Sintaxis y Semántica del lenguaje.

Tema: Autómatas Finitos de Pila

y

Máquinas de Turing

Autómata finito de pila (AFP)

- Los autómatas finitos no son lo suficientemente poderosos para aceptar los lenguajes libres de contexto (LLC).

Vamos a analizar, qué tipo de autómata necesitaríamos para abordar los siguientes lenguajes:

1)Lenguaje de los paréntesis bien balanceados, en donde deberíamos distinguir las palabras con paréntesis bien balanceados, por ejemplo: (( ))( ), de las palabras con paréntesis desbalanceados.

Autómata finito de pila (AFP)

1) En este caso, si tomamos como ejemplo: Ej: (( ))( )

podríamos llevar un registro con un **contador de paréntesis** que abren y de paréntesis que cierran.

Al finalizar el análisis de la secuencia, el cálculo debería dar 0.

En el ejemplo dado:

(( )) () el registro tomaría los siguientes valores 1 2 1 0 1 0

Cada vez que viene un paréntesis que abre, el contador se incrementa en 1, y cuando viene un paréntesis que cierra, se decrementa en 1.

Autómata finito de pila (AFP)

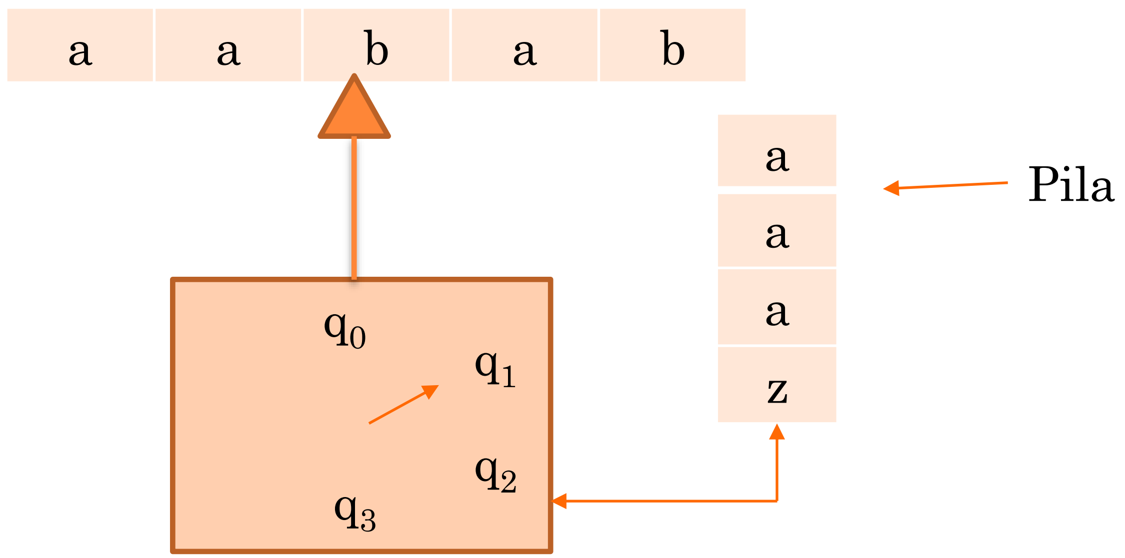
2)Lenguaje de los palíndromos, en donde se deben reconocer palabras que se escriben igual al derecho y al revés, como por ejemplo: ANITALAVALATINA.

En este caso, la máquina contadora no funcionaría porque se necesitaría recordar la 1° mitad de la palabra para poder compararla con la 2° mitad.

