Autómatas de pila

Universidad Nacional de Quilmes

Emanuel Alvaredo, Federico Lochbaum

Los autómatas de pila modelan programas con memoria restringida, es decir, una memoria con estructura de pila (LIFO)

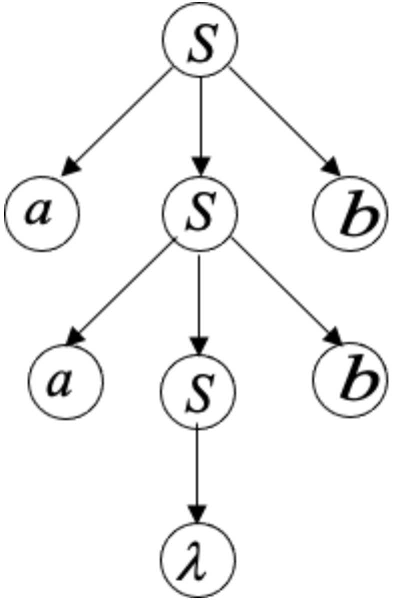
La idea va a ser extender los autómatas finitos (que no eran capaces de “recordar”) con una pila auxiliar

Recordemos **L = $a^n b^n$** cuya estructura no es lineal y requiere cierto criterio de memoria

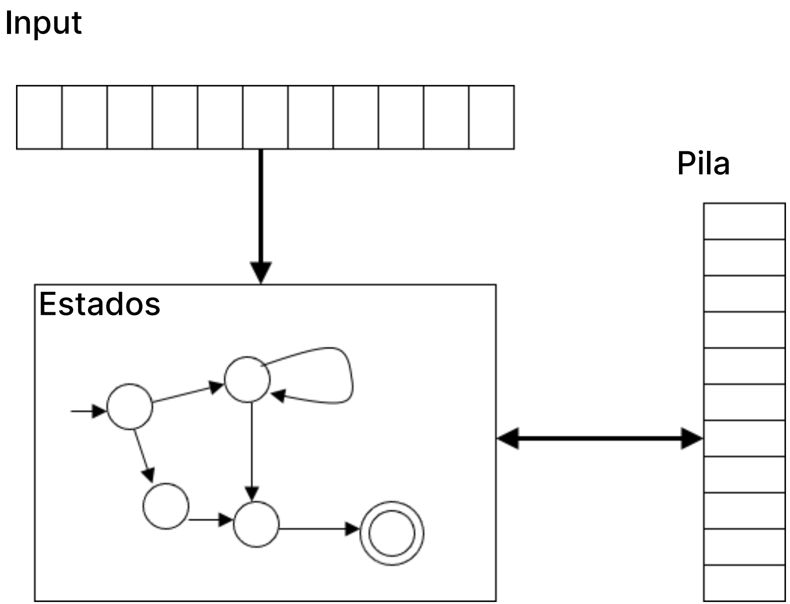
Vemos que **L = $a^n b^n$** no tiene estructura de lista, sino que tiene una estructura arbórea, es decir, para cada **n**, tiene asociado un árbol de **n+1** niveles

“Necesitamos, de alguna manera, recordar el número de a’s leídos

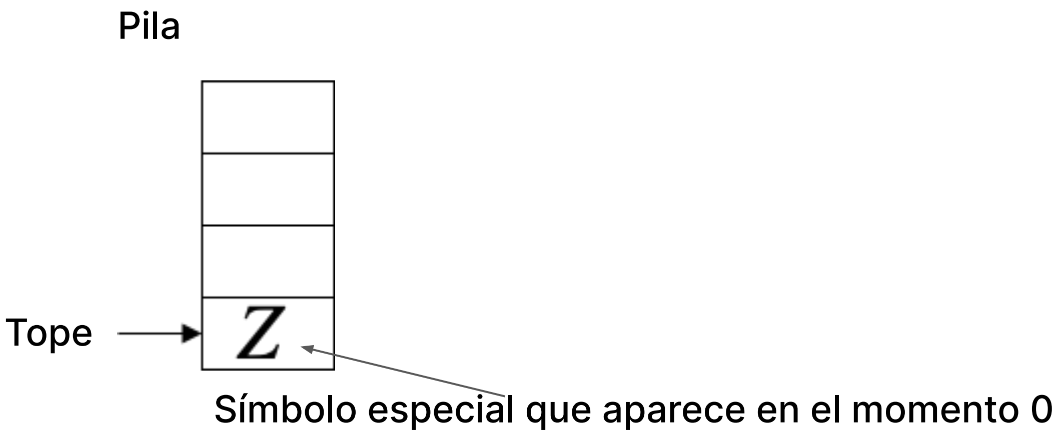
para saber cuantas b’s debe leer”



Autómatas de pila (APs)

