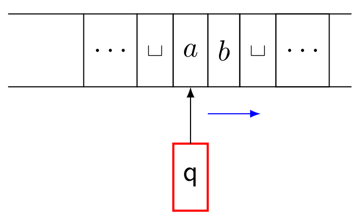
Autómatas y Lenguajes Formales 2017-1
Facultad de Ciencias UNAM
Nota de Clase 13

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

17 de noviembre de 2016

1. Máquinas de Turing

Las máquinas de Turing (MT) son máquinas **idealizadas** capaces de realizar cómputos. Una máquina de Turing consiste de una cinta infinita dividida en sectores (cuadros) y una cabeza de lecto-escritura. Cada sector de la cinta contiene un símbolo de cierto alfabeto de entrada o bien el símbolo blanco. La cabeza lee el sector y puede escribir sobre él así como moverse a la izquierda o a la derecha.



Definición 1 *Una máquina de Turing es una septupla:*

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F \rangle$$

- $Q \neq \varnothing$ *es un conjunto finito de estados.*
- Σ *es el alfabeto de entrada.*
- Γ *es el alfabeto de la cinta, el cual incluye a* Σ*, es decir,* $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ *es la función de transición*
- $q_0 \in Q$ *es el estado inicial.*
- $\sqcup \in \Gamma$ *es el símbolo blanco tal que* $\sqcup \notin \Sigma$.
- $F \subseteq Q$ *es el conjunto de estados finales.* F *podría ser vacío.*

La función de transición está determinada por:

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$$

es decir $\delta(q, a) = (p, b, D)$ donde

- El estado actual es q y el símbolo a leer es a.
- La transición es hacia el estado p.
- b es el símbolo escrito en el lugar de a o en el sector donde se encontraba a.
- La cabeza se mueve un sector según la dirección dada por $D \in \{\leftarrow, \to\}$.
  Dichos movimientos se realizan después de leer a y escribir b.

Ejemplo: Si $\delta(q, a) = (p, b, \to)$ entonces

- Estado actual: q
- Símbolo a leer: a.
- La cabeza borra a y escribe b.
- El nuevo estado es p.
- La cabeza se mueve un sector a la derecha.

Ejemplo: Considere la siguiente máquina

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{a, b, \sqcup\}, \delta, q_0, \sqcup, \{q_1\} \rangle$$

donde la función de transición está definida por:

$$\delta(q_0, a) = (q_0, b, \to) \qquad \delta(q_0, b) = (q_0, b, \to) \qquad \delta(q_0, \sqcup) = (q_1, \sqcup, \leftarrow)$$

Ejemplo: Considere el lenguaje $L = \{a^n b^n \mid n \geq 1\}$, una MT que reconoce cadenas del lenguaje está determinada por la siguiente función de transición donde $F = \{q_4\}$:

δ	a	b	X	Y	$\sqcup$
q_0	$(q_1, X, \to)$			$(q_3, Y, \to)$	
q_1	$(q_1, a, \to)$	$(q_2, Y, \leftarrow)$		$(q_1, Y, \to)$	
q_2	$(q_2, a, \leftarrow)$		$(q_0, X, \to)$	$(q_2, Y, \leftarrow)$	
q_3				$(q_3, Y, \to)$	$(q_4, \sqcup, \to)$
q_4					

1.1. Representación gráfica de una Máquina de Turing

Como hemos visto en otras máquinas o autómatas, existe una representación gráfica que modela la transición entre estados:

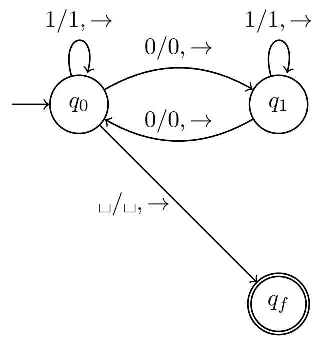
- cada nodo es un estado;
- las aristas dirigidas son las transiciones y están etiquetadas por el símbolo que lee la cabeza, el símbolo que escribirá y el movimiento que realizará;
- hay un estado inicial y uno o varios estados finales.