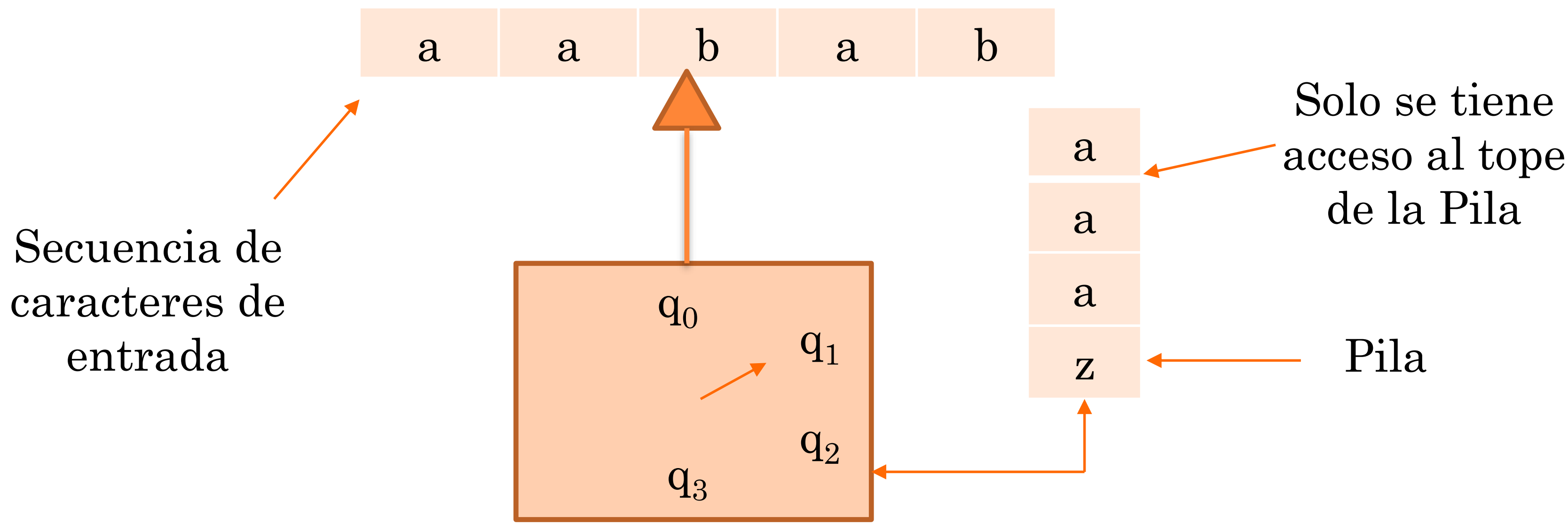
Podríamos pensar entonces en una máquina capaz de recordar cadenas de caracteres arbitrarias y no números.

AUTÓMATA FINITO DE PILA (AFP)

Esto puede lograrse añadiendo al AF un almacenamiento auxiliar llamado pila, es por ello que se llama **Autómata finito de Pila**. En dicha pila se pueden ir depositando carácter a carácter cadenas arbitrariamente grandes.



Autómata finito de pila (AFP)



La **pila** tiene un **alfabeto propio**, que puede o no coincidir con el alfabeto de la palabra de entrada, porque puede ser necesario agregar a la pila caracteres especiales, según las necesidades de diseño del autómata.

Funcionamiento del AFP

- Al principio la pila está vacía (estado z_0).

Durante el funcionamiento del AFP, la pila almacena y desapila caracteres de acuerdo a las transiciones ejecutadas.

Para aceptar una palabra la pila debe estar nuevamente vacía.

En los AFP las **transiciones** de un estado a otro indican:

- **los caracteres que se consumen de la entrada**
- **lo que se saca del tope de la pila**
- **lo que se pone en la pila.**

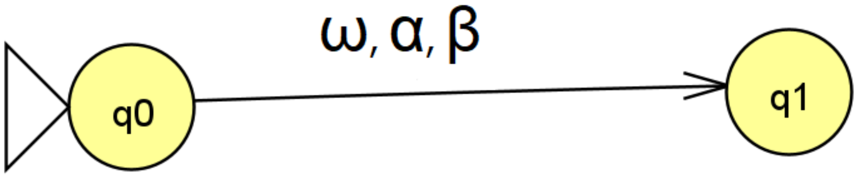
Funcionamiento del AFP

Para las transiciones usamos la notación **"ω/ α /β"**

ω es la secuencia de caracteres de entrada que se consume

α es lo que se saca de la pila

β es lo que se pone en la pila

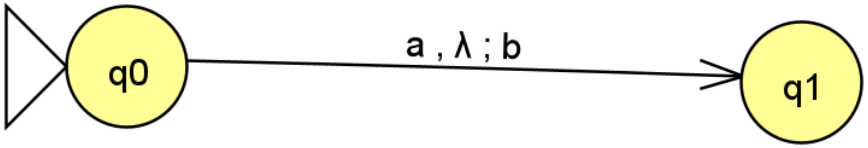


1° se saca de la pila y luego se pone.