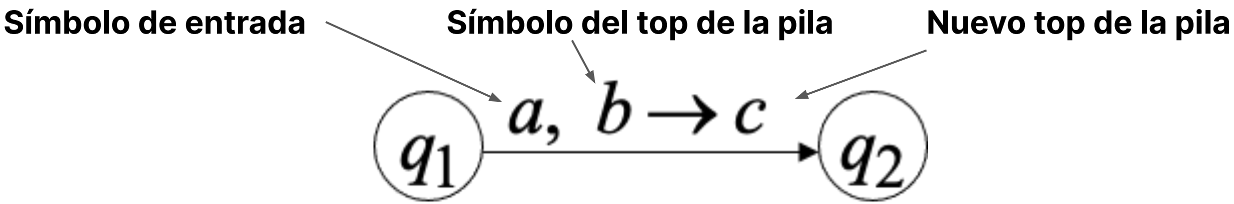


Símbolo inicial de la pila

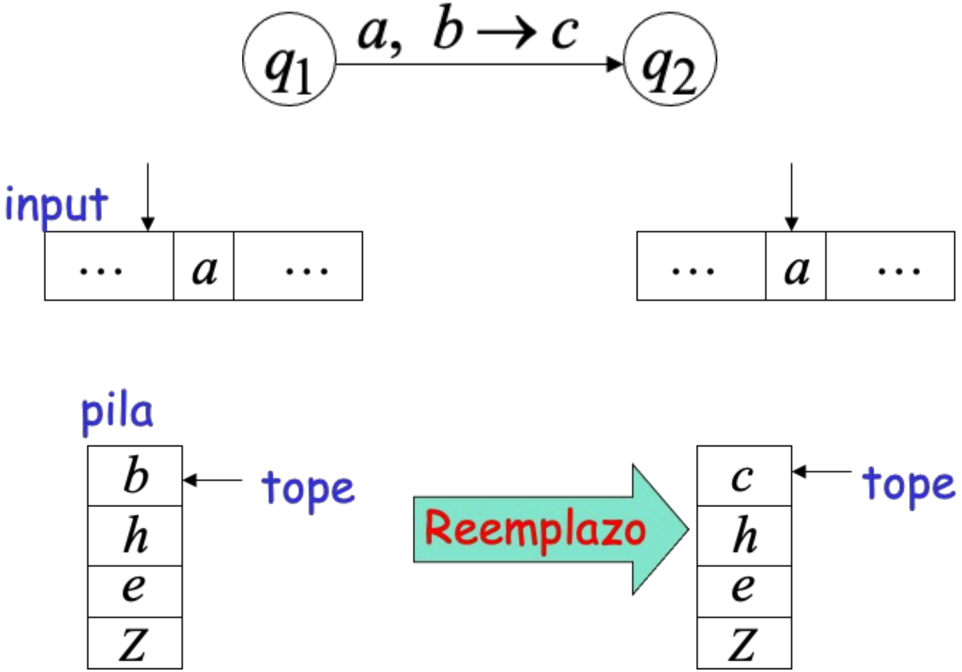


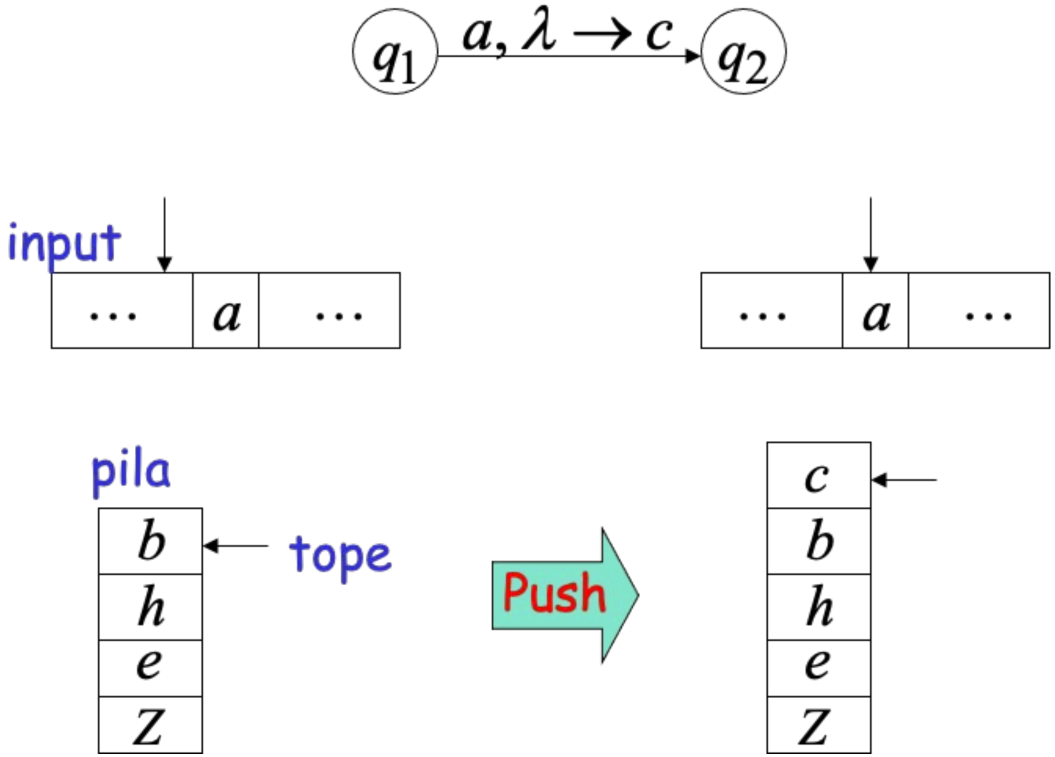
Transición

Una transición de un autómata de pila se verá como

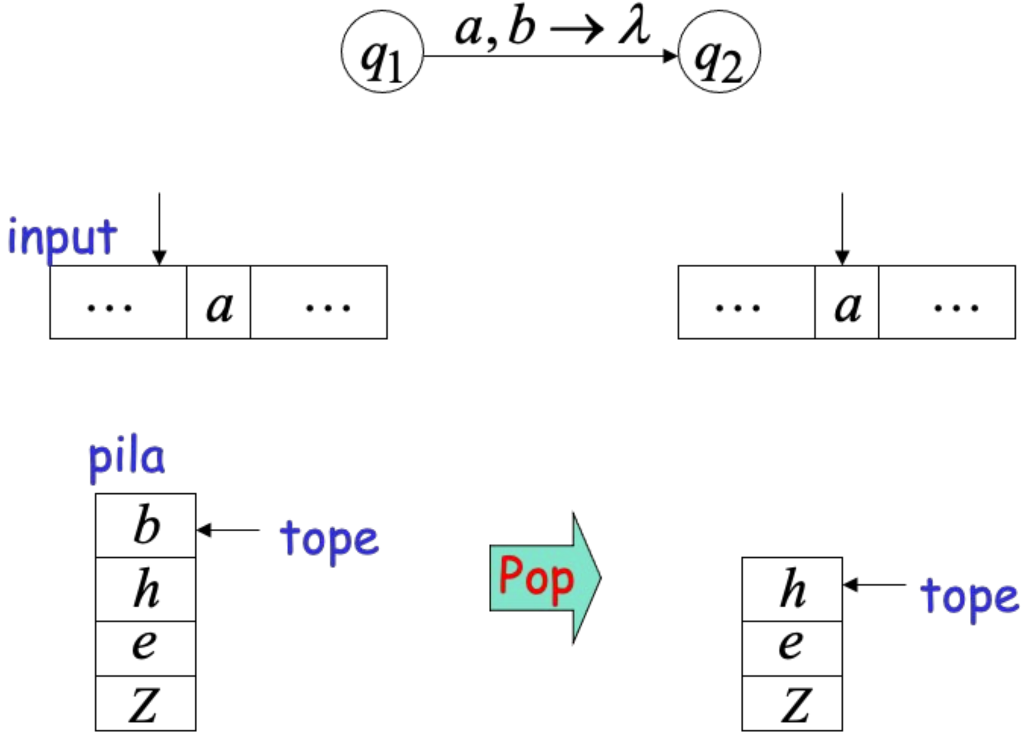


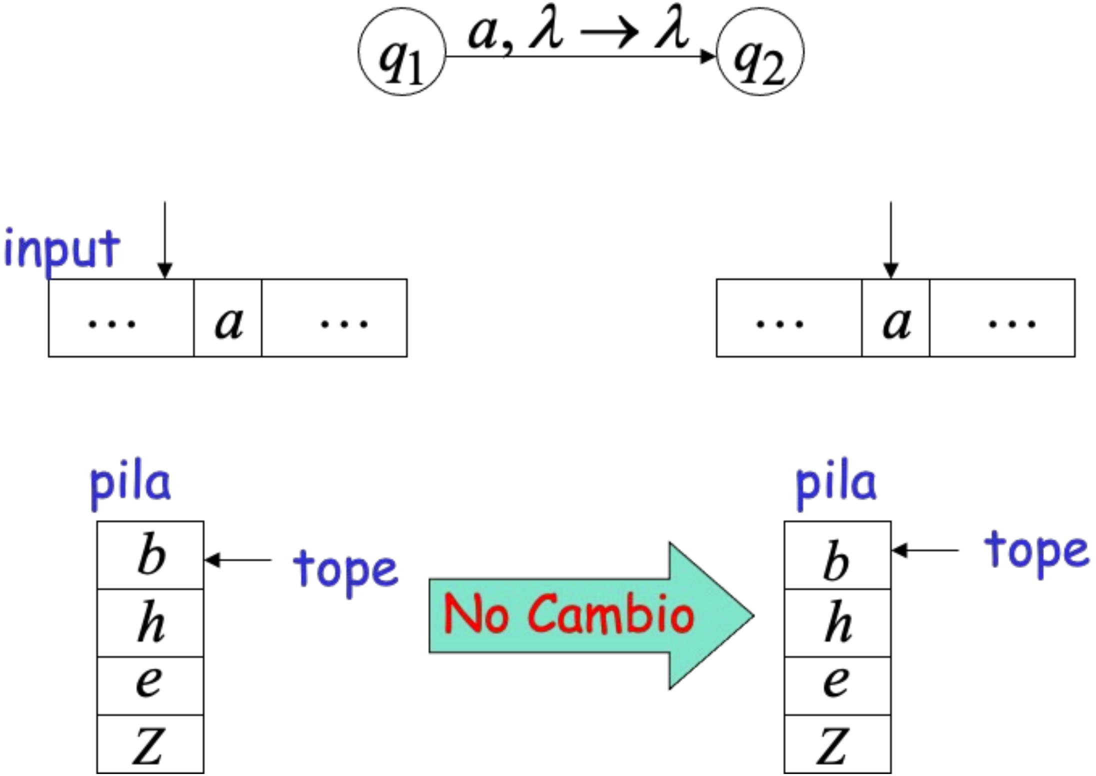