Ejemplo: Una máquina de Turing que acepta al lenguaje

$$L = \{w \in \{0,1\}^\star \mid w \text{ tiene un número par de ceros } \}$$

está determinada por la función de transición siguiente y se describe gráficamente como:

δ	0	1	$\sqcup$
q_0	$(q_1, 0, \rightarrow)$	$(q_0, 1, \rightarrow)$	$(q_f, \sqcup, \rightarrow)$
q_1	$(q_0, 0, \rightarrow)$	$(q_1, 1, \rightarrow)$	
q_f			



Existen diferentes versiones de máquinas de Turing, cada una para ser utilizada en casos especiales. Estudiemos primero la formalización estándar de ellas y enfatizaremos sus características.

1.2. Máquina Estándar de Turing

Estas máquinas tienen una cinta infinita en ambas direcciones. Se permite un número arbitrario de movimientos en cualquier dirección.

Una característica importante es que estas máquinas son deterministas: δ define a lo más un movimiento para cada configuración posible. No hay transiciones desde estados finales, es decir, $\delta(q, a)$ no está definida si $q \in F$. Tampoco hay un archivo especial de entrada o salida, se asume que la máquina contiene algo al final y al principio del proceso.

Se considera que la cadena a procesar está almacenada en la cinta: cada símbolo de la cadena está en un sector de la cinta. La cabeza se encuentra al inicio de la cadena, o puede suponerse que se encuentra un sector antes de la cadena. Por esto se puede pensar que la cinta es un dispositivo de almacenamiento.

Definición 2 *Una configuración o descripción instantánea*

$$a_1 a_2 \ldots a_{k-1} \boldsymbol{q} a_k a_{k+1} \ldots a_n$$

está determinada por:

- *El estado actual de la unidad de control (cabeza).*
- *El contenido de la cinta.*
- *La posición de la unidad de control.*
- *La configuración inicial es:* $q_0 w$

Las configuraciones nos permiten formalizar la noción de cómputos en las Máquinas de Turing:

Definición 3 *Un cómputo o paso de computación es el cambio de una descripción instantánea a otra mediante una transición dada por* δ*:*

$$u\boldsymbol{q}av \vdash ub\boldsymbol{p}v \quad \textit{si y sólo si} \quad \delta(q,a) = (p,b,\rightarrow)$$

$$uc\boldsymbol{q}av \vdash u\boldsymbol{p}cbv \quad \textit{si y sólo si} \quad \delta(q,a) = (p,b,\leftarrow)$$

Además $\vdash^\star$ *se define de la manera usual.*

El símbolo para el espacio en blanco tiene diferentes interpretaciones, algunos autores no hacen distinción entre la cadena vacía y el espacio en blanco. Veamos algunos casos especiales para la cadena vacía:

- $u\mathbf{q}a \vdash ub\mathbf{p}_\sqcup$ si y sólo si $\delta(q,a) = (p,b,\rightarrow)$
- $\mathbf{q}av \vdash \mathbf{p}_\sqcup bv$ si y sólo si $\delta(q,a) = (p,b,\leftarrow)$

Algunas situaciones especiales en cuanto a cómputos:

- Cómputos bloqueados:
  el cómputo se bloquea porque la siguiente transición no está definida: $\quad u\mathbf{q}v \not\vdash^\star$
- Cómputos infinitos:
  el cómputo entra en un ciclo infinito: $\quad u\mathbf{q}v \vdash^\star \infty$

Definición 4 *El lenguaje de aceptación se define como todas aquellas cadenas de entrada con las cuales la máquina se detiene en un estado final:*