Funcionamiento del AFP

Si aparece **ε** en una transición significa que no se pone o no se saca nada de la pila. **ε** también se puede representar con **λ**.

Por ejemplo: La transición **"a/ε/b"** , indica que se consume de la entrada un carácter **a**, no se saca nada de la pila y se pone **b** en la pila.



Funcionamiento del AFP

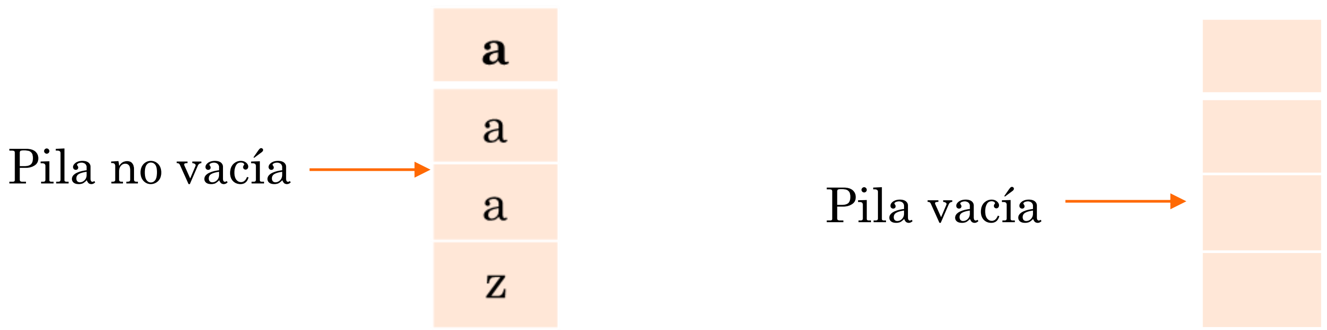
Al igual que los AF, los AFP tienen estados finales, que permiten distinguir cuando una palabra de entrada es aceptada.

Para que una **palabra de entrada sea aceptada** en un AFP se deben cumplir las siguientes condiciones:

1) **La palabra de entrada se debe haber consumido.**
2) **El AFP se debe encontrar en un estado final.**
3) **La pila debe estar vacía.**

Funcionamiento del AFP

El símbolo (z_0) de la pila, que vamos a simbolizar con la letra **z** marca el fondo de la pila y permite comprobar que la pila está vacía.



Cuando la palabra de entrada se termina y en la pila quedó sólo una z se la saca y entonces la pila queda vacía.

Formalización de un AFP

Un autómata de pila es un séxtuplo $(K, \sum, \sigma, \Delta, S, F)$ donde

K es un conjunto de estados

$\sum$ es el alfabeto de entrada

σ es el alfabeto de la pila

S es el estado inicial

F es el conjunto de estados finales

$\Delta \subseteq (K \times \sum^* \times \sigma^*) \times (K \times \sigma^*)$ es la relación de transición

Diseño de un AFP

Ejemplo 1:

Diseñar el **AFP** para el lenguaje **L={ω / ω=$a^n b^n$ con n>=1}**

Las cadenas deben ser de la forma:

ω =ab

ω =aabb

ω =aaabbb → Tienen igual cantidad de a's que b's

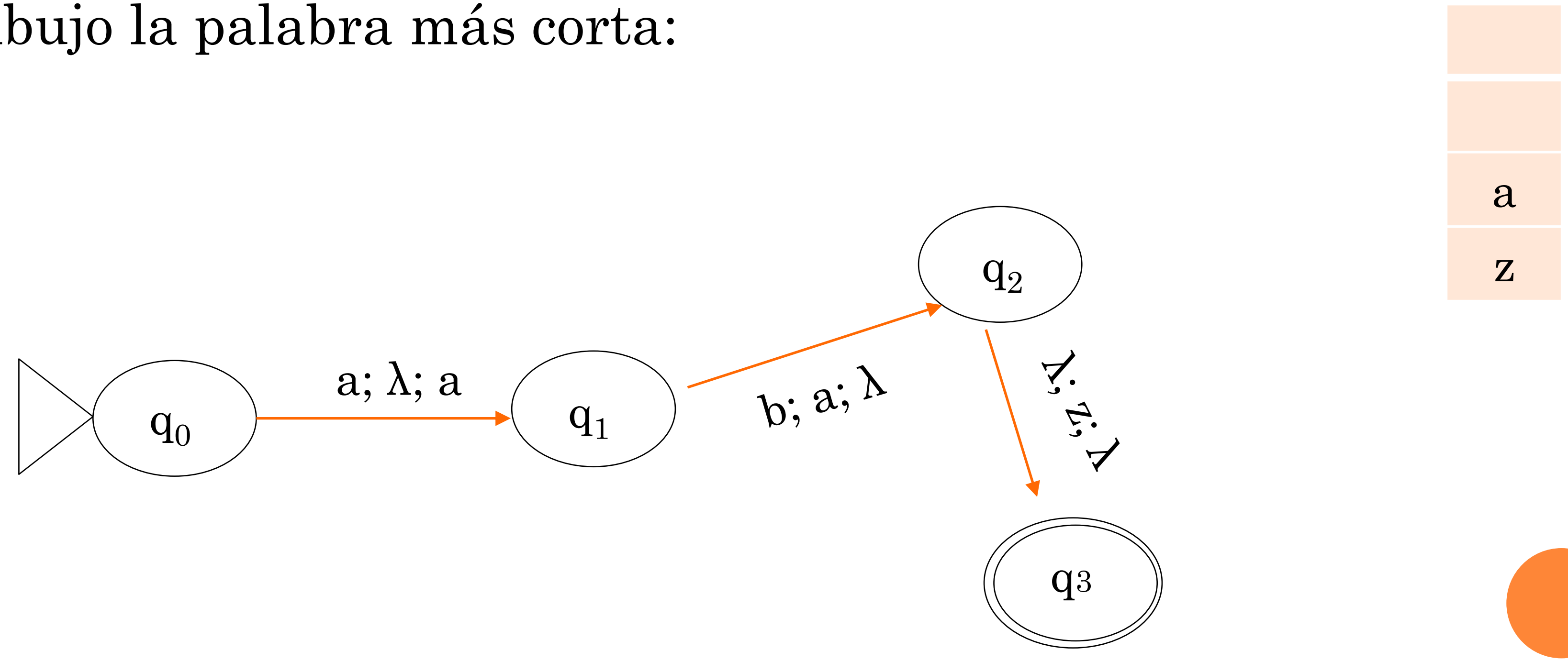
Al diseñar el AFP habrá que ver qué pasa cuando llega una **a** y qué pasa cuando llega una **b**.

Vamos a utilizar la pila como contador de cantidad de a's que se consumen y luego confrontarla con la cantidad de b's.

DISEÑO DE UN AFP

Se parte del estado inicial z, cada vez que llega una **a** se coloca en el tope de la pila, hasta que llegue la 1° **b** y comienzan a desapilarse las **a.**