Tipos de transiciones


q_1
$a, \; b \rightarrow c$
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
pila
b
h
e
Z
tope
Reemplazo
c
h
e
Z
tope

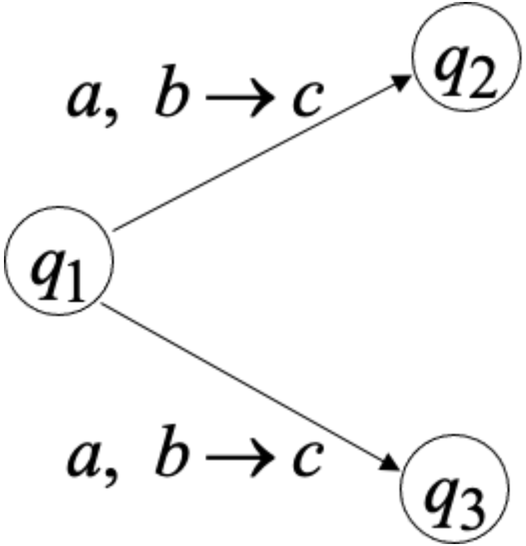

$q_1 \xrightarrow{a, \lambda \to c} q_2$
input
... a ...
pila
b
h
e
Z
tope
Push
... a ...
c
b
h
e
Z