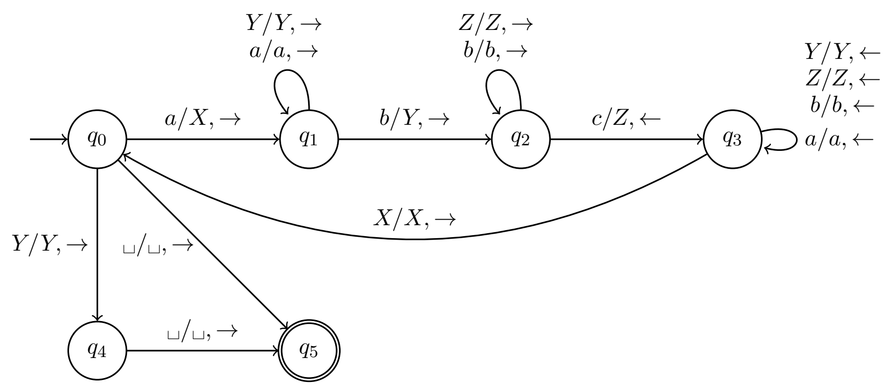
$$L(M) = \{w \in \Sigma^\star \mid q_0 w \vdash^\star w_1 q_f w_2 \quad q_f \in F\}$$

Observaciones: **Las máquinas estándar se denominan máquinas aceptadoras ya que determinan si una cadena pertenece a un lenguaje:**

- A diferencia con los autómatas se puede aceptar una cadena en el momento en que el proceso llega a un estado final.
- **NO** es necesario consumir toda la cadena, es decir, se deben contemplar w_1 y w_2 cadenas, inclusive pueden ser una subcadena de w.
- Si $\boldsymbol{\varepsilon} \in L$ para algún lenguaje entonces la cinta no contiene información al inicio y se acepta al avanzar sólo un sector vacío para llegar al estado de aceptación.

- Una máquina que acepta el lenguaje vacío debe aceptar al tener una transición hacia el estado final.
- Recordemos que se asume en el modelo estándar que no hay transición alguna desde un estado final, lo cual evita ambigüedades.
- Si la máquina se detiene en un estado no final o se encuentra bloqueada se puede concluir que la cadena a procesar no pertenece al lenguaje.
- También se puede considerar la existencia de un estado de rechazo y en tal caso la máquina se detiene al llegar a ese estado rechazando a la cadena.

Ejemplo: La siguiente máquina acepta el lenguaje $L = \{a^i b^i c^i \mid i \geq 0\}$



2. Variaciones en MT

Existen diversas variaciones en la definición de máquinas de Turing. Todas ellas resultan equivalentes, es decir, el poder de computación de cualquier modelo resulta equivalente al de la máquina estándar. Las variaciones son útiles para simplificar la presentación o programación de diversos problemas.

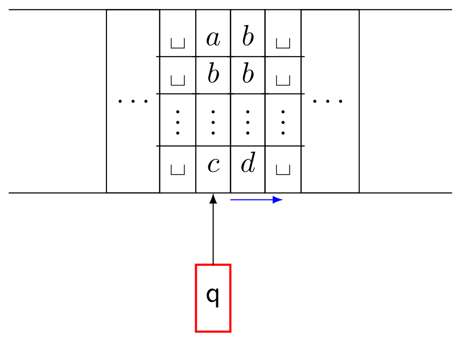
1. MT con cabeza lectora estacionaria
   - Se permite que al leer y escribir un símbolo la cabeza no realice movimiento alguno.
   - El conjunto de direcciones se amplia a $\{\leftarrow, \rightarrow, -\}$.
   - La transición
     $$\delta(q, a) = (p, b, -)$$
     significa que la cabeza lee a, escribe b y no se mueve.
   - Tales transiciones pueden simularse mediante un nuevo estado y movimientos consecutivos a la izquierda y a la derecha.

2. MT con múltiples pistas

   - Idea: la cinta se divide en multiples pistas.
   - La función de transición es:

   $$\delta : Q \times \Gamma^n \to Q \times \Gamma^n \times \{\leftarrow, \rightarrow\}$$

   es decir $\delta(q, \langle a_1, \ldots, a_n \rangle) = (p, \langle b_1, \ldots, b_n \rangle, D)$ con $D \in \{\leftarrow, \rightarrow\}$.

