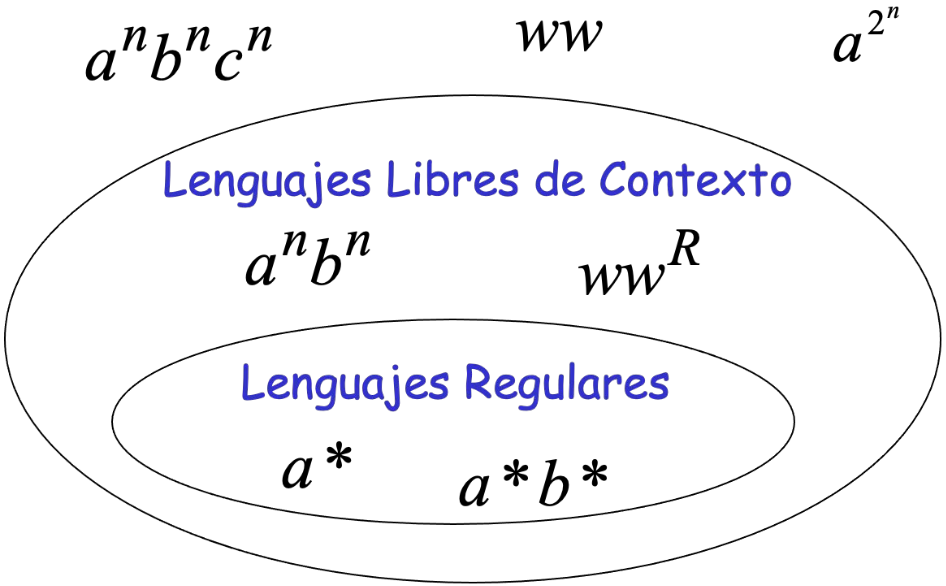
Gramáticas irrestrictas

Emanuel Alvaredo, Federico Lochbaum

Con la definición de máquina de Turing se permite

- Formalizar la matemática, la lógica y estudiar sus alcances e interrelaciones
- Establecer la computación como una disciplina científica y sus límites
- Caracterizar la clase de lenguajes reconocibles/generales con algoritmos


$a^n b^n c^n$
ww
a^{2^n}
Lenguajes Libres de Contexto
$a^n b^n$
ww^R
Lenguajes Regulares
a^*
$a^* b^*$

Gramáticas sin restricciones

Definimos una gramática irrestricta (o sin restricciones) como una tupla $G = (V, \Sigma, S, P)$ donde **todas** las producciones tienen la forma $\alpha \rightarrow \beta$ con $\alpha, \beta \in (V \cup \Sigma)^*$

Ejemplo

Sea G :

$S \rightarrow aSBC \mid \lambda \qquad CB \rightarrow BC$

$aB \rightarrow ab \qquad bB \rightarrow bb$

$bC \rightarrow bc \qquad cC \rightarrow cc$

$$S \Rightarrow^{3x1} aaaSBCBCBC \Rightarrow^{1} aaaBCBCBC \Rightarrow^{4} aaaBBCCBC \Rightarrow^{4}$$
$$aaaBBCBCC \Rightarrow^{4} aaaBBBCCC \Rightarrow^{2} aaabBBCCC$$
$$\Rightarrow^{5} aaabbBCCC \Rightarrow^{5} aaabbbCCC \Rightarrow^{3} aaabbbcCC$$
$$\Rightarrow^{6} aaabbbccC \Rightarrow^{6} aaabbbccc$$

