text{ si } \delta_2(q, Y) = (p, Z, \bar{D})$$

5. Si $D = R$ entonces $E = U$ y si $D = L$ entonces $E = L$

$$\delta_1([q, U], [X, \text{¢}]) = \delta_1([q, L], [X, \text{¢}]) = ([p, E], [Y, \text{¢}], R) \text{ si } \delta_2(q, X) = (p, Y, D)$$

Ejercicio 2 Demuestre que T_1 y T_2 aceptan el mismo lenguaje.

4.2. Múltiples cintas

Una máquina de Turing con múltiples cintas consiste de un control finito con k cabezas lectoras y k cintas que son infinitas en ambas direcciones. Dependiendo del estado del control finito y el símbolo leído por cada cabeza lectora la máquina puede en un sólo movimiento:

1. cambiar de estado,
2. imprimir un símbolo nuevo sobre cada una de las celdas leídas por las cabezas de las cintas,
3. mover cada una de las cabezas de las cintas, independientemente, una celda a la derecha, a la izquierda o manternerla estacionaria.

Inicialmente la entrada está sobre la primer cinta y las otras cintas están en blanco.

Ejercicio 3 Defina formalmente la máquina de Turing con múltiples cintas.

Teorema 2 Si un lenguaje L es aceptado por una máquina de Turing con múltiples cintas entonces es aceptada por una máquina de Turing con una sola cinta.

Prueba. Sea L aceptada por una máquina de Turing T_1 con k cintas. Podemos construir una máquina de Turing T_2 con una sola cinta pero con $2k$ pistas, es decir, dos pistas por cada una de las cintas de T_1. Una pista registra el contenido de la cinta correspondiente de T_1 y la otra está en blanco, excepto por el marcador X en la celda que mantiene al símbolo leído por la cabeza lectora correspondiente de T_1.

El control finito de T_2 almacena información correspondiente a cuáles marcadores de cabeza están a la izquierda y cuáles a la derecha de la cabeza lectora de T_2, El estado de T_1 se almacena también en el control finito de T_2.

Para simular un movimiento de T_1 la máquina de Turing T_2 debe visitar cada una de las celdas con marcadores de cabeza y registrar, uno por uno, el símbolo leído por cada cabeza lectora de T_1. Cuando T_2 pase a través de un marcador de cabeza debe actualizar la dirección en la que encontró este marcador. Después de recabar la información necesaria, T_2 determina el movimiento que haría T_1. Luego T_2 vuelve a visitar, uno por uno, los marcadores de cabeza, para cambiar los símbolo leidos y mover los marcadores, si es necesario, una celda. Por supuesto, si el nuevo estado de T_1 es de aceptación entonces T_2 acepta.

Ejercicio 4 Formalice las transiciones de la máquina de Turing con una sola cinta T_2 para que pueda simular a la máquina de Turing con k cintas T_1 y así poder demostrar que si $w \in \mathcal{L}(T_1)$ entonces $w \in \mathcal{L}(T_2)$.

5. La clase de conjuntos recursivos

Una vez demostrado que hay lenguajes que no son enumerables recursivamente debemos preguntarnos cuál es la relación entre los lenguajes recursivos y los lenguajes enumerables recursivamente, por la definición de ambas clases de lenguajes es claro que si un lenguaje es recursivo entonces es enumerable recursivamente (justifique). Pero ¿la contención será propia o impropia?, es decir, ¿existirán lenguajes que siendo enumerables recursivamente no sean recursivos? Recuerde que no es suficiente responder de manera existencial, es necesario exhibir al lenguaje con tales características. Para ello veamos primero dos lemas.

Lema 1 Si un conjunto es recursivo su complemento es recursivo.

Prueba. Si L es un conjunto recursivo, $L \subseteq \Sigma^*$, entonces existe una máquina de Turing T que acepta L y que siempre para, por lo que está garantizado que, después de aceptar, T ya no se mueve. Construya T_1 a partir de T agregando un estado q, el cual es el único estado de aceptación de T_1. Las transiciones de T_1 incluyen todas las transiciones de T, de tal manera que T_1 simula a T. Además, para cada par compuesto de un estado de no aceptación y un símbolo de cinta de T para el cual no esté especificado un movimiento de T, T_1 transfiere el control al estado q y luego para.