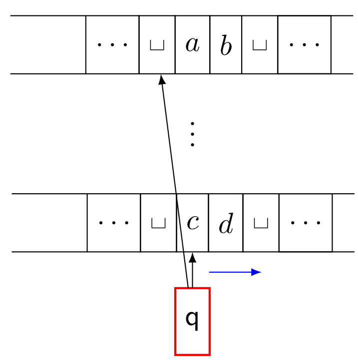


3. MT con múltiples cintas

   - Idea: se agregan más cintas a la máquina.
   - La función de transición es:

   $$\delta : Q \times \Gamma^n \to Q \times (\Gamma \times \{\leftarrow, \rightarrow\})^n$$

   es decir $\delta(q, \langle a_1, \ldots, a_n \rangle) = (p, \langle b_1, D_1 \rangle, \ldots, \langle b_n, D_n \rangle)$ con $D_i \in \{\leftarrow, \rightarrow\}$.



4. MT No-deterministas

   - La función de transición es:

   $$\delta : Q \times \Gamma^n \to \mathcal{P}(Q \times \Gamma \times \{\leftarrow, \rightarrow\})$$

   es decir $\delta(q, \langle a_1, \ldots, a_n \rangle) = \{\langle b_1, D_1 \rangle, \ldots, \langle b_n, D_n \rangle\}$
   - Las máquinas no-deterministas juegan un papel central en la teoría de la complejidad.

3. MT Calculadoras

Otro tipo de máquinas pueden ser usadas como *macros* cuyo objetivo es calcular sin necesidad de aceptar o generar cadenas en un lenguaje. La máquina de Turing

$$M = \langle \{q_0, \ldots, q_{\mathfrak{F}}\}, \Sigma, \Gamma, \delta, q_0, \sqcup, \varnothing \rangle$$

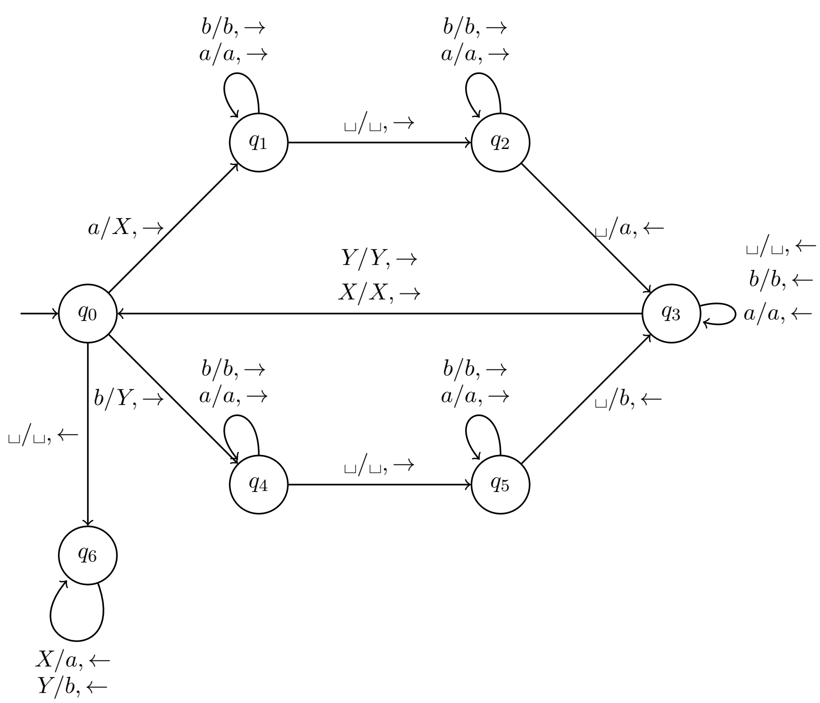
calcula una función $\mathfrak{F} : \Sigma^\star \to \Gamma^\star$ si

$$q_0 w \vdash^\star q_{\mathfrak{F}} v \quad \text{donde } \mathfrak{F}(w) = v$$

Obsérvese que :

1. No hay estados finales, el estado $q_{\mathfrak{F}}$ se usa para detener la máquina, es decir, no hay transiciones desde $q_{\mathfrak{F}}$.
2. El proceso se termina en $q_{\mathfrak{F}} v$, es decir la cabeza debe estar leyendo el primer símbolo de la salida v.