Otro ejemplo, L = { a^{2n}, n >= 0 }

S → LaR

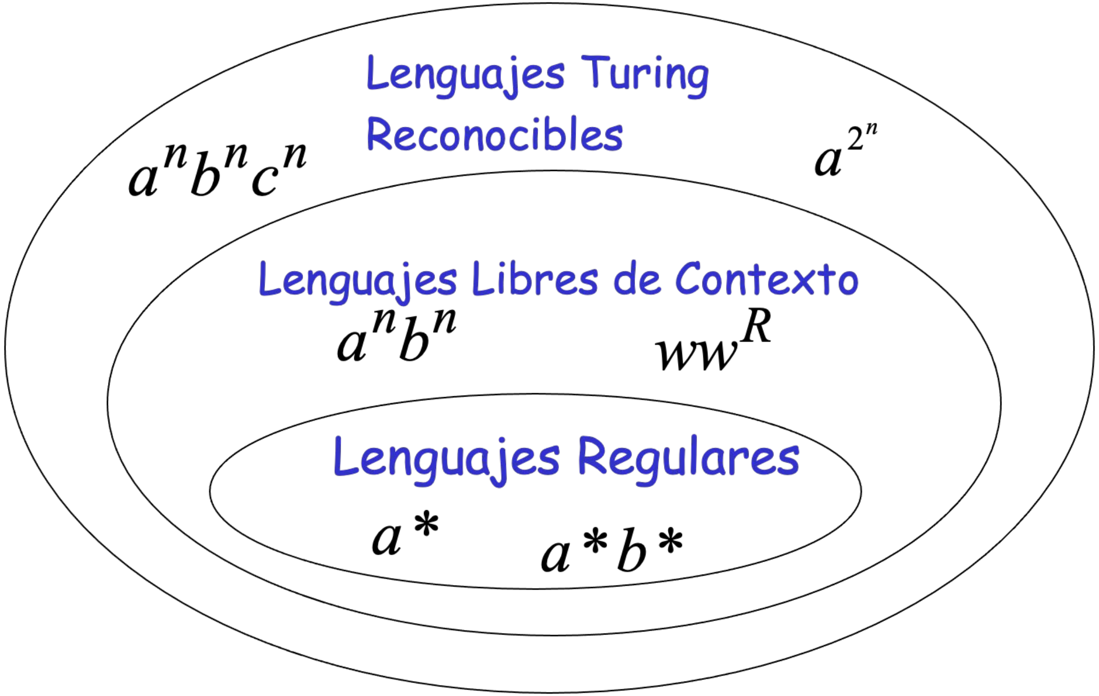
L → LD | λ

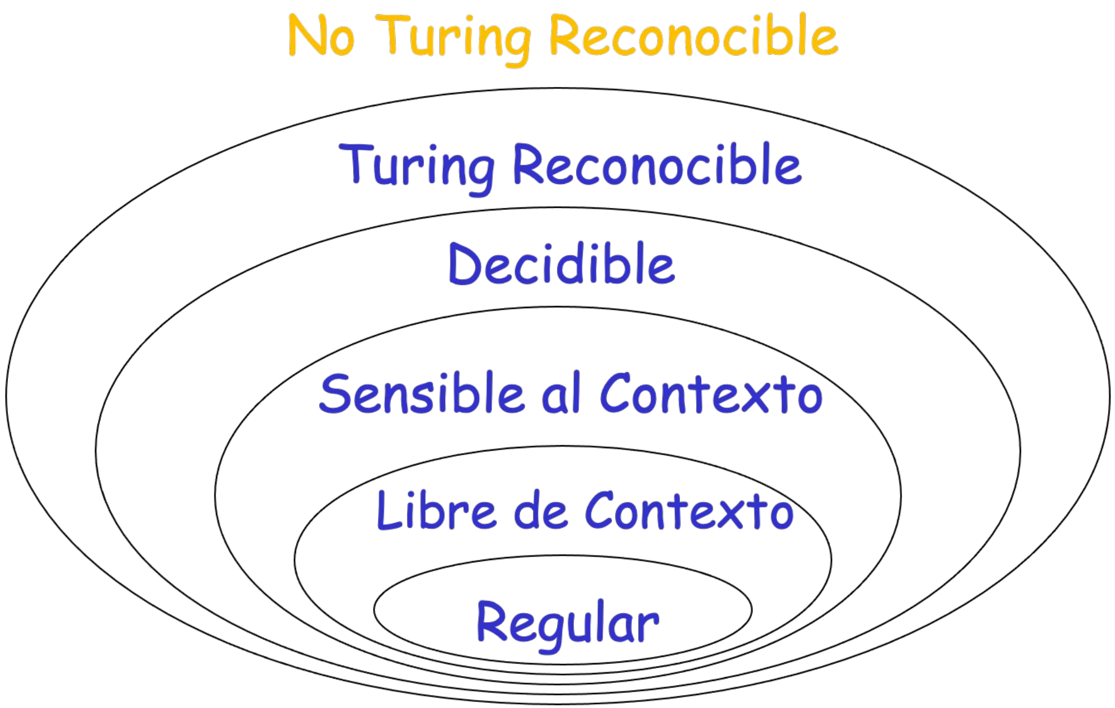
Da → aaD

DR → R

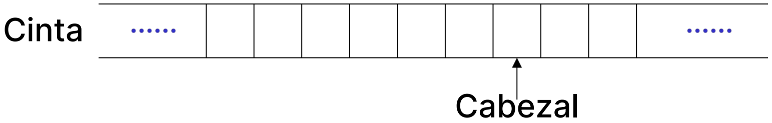
R → λ

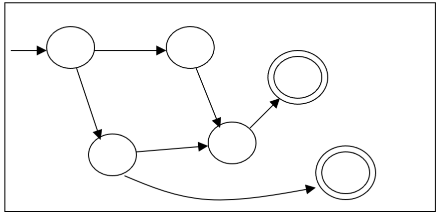
Notar que D es el no terminal “duplicador”