Así T_1 simula a T hasta que T para. Si T para en uno de sus estados de aceptación T_1 para sin aceptar. Si T para en un estado de no aceptación seguramente no ha aceptado su entrada, así T_1 hace un movimiento adicional al estado q y acepta. Por lo tanto concluimos que T_1 acepta $\Sigma^* - L$.

Lema 2 Sean $x_1, x_2, \ldots$ una enumeración efectiva de todas las sentencias sobre algún alfabeto finito Σ y $T_1, T_2, \ldots$ una enumeración efectiva de todas las máquinas de Turing con símbolos de cinta elegidos de algún alafabeto finito que incluya a Σ. Sea L_2 el conjunto $\{x_i | x_i$ es aceptado por $T_i\}$. L_2 es un conjunto enumerable recursivamente cuyo complemento no es enumerable recursivamente.

Prueba. Las sentencias de L_2 pueden ser aceptadas por una máquina de Turing T que opera como sigue. Note que T no necesariamente para con sentencias que no están en L_2.

1. Dada una sentencia x, T enumera sentencias $x_1, x_2, \ldots$ hasta que encuentra la sentencia $x_i = x$ determinando de esta manera que x es la i-ésima sentencia en la enumeración,

2. T genera T_i y transfiere el control a una máquina de Turing universal que simula T_i con entrada x_i,

3. Si T_i con entrada x_i para y acepta entonces T para y acepta; si T_i para y rechaza x_i entonces T para y rechaza x_i. Finalmente si T_i no para entonces T no para.

Así L_2 es enumerable recursivamente ya que L_2 es el conjunto aceptado por T. Ahora $\bar{L}_2$ no puede ser enumerable recursivamente, ya que si T_j fuera la máquina de Turing que acepta $\bar{L}_2$ entonces x_j está en $\bar{L}_2$ ssi x_j no es aceptado por T_j, lo cual contradice la suposición de que $\bar{L}_2$ es el lenguaje aceptado por T_j.

Ejercicio 5 **Teorema 3** Demuestre que existe un conjunto enumerable recursivamente que no es recursivo. *Sugerencia: primero use el lema 2 y luego el lema 1.*

Ejercicio 6 Pruebe que no existe un algoritmo para determinar si una máquina de Turing arbitraria que empieza con su cinta en blanco eventualmente para.

6. Máquinas de Turing y gramáticas de tipo 0

Teorema 4 Si L es generado por una gramática de tipo 0 entonces L es reconocido por una máquina de Turing .

Prueba. Sean $G = (V_N, V_T, P, S)$ una gramática de tipo 0 y $L = L(G)$. Se describe informalmente a una máquina de Turing no determinista T que acepta a L. Sea

$$T = (K, V_T, \Gamma, \delta, q_0, F), \text{ donde } \Gamma = V_N \cup V_T \cup \{B, \#, X\}$$

Se asume que los tres últimos símbolos no están en V_N o V_T. No se enumerarán todos los estados de K pero se designarán algunos cuando sea necesario. Informalmente si hace falta se permite que T imprima el símbolo blanco B.

Para empezar T tiene una entrada $w \in V_T^*$ sobre su cinta. T inserta # antes de w, corriendo los símbolos de w una celda a la derecha y a continuación pone $\#S\#$. El contenido de la cinta ahora es $\#w\#S\#$.

Después T simula de manera no determinista una derivación en G iniciando con S. Cada forma sentencial en la derivación aparecerá en su momento entre los dos últimos #. Si alguna elección de movimientos conduce a una cadena de símbolos terminales entonces esa cadena se compara con w. Si son iguales T acepta.

Formalmente, T tiene $\#w\#A_1A_2\cdots A_k\#$ sobre su cinta. T mueve su cabeza a través de $A_1A_2\cdots A_k$ de manera no determinista eligiendo una posición i y una constante r entre 1 y la longitud máxima del lado izquierdo de las producciones en P. Luego T examina la subcadena $A_iA_{i+1}\cdots A_{i+r-1}$ si ésta coincide con el lado izquierdo de alguna producción en P entonces la subcadea se puede remplazar por el lado derecho de P. Puede ser que T tenga que correr $A_{i+r}A_{i+r+1}\cdots A_k\#$ a la izquierda o a la derecha para cerrar o abrir espacio, recuerde que el lado derecho de la producción usada puede tener una longitud distinta de r (el símbolo X se usa en el corrimiento a la derecha).