Ejemplo: Un macro que copia una cadena se basa en la idea de recorrer la cadena de entrada almacenada en la cinta símbolo por símbolo, para copiarlo en los sectores posteriores a la cadena pero usando un sector en blanco para separar las cadenas.
La siguiente máquina copia cadenas sobre el alfabeto $\Sigma = \{a, b\}$:



4. MT Generadoras

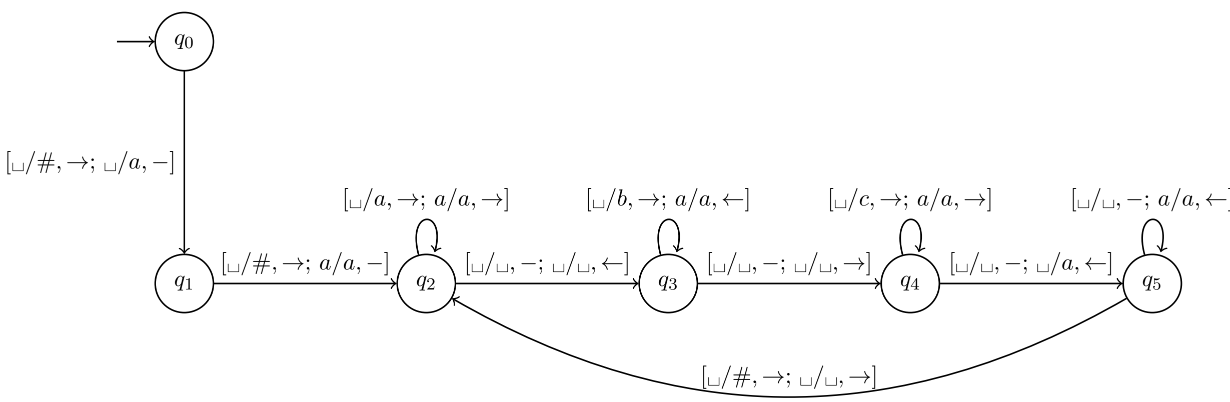
Así como hay variantes en las máquinas aceptadoras, se puede hablar de máquinas generadoras que tienen suficiente poder como para generar lenguajes y calcular funciones. Una máquina de Turing M con múltiples cintas genera al lenguaje $L \subseteq \Sigma^\star$ si

- M comienza a operar con las cintas en blanco en q_0.
- Cada vez que M regresa a q_0 hay una cadena de L escrita sobre la cinta principal.
- Eventualmente se generan todas las cadenas de L que se almacenarán en la cinta principal separadas por un espacio en blanco o algún otro símbolo para este efecto.

Ejemplo: Analicemos una máquina que genera todas las cadenas del lenguaje: $L = \{a^i b^i c^i \mid i \geq 0\}$. Utilizaremos dos cintas y la idea del diseño es la siguiente:

1. se inicia con la cinta $C1$ vacía y se dejan dos símbolos # para indicar que la cadena vacía pertenece al lenguaje
2. simultáneamente se escribe una a en la cinta $C2$ y la cabeza lectora retrocede
3. a continuación se generarán las cadenas utilizando un ciclo que no termina como sigue:
    a) en la cinta $C1$ se escribe una a por cada a en la cinta $C2$
    b) la cabeza en $C2$ regresa a la izquierda para escribir b en $C1$ por cada a que lea
    c) nuevamente se mueve la cabeza a la derecha en $C2$ para escribir en $C1$ tantas c como a
    d) se incrementa una a en la cinta $C2$ para generar la siguiente cadena
    e) se escirbe el símbolo # en $C1$ para separar las cadenas

Veamos su diagrama:



5. Lenguajes Recursivos y Recursivamente enumerables

La aceptación en una máquina de Turing definen unas clase de lenguajes muy expresivos:

- Un lenguaje L es **recursivamente enumerable** si es reconocido por una máquina de Turing, es decir, si existe una máquina de Turing M tal que $L = L(M)$.
- Un lenguaje L es **recursivo** si es reconocido por una máquina de Turing que **siempre se detiene**, es decir, si existe una máquina de Turing M que se detiene con todas las cadenas de entrada y $L = L(M)$.