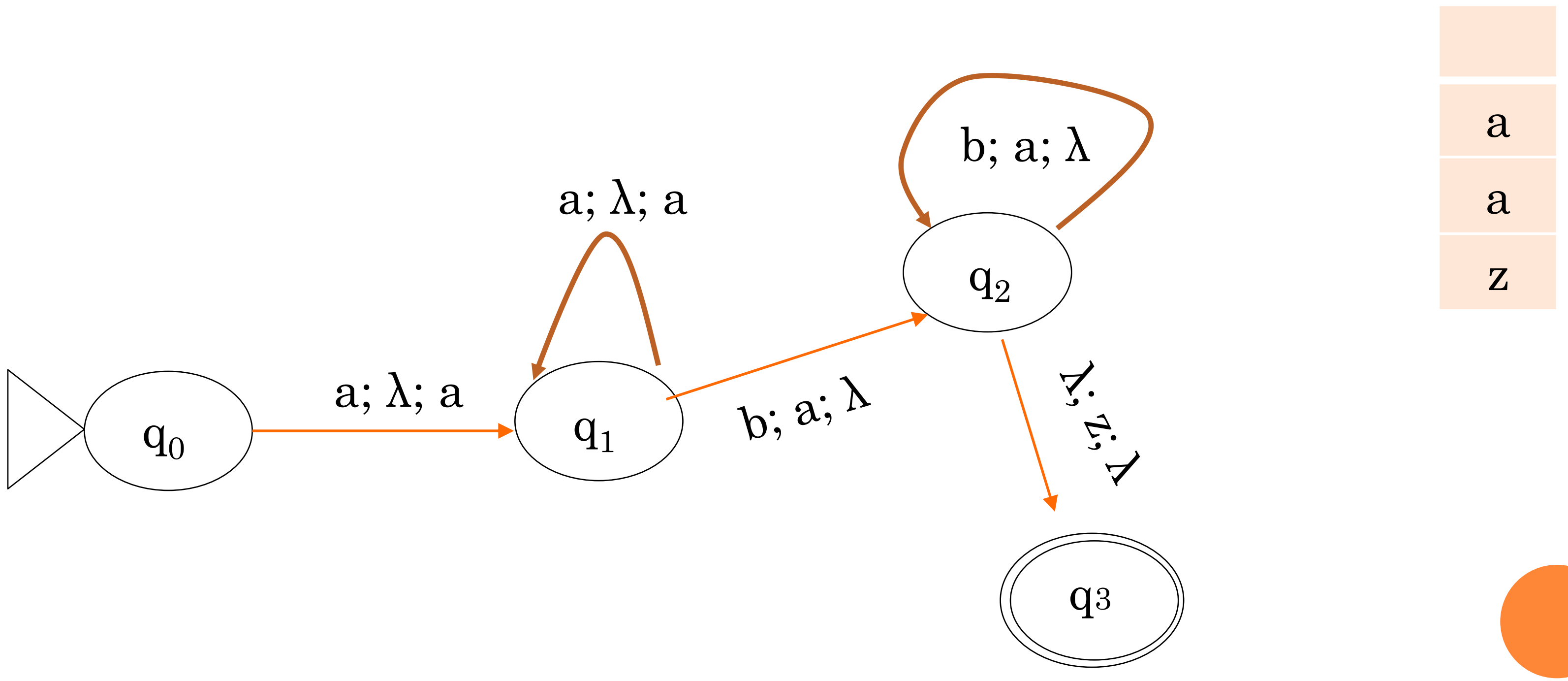
Dibujo la palabra más corta:



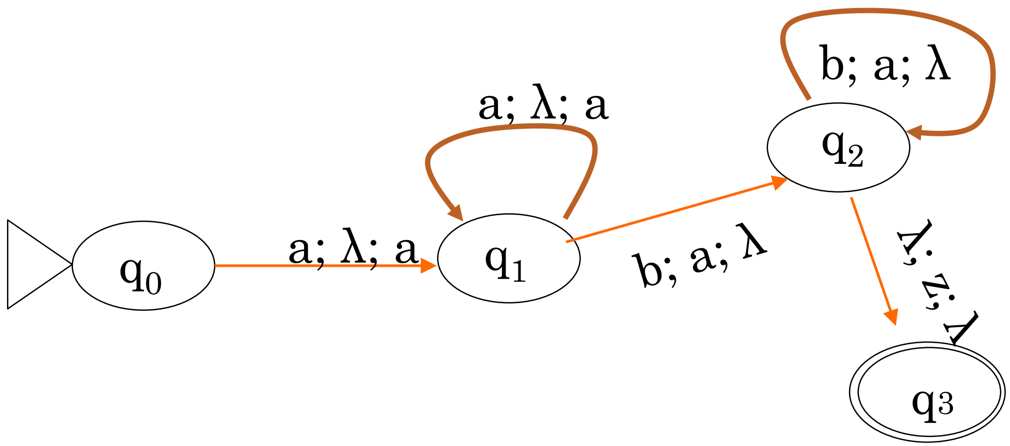
DISEÑO DE UN AFP

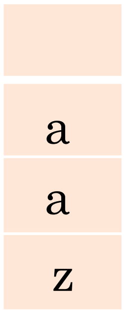
Completo el AFP: cada vez que llega una **a** se apila, hasta que llegue la 1° **b** y comienzan a desapilarse las **a.**

Al terminar la cadena, la pila debe estar vacía.



Diseño de un AFP





1°) Al principio la pila está vacía, en el fondo de la pila está **z**.

2°) Lee una **a** → apila una **a,** se repite por cada **a** leída.

3°) Lee una **b** → desapila una **a**, se repite por cada **b** leída

4°) Finaliza la cadena → sin leer símbolo de entrada, saca **z** y se vacía la pila.

Por lo tanto, la palabra es aceptada.

Diseño de un AFP

Utilizaremos una “traza de ejecución” para verificar

el funcionamiento del autómata para la cadena **aabb**.