Lenguajes Turing
Reconocibles
$a^n b^n c^n$
a^{2^n}
Lenguajes Libres de Contexto
$a^n b^n$
ww^R
Lenguajes Regulares
a^*
$a^* b^*$


No Turing Reconocible
Turing Reconocible
Decidible
Sensible al Contexto
Libre de Contexto
Regular

Máquinas de Turing





- La cinta **NO** tiene limite (longitud infinita)

- El cabezal se mueve de izquierda a derecha

En cada transición (paso de tiempo), el cabezal:

1. Lee un símbolo
2. Escribe un símbolo
3. Se Mueve a Izquierda o Derecha

1 - Lee a

2 - Escribe k

3 - Se mueve a la izquierda

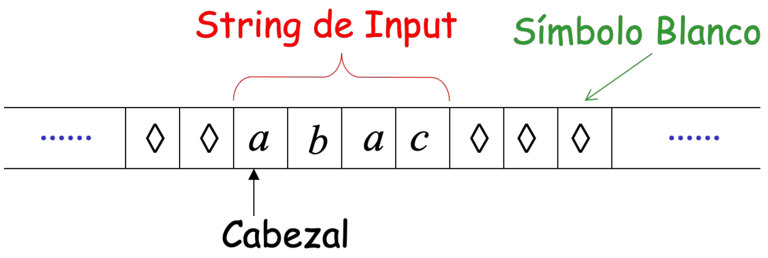
1 - Lee b

2 - Escribe f

3 - Se mueve a la derecha

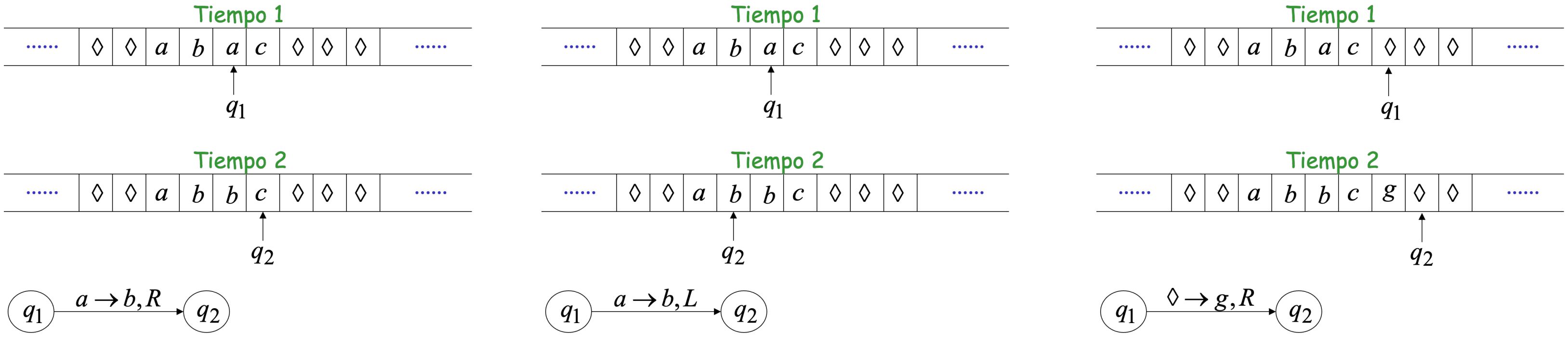
Detalles

El cabezal siempre comienza en el extremo izquierdo del string de input (no es el comienzo de la cinta)

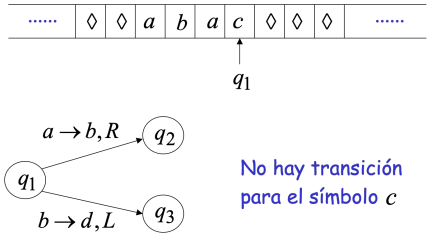


Denotamos las transiciones como

Ejemplos

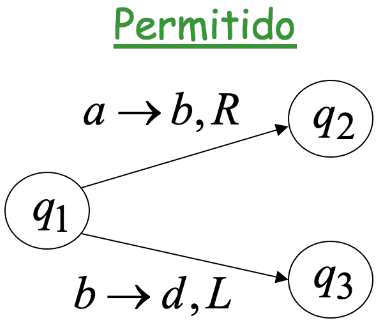


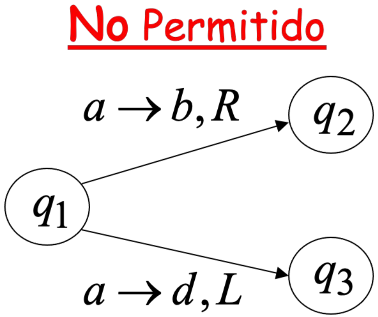
Se permite la ausencia de una transición para algún símbolo que esté en sigma