Los lenguajes definidos gozan de algunas propiedades de cerradura:

- Si L es recursivo entonces $\overline{L}$ es recursivo.
- Si L, M son recursivos entonces $L \cup M$ es recursivo.
- Si L, M son rec. enumerables entonces $L \cup M$ es rec. enumerable.
- L es recursivo si y sólo si L y $\overline{L}$ son rec. enumerables.

6. Máquinas de Turing y gramáticas

Recordando la Jerarquía de Chomsky y su relación con autómatas:

	Tipo de Gramática	Máquina equivalente
3	Regulares	autómatas finitos
2	Libres de Contexto	autómatas de pila
1	Sensibles al Contexto	**??**
0	Irrestrictas	**??**

¿Qué pasa con las gramáticas tipo 1 (sensibles al contexto)?
¿Qué pasa con las gramáticas tipo 0 (irrestrictas)?

Ejemplo: El lenguaje $L = \{a^i b^i c^i \mid i \geq 0\}$ es generado por la siguiente gramática sensible al contexto:

$$\begin{array}{rcl} S & \rightarrow & A \\ A & \rightarrow & AABC \mid aBC \\ CB & \rightarrow & BC \\ bB & \rightarrow & bb \\ bC & \rightarrow & bc \\ cC & \rightarrow & cc \end{array}$$

Al restringir que $i \geq 1$ tenemos la siguiente gramática:

$$\begin{array}{rcl} S & \rightarrow & aSBc \mid abc \\ cB & \rightarrow & Bc \\ bB & \rightarrow & bb \end{array}$$

Recordemos que las gramáticas de tipo 1 permiten tener cadenas de símbolos terminales y no terminales en ambos lados de $\rightarrow$ de la forma:

$$\alpha A\beta \rightarrow \alpha\gamma\beta \text{ donde } \gamma \neq \boldsymbol{\varepsilon},\ A \in V \text{ y } \alpha, \beta, \gamma \in (V \cup \Sigma)^{\star}$$

Se denominan sensibles al contexto dado que la reescritura de A en γ depende del contexto $\alpha\gamma$. Y los autómatas que reconocen estos lenguajes son los llamados acotados.

6.1. Gramáticas Sensibles al Contexto y Máquinas de Turing

La correspondencia entre los autómatas linealmente acotados y las gramáticas es consecuencia de que todo lenguaje sensible al contexto es recursivo y está descrita por:

1. Dada una gramática sensible al contexto G, existe un autómata linealmente acotado M tal que $L(G) = L(M)$. Es decir, los lenguajes sensibles al contexto son reconocidos por autómatas linealmente acotados.

2. Si $L = L(M)$ es un lenguaje reconocido por un autómata linealmente acotado M entonces existe una gramática sensible al contexto G tal que $L(M) = L(G)$. Es decir, los lenguajes reconocidos por autómatas linealmente acotados son sensibles al contexto.

A continuación describimos el proceso en el que dada una gramática sensible al contexto G se puede obtener una máquina de Turing que simule las derivaciones de G:

- Construir una máquina de Turing no-determinista con 3 cintas tales que:
  - la cinta $C1$ contiene la cadena a ser procesada
  - la segunda cinta $C2$ tiene la forma generada por la simulación
  - la última cinta $C3$ es la derivación

- Los pasos a seguir son los siguientes:
  1. $C3$ contiene $S\#$
  2. Una regla de producción $\alpha \rightarrow \beta$ se escoge desde $C2$
  3. Sea $\gamma\#$ la última cadena escrita en $C3$, así se toma una instancia de α en $\gamma = \gamma'\alpha\gamma''$ existente, sino se va al estado de rechazo
  4. escribir $\gamma'\alpha\gamma''\#$ en $C3$ despues de $\gamma\#$, esto indicará la aplicación de la regla
  5. Si $\gamma'\beta\gamma'' = w$ entonces la máquina se detiene y acepta
  6. Si $\gamma'\beta\gamma''$ aparece en otra posición dentro de $C3$ entonces es porque la máquina se ha detenido pero en el estado de rechazo
  7. Si $|\gamma'\beta\gamma''| > |w|$ entonces la máquina se detiene y también rechaza
  8. Repetir los pasos 2. al 6.