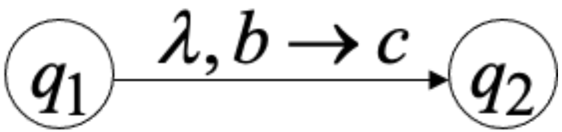
$$a, b \rightarrow \lambda$$




q_1
$a, \lambda \to \lambda$
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
pila
b
h
e
Z
tope
No Cambio
pila
b
h
e
Z
tope

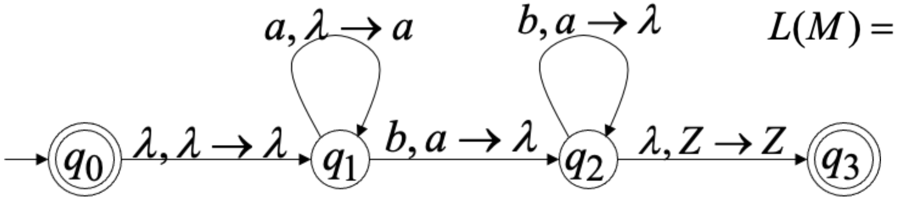
Notar que los autómatas de pila son NO deterministas





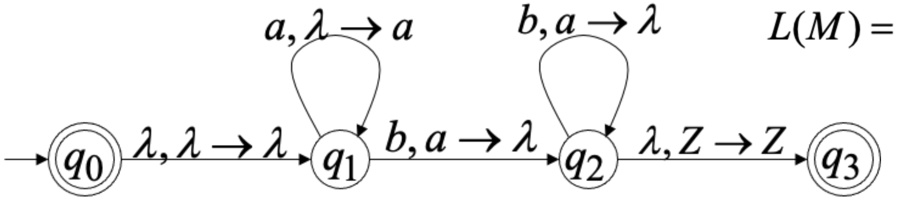
transición - λ

Un ejemplo de AP



$$L(M) = \{a^n b^n : n \geq 0\}$$

Un ejemplo de AP

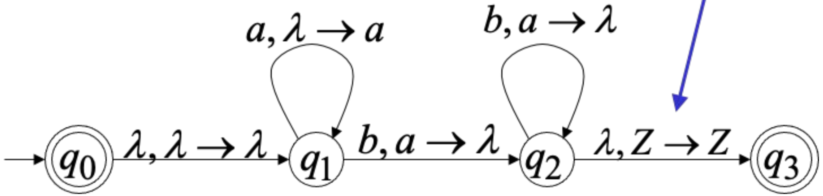


$L(M) = \{a^n b^n : n \geq 0\}$

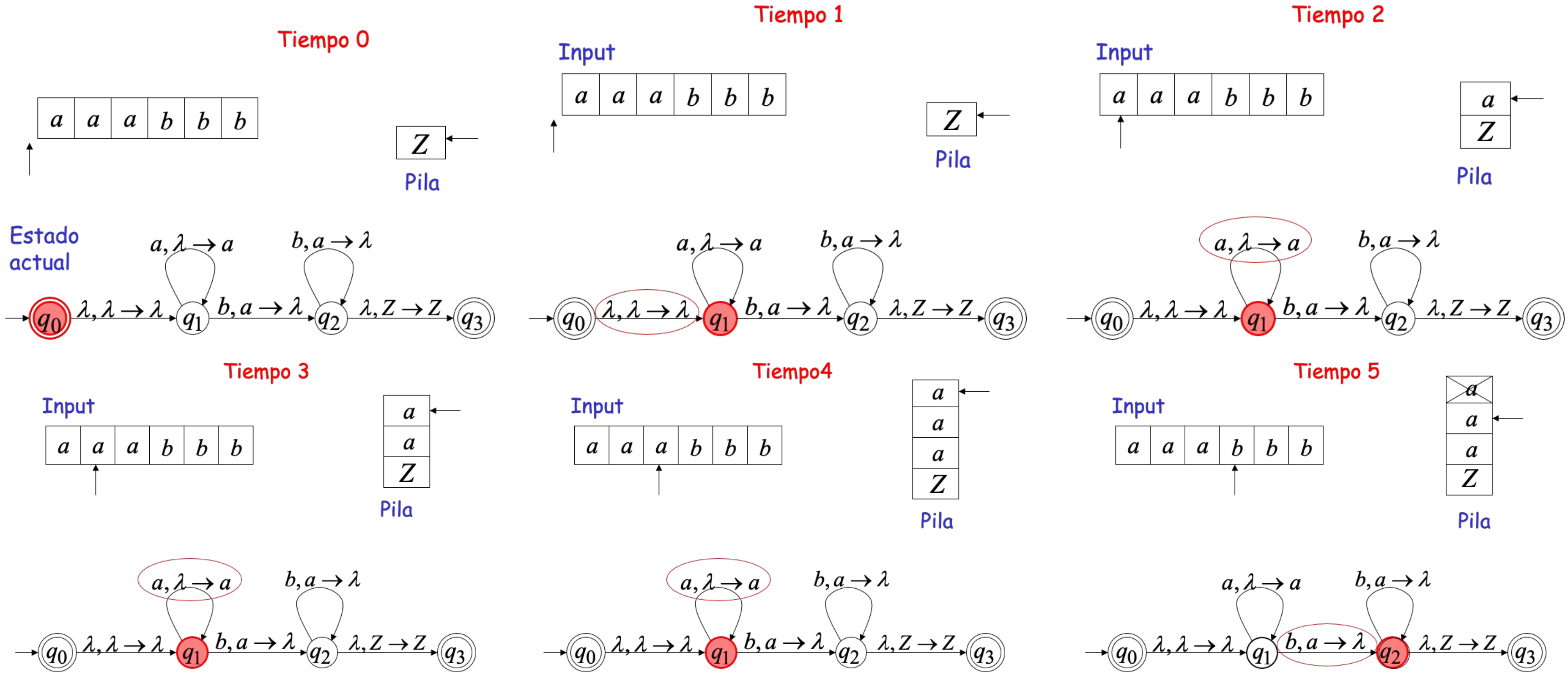
1. Apliar las a's

2. "Matchear" las b's del input con las a's en el tope de la pila

3. correcto!

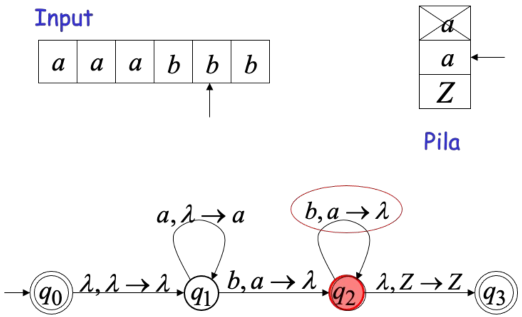