Determinismo

Las máquinas de Turing son deterministas

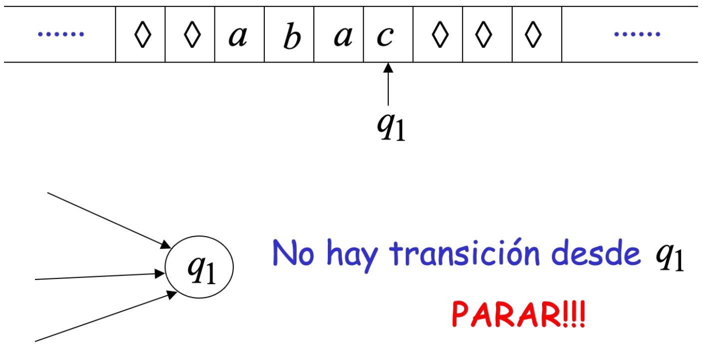


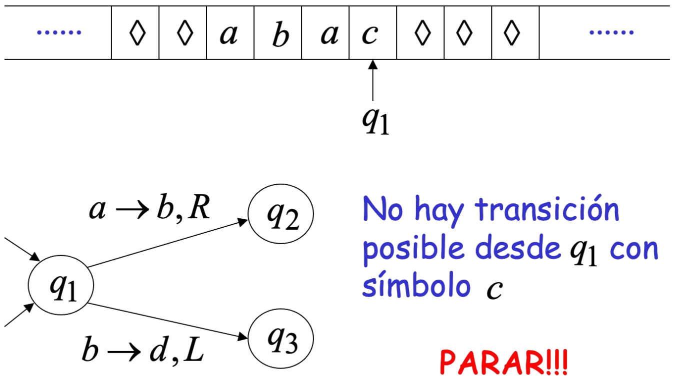


Las transiciones lambda no están permitidas

Detención

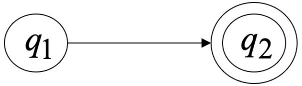
Una máquina de turing va a "**parar**" en un estado si no existe una transición para seguir consumiendo el input



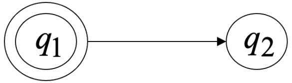


Aceptación

En una TM, los estados finales no tienen transiciones salientes, la máquina para y acepta



Permitido



No Permitido

Acepta el string de input → Si la máquina para en un estado final

Rechaza el string de input → Si la máquina para en un estado no final

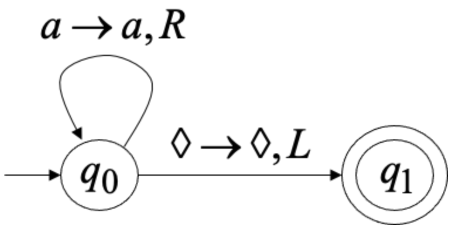
o

Si la máquina entra en un *ciclo infinito*

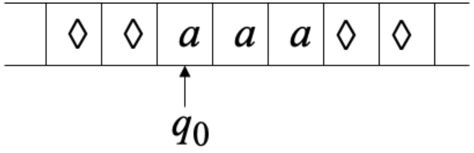
Notar que para aceptar un string de input, no es necesario procesar todos los símbolos en el string