A partir de esta simple simulación de derivaciones en G debe ser claro que T imprimirá sobre su cinta una cadena de la forma $\#w\#\alpha\#$, con $\alpha \in V_T^*$ exactamente cuando $S \Rightarrow_G^* \alpha$. Además si $\alpha = w$ entonces T acepta.

Teorema 5 Si L es reconocido por una máquina de Turing entonces L es generado por una gramática de tipo 0.

7. Autómatas acotados linealmente y lenguajes sensibles al contexto

Un *autómata acotado linealmete* (aal) es una máquina de Turing no determinista con una sola cinta que nunca deja las celdas sobre las que se escribió la entrada. Formalmente, un autómata acotado linealmete se denota como $M = (K, \Sigma, \Gamma, \delta, q_0, F)$. Los símbolos tienen esencialmente el mismo significado que el de la máquina de Turing . El conjunto de *estados* es K, el conjunto de *estados finales* es $F \subseteq K$, el conjunto de *símbolos de cinta* es Γ, el conjunto de *símbolos de entrada* es $\Sigma \subseteq \Gamma$, el *estado inicial* es $q_0 \in K$, δ es un mapeo de $K \times \Gamma$ a subconjuntos de $K \times \Gamma \times \{L, R\}$.

Σ contiene dos símbolos especiales, usualmente denotados como ¢ y \$, los cuales son respectivamente los *marcadores de los extremos izquierdo y derecho.* Estos símbolos están inicialmente en los extremos de la entrada, su función es evitar que la cabeza de cinta abandone la región de la cinta donde está la entrada.

Una configuración de M y la relación $\vdash_M$, entre configuraciones que se establece si la segunda se puede derivar de la primera al aplicar una regla de δ, están definidas esencialmente como en el caso de la máquina de Turing . Una *configuración* de M se denota por $(q, A_1A_2\cdots A_n, i)$ donde $q \in K$, $A_1, A_2, \ldots, A_n$ están en Γ e i es un entero entre 1 y n. Suponga que $\delta(q, A_i)$ contiene (p, A, L) y que $i > 1$ entonces decimos que

$$(q, A_1A_2\cdots A_n, i) \vdash_M (p, A_1A_2\cdots A_{i-1}AA_{i+1}\cdots A_n, i-1)$$

Si $\delta(q, A_i)$ contiene (p, A, R) e $i < n$ entonces decimos que

$$(q, A_1A_2\cdots A_n, i) \vdash_M (p, A_1A_2\cdots A_{i-1}AA_{i+1}\cdots A_n, i+1)$$

Es decir, M imprime A sobre A_i, cambia su estado a p y mueve su cabeza a la izquierda o a la derecha, pero no sale de la región en la que originalmente aparecian los símbolos. Como es usual definimos la relación $\vdash_M^*$ como la cerradura reflexiva y transitiva de $\vdash_M$, es decir, como $(q, \alpha, i) \vdash_M^* (q, \alpha, i)$ y si

$$(q_1, \alpha_1, i_1) \vdash_M^* (q_2, \alpha_2, i_2) \text{ y } (q_2, \alpha_2, i_2) \vdash_M (q_3, \alpha_3, i_3)$$

entonces

$$(q_1, \alpha_1, i_1) \vdash_M^* (q_3, \alpha_3, i_3)$$

El *lenguaje aceptado por M* es

$$\{w | w \in (\Sigma - \{¢, \$\})^* \text{ y } (q_0, ¢w\$, 1) \vdash_M^* (q, \alpha, i) \text{ para algún } q \in F, \alpha \in \Gamma^* \text{ y un entero } i\}.$$

Decimos que M es *determinista* si $\delta(q, A)$ no contine más de un elemento para cualquier $q \in K$ y $A \in \Gamma$.