Ejecución

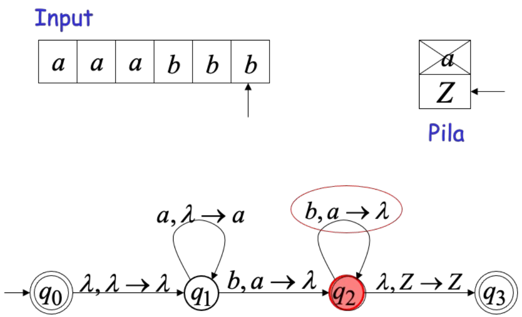

Tiempo 0
a a a b b b
Z
Pila
Estado actual
$a, \lambda \to a$
$b, a \to \lambda$
q_0
$\lambda, \lambda \to \lambda$
q_1
$b, a \to \lambda$
q_2
$\lambda, Z \to Z$
q_3
Tiempo 1
Input
a a a b b b
Z
Pila
Tiempo 2
Input
a a a b b b
a
Z
Pila
Tiempo 3
Input
a a a b b b
a
a
Z
Pila
Tiempo4
Input
a a a b b b
a
a
a
Z
Pila
Tiempo 5
Input
a a a b b b
a
a
a
Z
Pila

Ejecución

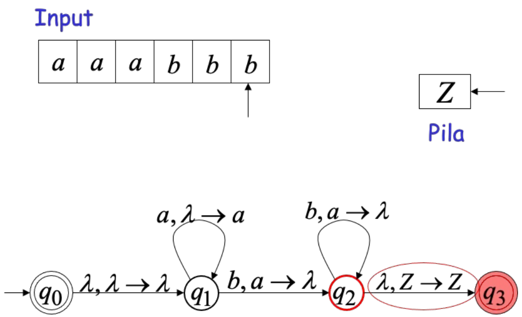
Tiempo 6



Tiempo 7



Tiempo 8



Aceptación y rechazo

Una cadena es aceptada si existe una computación tal que:

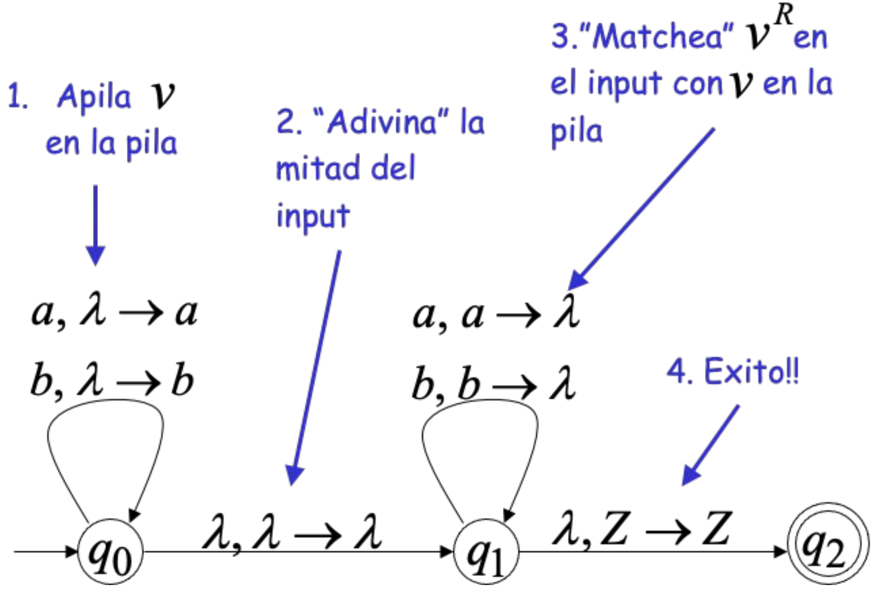
- Todo el input es consumido
- El último estado es un estado final