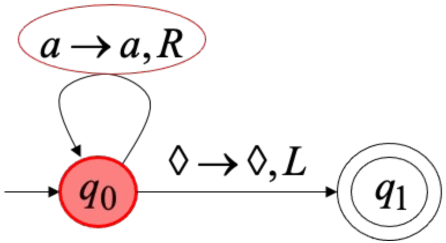
Ejemplo de TM

Sea $\Sigma = \{a, b\}$ y la TM, mostramos la aceptación de la cadena aaa



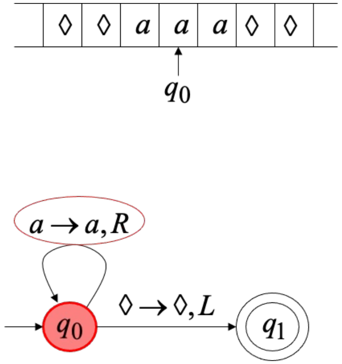
El instante 0 el TM se encuentra así



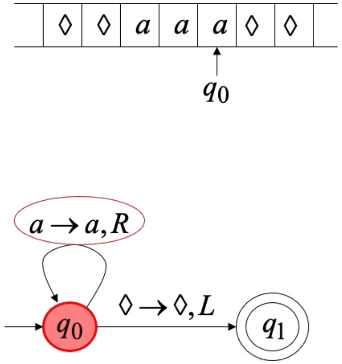


Ejemplo de TM

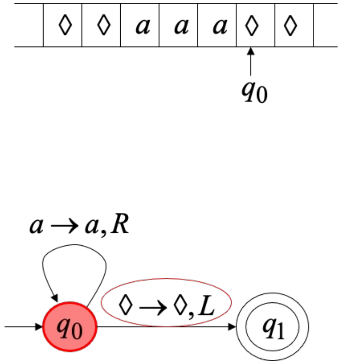
Tiempo 1



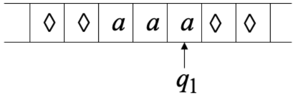
Tiempo 2

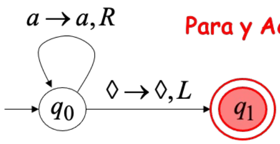


Tiempo 3



Tiempo 4

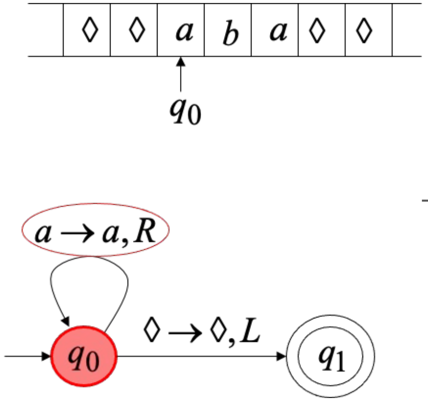




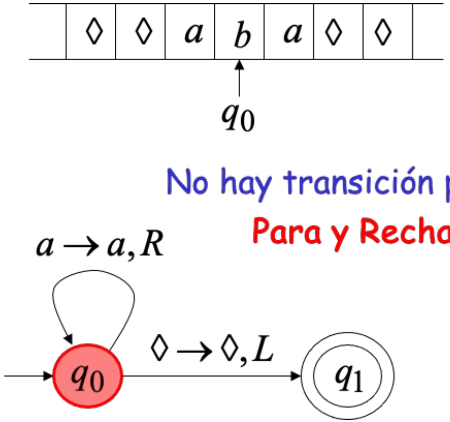
Para y Acepta

Ejemplo de rechazo

Tiempo 0



Tiempo 1