6.2. Autómatas Linealmente Acotados

Definición 5 *Un autómata linealmente acotado (ALA) es una máquina de Turing que satisface las siguientes condiciones:*

- *El alfabeto de entrada* Σ *incluye dos símbolos especiales* $[,]$ *que sirven como marcas de fin de cinta izquierda y derecha respectivamente.*
- *La cabeza lectora no puede desplazarse más allá de dichos límites y no puede sobreescribir tales sectores.*

Formalmente tenemos la tupla:

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, \sqcup, [,], F \rangle$$

con $[,] \in \Sigma$. *Además el lenguaje de aceptación es*

$$L(M) = \{w \in \Sigma^\star - \{[,]\} \mid q_0[w] \vdash^\star w_1 q_f w_2 \; q_f \in F\}$$

Las marcas $[,]$ no son consideradas como parte de la cadena a procesar pero son las que limitan el movimiento de la cabeza. Esta versión de máquina no puede moverse fuera de la cadena de entrada de ahí que se le nombre acotado.

6.3. MT y gramáticas irrestrictas

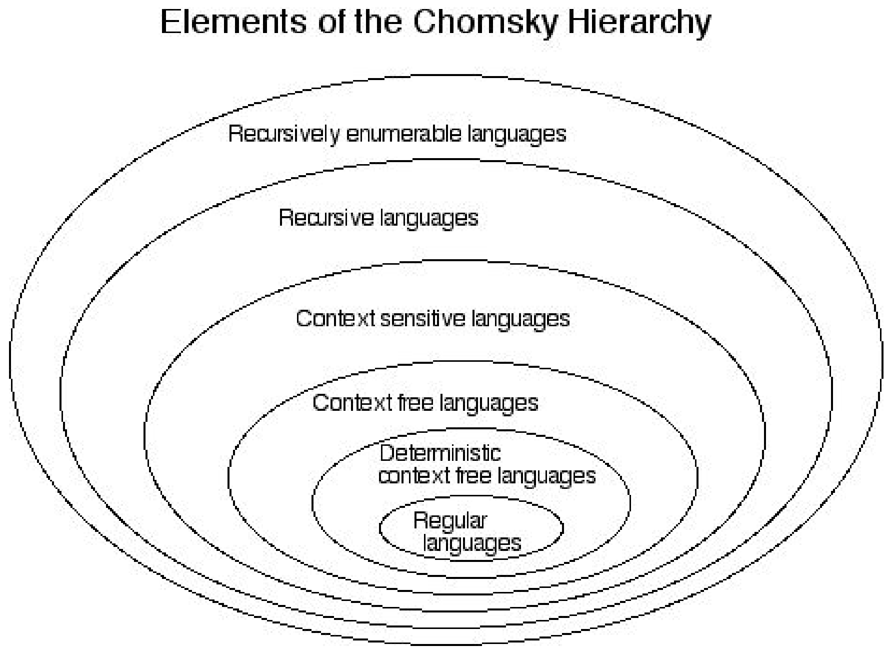
Las gramáticas sin restricciones son aquellas que están en correspondencia con las Máquinas de Turing:

- Para toda gramática G de tipo 0 existe una máquina de Turing M tal que $L(M) = L(G)$. Es decir, los lenguajes tipo 0 son recursivamente enumerables.
- Para toda máquina de Turing M existe una gramática G de tipo 0 tal que $L(G) = L(M)$. Es decir, los lenguajes recursivamente enumerables son lenguajes tipo 0.

Revisemos nuevamente la Jerarquía de Chomsky:

	Tipo de Gramática	Máquina equivalente
3	Regulares	autómatas finitos
2	Libres de Contexto	autómatas de pila
1	Sensibles al Contexto	autómatas linealmente acotados
0	Irrestrictas	máquinas de Turing

Finalmente se muestra a continuación un diagrama [1] que contiene la clasificación completa de los lenguajes:



[1] Diagrama original de Anup Kumar y Hemanth Kumar del Indian Institute of Science, Bangalore