Nota: No nos interesa el contenido de la pila si la computación termina

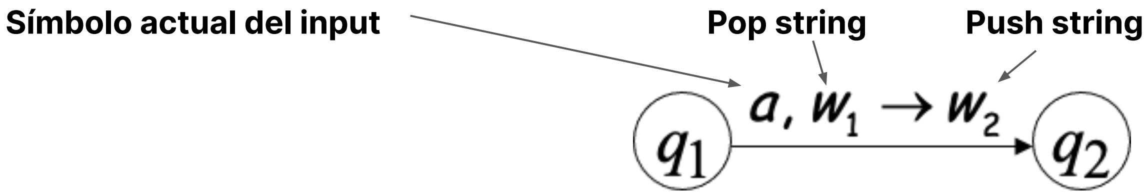
Así como la aceptación funciona igual que en los AFND, el rechazo de una cadena se considera de la misma manera, es decir, que no existe una computación que la acepte

Otro ejemplo

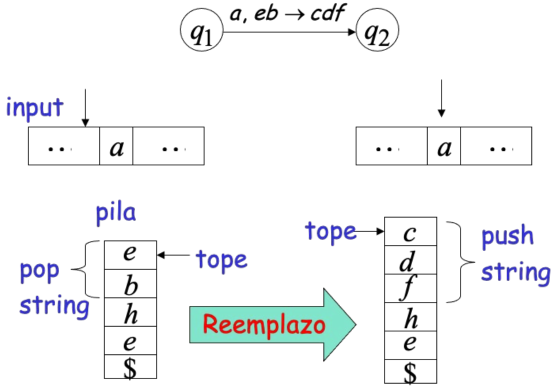
$$L(M) = \{vv^R : v \in \{a,b\}^*\}$$



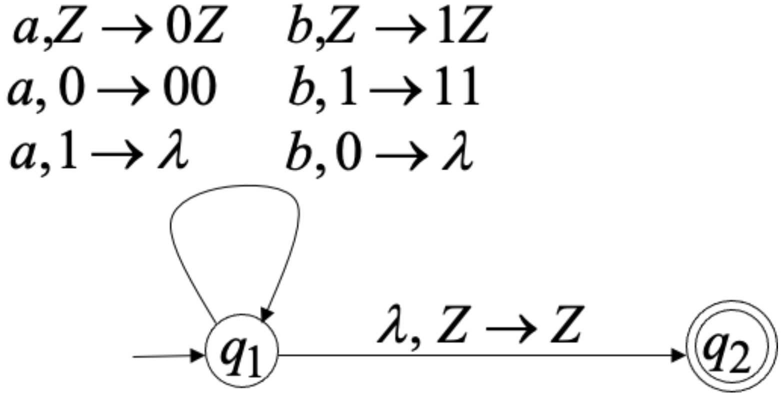
Simplificación: Pushing & Popping strings



Ejemplo

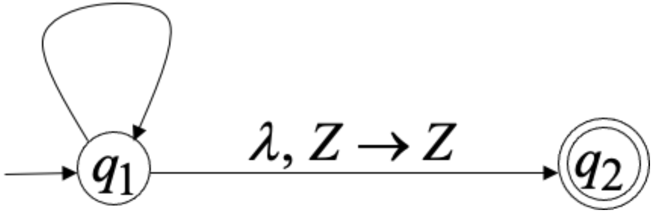


Qué lenguaje reconoce M ?



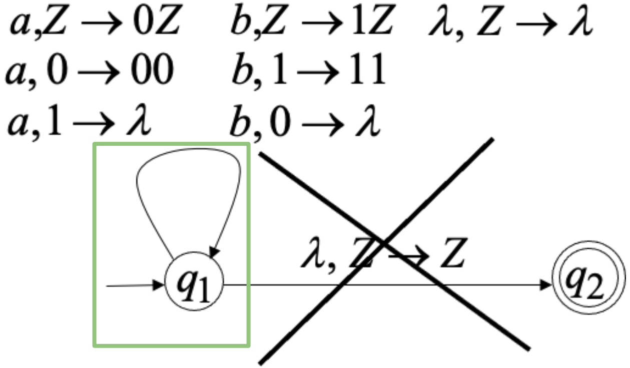
Qué lenguaje reconoce M ?

$$
\begin{array}{ll}
a,Z \rightarrow 0Z & b,Z \rightarrow 1Z \\
a,0 \rightarrow 00 & b,1 \rightarrow 11 \\
a,1 \rightarrow \lambda & b,0 \rightarrow \lambda
\end{array}
$$



$$\boxed{L(M) = \{w \in \{a,b\}^* : \quad n_a(w) = n_b(w)\}}$$

Observar que podríamos reemplazar la condición de alcanzar estado final **por la de pila vacía**. Si se **consumió el input** y la **pila está vacía** el string se **acepta**



Formalidades de los APs

$$M = (\Sigma, \Gamma, S, s_0, \delta, Z, F)$$

Alfabeto de entrada

Alfabeto de la pila

Estados

Estado Inicial

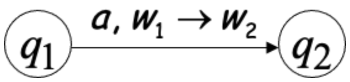
Función de Transición

Símbolo Inicial de pila

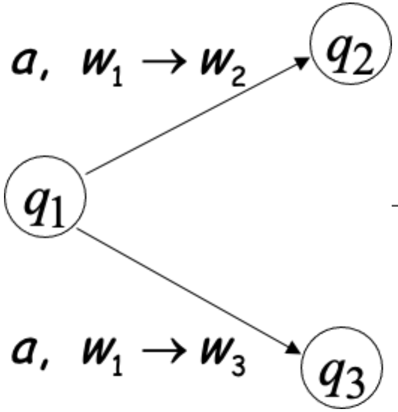
Estados finales

- Σ es el alfabeto de entrada
- Γ es el alfabeto de pila
- S es el conjunto finito y no vacío de estados
- s0 es el estado inicial (s0 $\in$ S)
- δ es la función de transición, δ:S×Σ×Γ→ s(S×Γ)
- Z es el símbolo inicial de pila (Z $\in$ Γ)
- F es el conjunto de estados finales (F $\subseteq$ S).