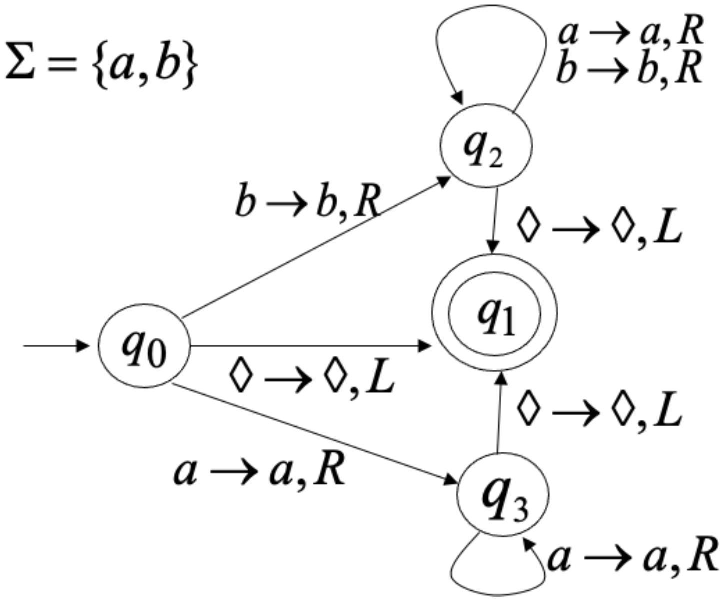


No hay transición posible

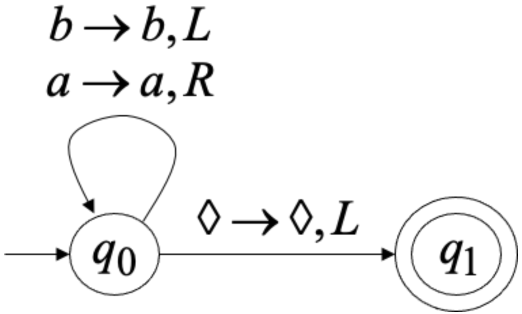
Para y Rechaza

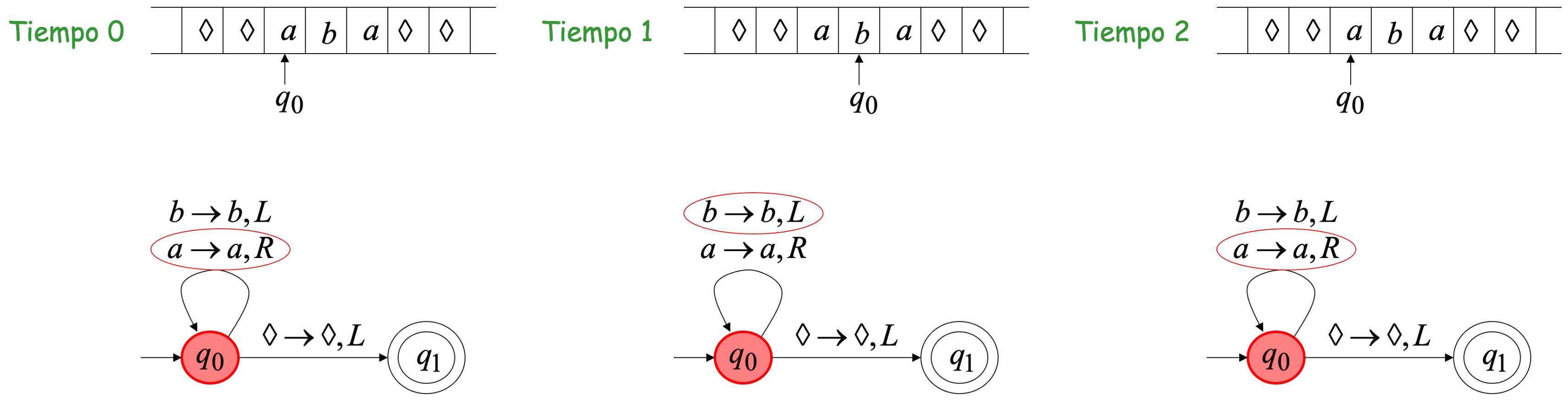
Otro ejemplito

Una Máquina de Turing para el lenguaje $a^*+b(a+b)^*$

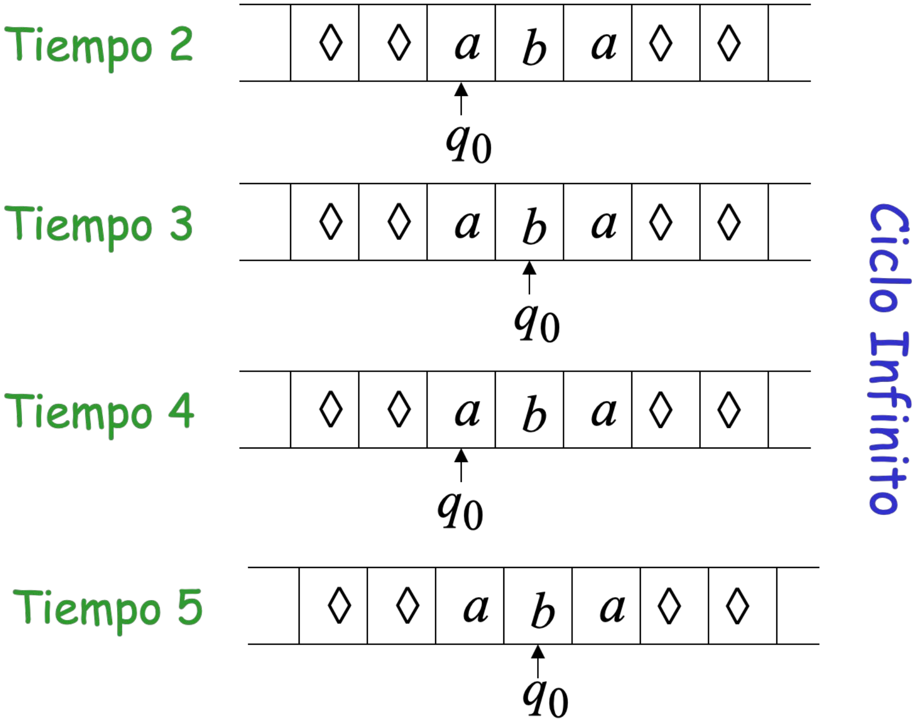


Esta otra TM también acepta $a^*+b(a+b)^*$





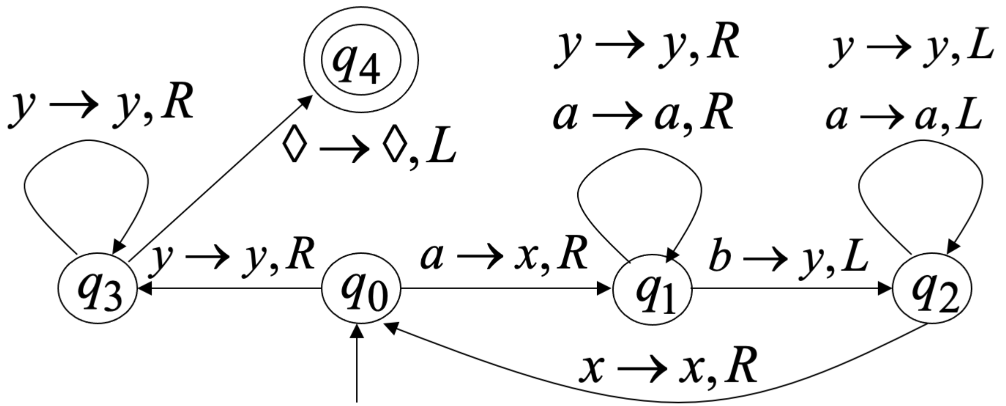
Pero... esta TM produce un ciclo infinito no?



Debido al ciclo infinito, nunca se puede alcanzar el estado final y la máquina nunca para. Por lo que vamos a **rechazar** la cadena de entrada

Otro ejemplo