No determinismo



$$\delta(q_1, a, w_1) = \{(q_2, w_2)\}$$



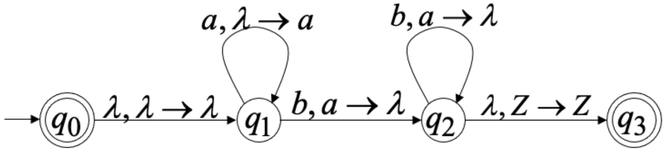
$$\delta(q_1, a, w_1) = \{(q_2, w_2),\ (q_3, w_3)\}$$

Un par ordenado $(\mathrm{t}, \gamma) \in \delta(\mathrm{s, a, A})$ causa un movimiento en el autómata, es decir, un cambio en la configuración actual. Denotamos la **transición de configuraciones** como

$$< \mathrm{s, ay, A}\beta > \quad < \mathrm{t, y}, \gamma\beta > .$$

Y con esto podemos representar una computación de la siguiente manera

$(q_0, aaabbb, Z)$ „ $(q_1, aaabbb, Z)$„
$(q_1, aabbb, aZ)$„ $(q_1, abbb, aaZ)$„ $(q_1, bbb, aaaZ)$„
(q_2, bb, aaZ) „ (q_2, b, aZ)„ (q_2, λ, Z)„ (q_3, λ, Z)



Convenientemente escribimos

$(q_0, aaabbb, Z)$ „* (q_3, λ, Z)

donde ,,* denota la clausura reflexiva transitiva de ,,

Lenguaje aceptado por un AP

Como siempre, el lenguaje L aceptado por un AP M, es decir, L(M)

$$L(M) = \{w: \ (q_0, w, z) \overset{*}{\vdash} (q_f, \lambda, s)\}$$

Estado inicial   Estado final

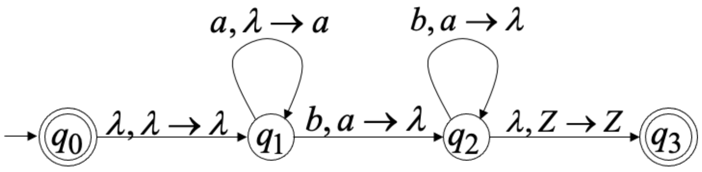
Ejemplo

$$(q_0, a^n b^n, Z) \overset{*}{\vdash} (q_3, \lambda, Z)$$

$$\Downarrow$$

$$a^n b^n \in L(M) \qquad \forall n \geq 0$$

AP M



$$L(M) = \{a^n b^n : n \geq 0\}$$

Lenguaje aceptado por un AP (versión de pila vacía)

Como siempre, el lenguaje L aceptado por un AP M, es decir, Λ(M)

$$\Lambda(M) = \{w : \quad (q_0, w, Z) \quad ^{*} \quad (q, \lambda, \lambda)\}$$

Estado inicial

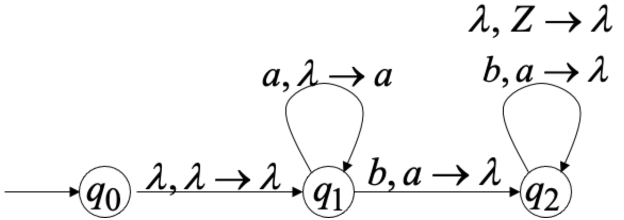
Pila vacía

Ejemplo

$$(q_0, a^n b^n, Z) \overset{*}{„} (q_2, \lambda, \lambda)$$

$$a^n b^n \in \Lambda(M)$$

AP : M



$$\Lambda(M) = \{a^n b^n : n \geq 0\}$$

Equivalencia entre estado final y pila vacía

Teorema

El conjunto de lenguajes reconocidos por el criterio de estado final **es exactamente igual** al conjunto de lenguajes reconocidos por el criterio de pila vacía

- Ida (⇒): Si L = L(P1) para algún AP P1 entonces existe un P2 tal que L = Λ(P2)
- Vuelta (<=): Si L = Λ(P1) para algún AP P1 entonces existe un P2 tal que L = L(P2)

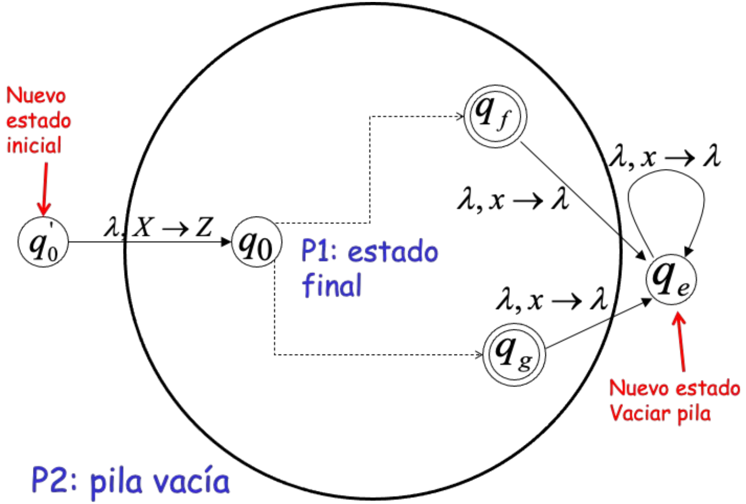
Ida

El conjunto de lenguajes reconocidos por el criterio de estado final **es exactamente igual** al conjunto de lenguajes reconocidos por el criterio de pila vacía