Una TM para el lenguaje L = { a^n b^n } con n >= 1



La idea es matchear las a's con b's, entonces:

iterativamente

reemplazamos la primer a con x,

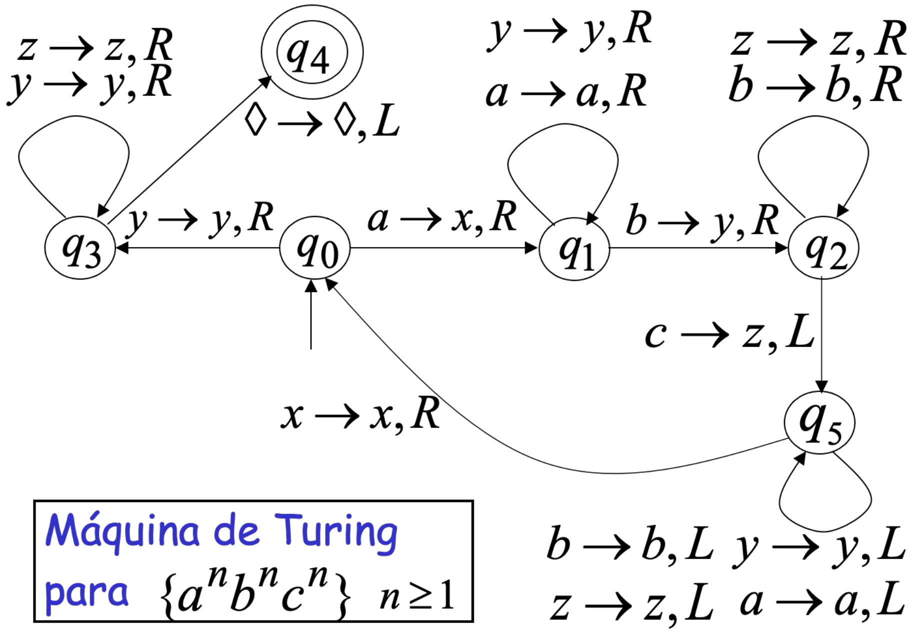
hallar la b más a la izquierda y

reemplazarla con y hasta que no

halla más a's o b's

Si hay una a o una b, rechazamos.

Nota: Si modificamos la máquina para el lenguaje { a^n b^n }, podemos construir { a^n b^n c^n }



Máquina de Turing para $\{a^n b^n c^n\}$ $n \geq 1$

Definición formal - función de transición

$$q_1 \xrightarrow{a \to b, R} q_2 \qquad\qquad q_1 \xrightarrow{c \to d, L} q_2$$

$$\delta(q_1, a) = (q_2, b, R) \qquad\qquad \delta(q_1, c) = (q_2, d, L)$$

Definición formal - Máquina de Turing

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \Diamond, F)$$

Q el conjunto finito de estados del control finito

Σ el alfabeto de símbolos de entrada

Γ el alfabeto de la cinta, $\Sigma \subset \Gamma$

δ la función (parcial) de transición

$$\delta : Q \times \Sigma \to Q \times \Gamma \times \{L, R\}$$

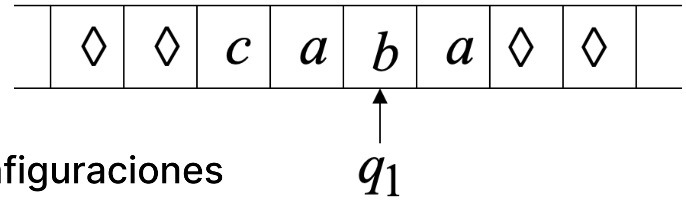
q_0 el estado inicial $q_0 \in Q$

$\Diamond$ el símbolo blanco $\Diamond \in \Gamma, \Diamond \notin \Sigma$

F el conjunto de estados finales $F \subseteq Q$

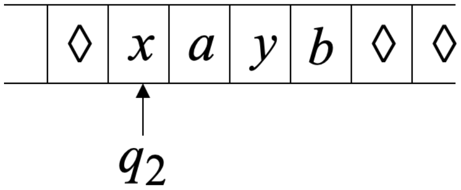
Definición formal - Configuración

Decimos que la configuración caq_1ba denota la configuración instantánea del siguiente input

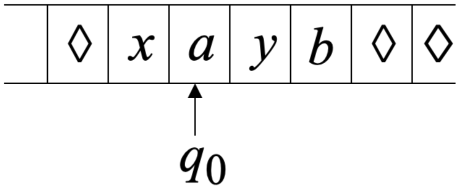


Y dado las siguientes dos configuraciones

Tiempo 4



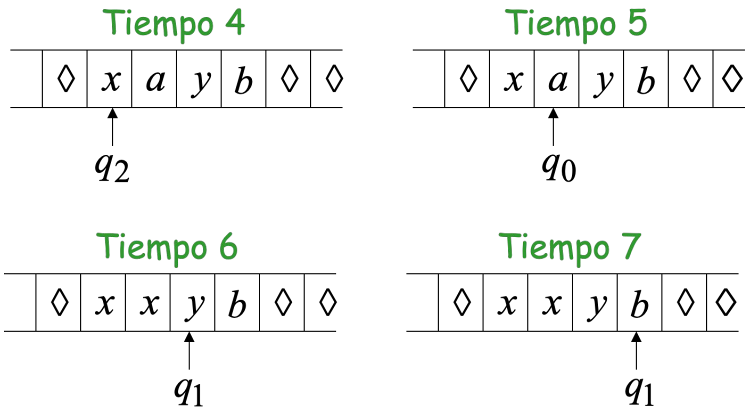
Tiempo 5



Denotamos el movimiento entre 4 y 5 como $q_2xayb \succ xq_0ayb$

Definición formal - Computación

Y denotamos una computación como una secuencia de movimientos

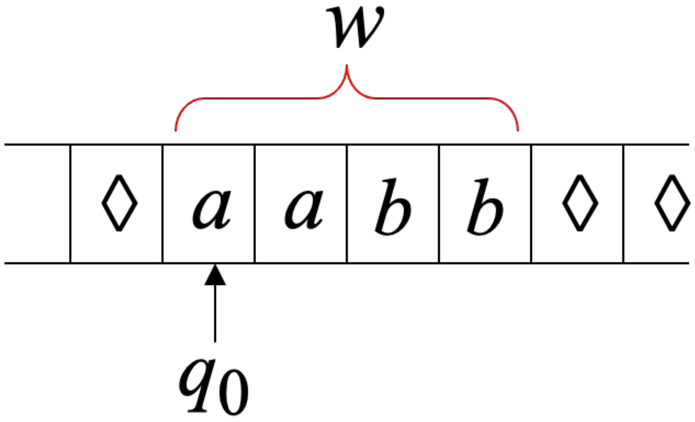


$$q_2xayb \succ xq_0ayb \succ xxq_1yb \succ xxyq_1b$$

$$q_2xayb \overset{*}{\succ} xxyq_1b$$

Definición formal - Configuración inicial

Denotamos q_0w a la configuración inicial para cualquier TM, donde q_0 es el estado inicial y w el input de entrada



Definición formal - Lenguaje aceptado

Para una TM M, denotamos el lenguaje aceptado por M como

$$L(M) = \{w : q_0 w \overset{*}{\succ} x_1 q_f x_2\}$$

Estado Inicial

Estado Final

Si un lenguaje L es aceptado por una TM M, entonces decimos que L es **Turing reconocible**

Nota: **Turing reconocible** es equivalente a decir **Turing aceptable** o **recursivamente enumerable**

Teorema

Un lenguaje L es aceptado por una Máquina de Turing M si y sólo si es generado por una **gramática irrestricta** (sin restricciones), es decir, de tipo 0

La prueba pueden encontrarla en la página 354 de *Theory of Finite Automata with an Introduction to Formal Languages. John Carroll and Darrell Long*