Ida (⇒): Si L = L(P1) para algún AP P1 entonces existe un P2 tal que L = Λ(P2)

Prueba: Lo que se busca es vaciar la pila una vez alcanzado un estado final de P1. Para esto se agregará un estado al que se llega desde los estados finales de P1 por una transición λ y donde se vacía la pila. Además, se debe agregar un nuevo símbolo de pila inicial para evitar que se acepte por pila vacía en otros estados que no son finales en P1, ya que este nuevo símbolo de pila no estará en las transiciones originales de P1. Finalmente un estado inicial que apila el símbolo inicial de P1.

Prueba: Lo que se busca es vaciar la pila una vez alcanzado un estado final de P1. Para esto se agregará un estado al que se llega desde los estados finales de P1 por una transición λ y donde se vacía la pila. Además, se debe agregar un nuevo símbolo de pila inicial para evitar que se acepte por pila vacía en otros estados que no son finales en P1, ya que este nuevo símbolo de pila no estará en las transiciones originales de P1. Finalmente un estado inicial que apila el símbolo inicial de P1.



Transformación

Sea **P1 =< Σ, Γ, S, q0, δ, Z, F >**. Se define **P2 =< Σ, Γ ∪ {X}, S ∪ {qe, q0' }, q0' , δ2, X, ∅ >**, donde:

- $\delta 2(q0', \lambda, X) = \{< q0, ZX >\}$
- $(\forall s \in S)(\forall a \in (\Sigma))(\forall A \in \Gamma)\ \delta(s, \lambda, A) \subseteq \delta 2(s, \lambda, A)$
- $(\forall s \in F)(\forall A \in (\Gamma \cup \{X\}))\ < qe, \lambda > \in \delta 2(s, \lambda, A)$
- $(\forall A \in (\Gamma \cup \{X\}))\ < qe, \lambda > \in \delta 2(qe, \lambda, A)$
- No hay ninguna transición adicional a las especificadas

Nota: Aunque P1 sea determinístico, P2 podría no serlo.

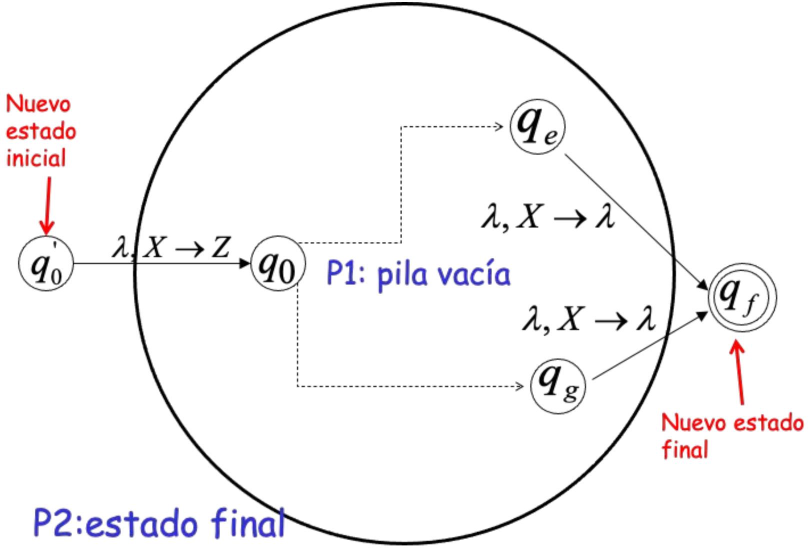
Vuelta

El conjunto de lenguajes reconocidos por el criterio de estado final **es exactamente igual** al conjunto de lenguajes reconocidos por el criterio de pila vacía

Vuelta (<=): Si L = Λ(P1) para algún AP P1 entonces existe un P2 tal que L = L(P2)

Prueba: Es necesario detectar cuando se vacía la pila y, en ese caso, llevar a un estado final. (Este estado final no pertenece al conjunto de estados de P1). Para esto se agrega un nuevo símbolo inicial de pila, que al aparecer al tope de la pila en cualquier estado de P1 se debe llevar hacia el estado final por una transición λ

Prueba: Es necesario detectar cuando se vacía la pila y, en ese caso, llevar a un estado final. (Este estado final no pertenece al conjunto de estados de P1). Para esto se agrega un nuevo símbolo inicial de pila, que al aparecer al tope de la pila en cualquier estado de P1 se debe llevar hacia el estado final por una transición λ



Transformación

Sea **P1 =< Σ, Γ, S, q0, δ, Z, F >**. Se define **P2 =< Σ, Γ ∪ {X}, S ∪ {qf , q0’ }, q0’ , δ2, X, {qf } >**, donde:

- δ2(q0’ , λ, X) = {< q0, ZX >}
- (∀s∈S)(∀a∈Σ)(∀A∈Γ) δ2(s, a, A)=δ(s, a, A)
- (∀s ∈ S) δ2(s, λ, X) = {< qf , λ >}