Una “traza de ejecución”, contiene tres columnas:

Estado: representa el estado en el que se encuentra el autómata.

Por leer: es lo que falta por leer de la palabra de entrada.

Pila: es el contenido de la pila.

Estado	Por leer	Pila
q0	aabb	z
q1	abb	a
q1	bb	aa
q2	b	a
q2	ε	z
q3	-----	vacía

GLC ASOCIADA AL LENGUAJE L

Dado el lenguaje **L={ω / ω=$a^n b^n$ con n>=1}** determinar cuál es la GLC asociada a dicho lenguaje y al AFP recién construído

Las cadenas son de la forma:

ω =ab

ω =aabb

ω =aaabbb

Entonces la GLC es:

S → ab verifica la palabra más corta

S → aSb verifica las restantes palabras

La última regla tiene 3 símbolos combinados.

Diseño de un AFP

Ejemplo 2:

Diseñar el AFP para el lenguaje **L={ω / ω=0^n1^{2n} con n>=1}**

Las cadenas son de la forma:

ω=011   n=1

ω =001111   n=2 (se tiene el doble de 1 que de 0)

En la pila se parte del estado inicial z_0 y cada vez que llega un 0 se coloca en el tope de la pila, hasta que llegue el primer 1, en ese momento no se saca ni se pone nada en la pila.

Diseño de un AFP

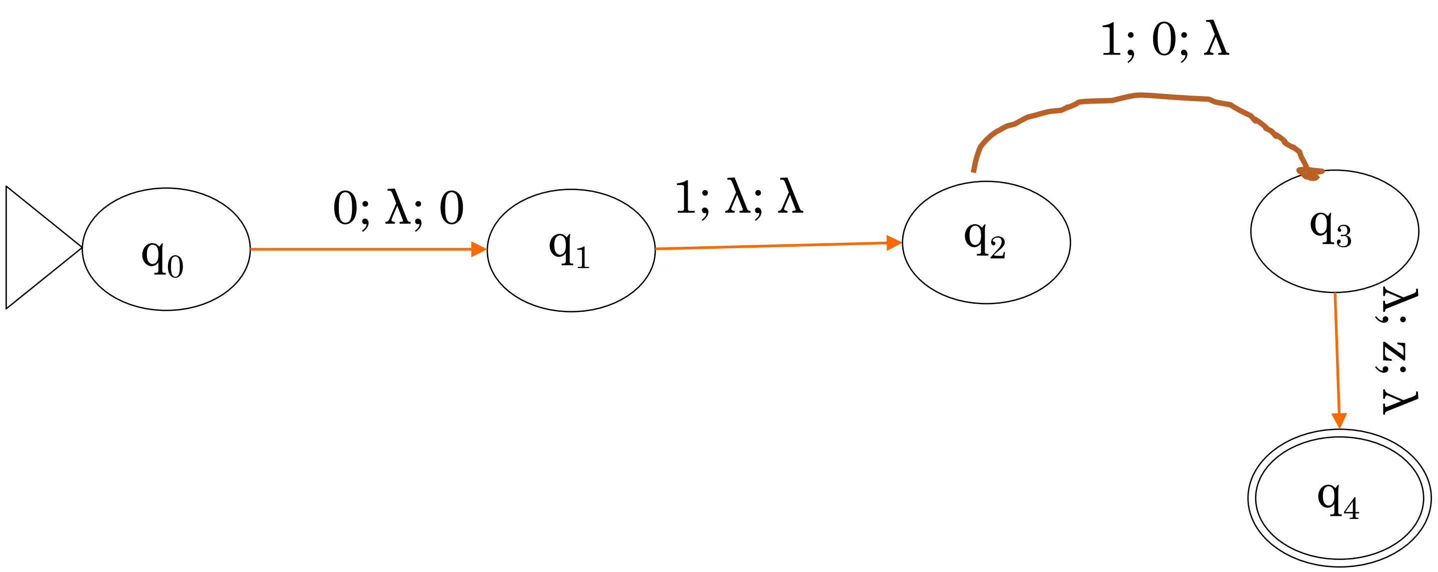
Cuando se lee el 2° 1 sucesivo recién se desapila un 0. Es decir se desapila un 0 por cada dos 1 seguidos que vengan.

Al terminar la cadena, la pila debe estar vacía

a) ω=011 n=1

a)

0
z



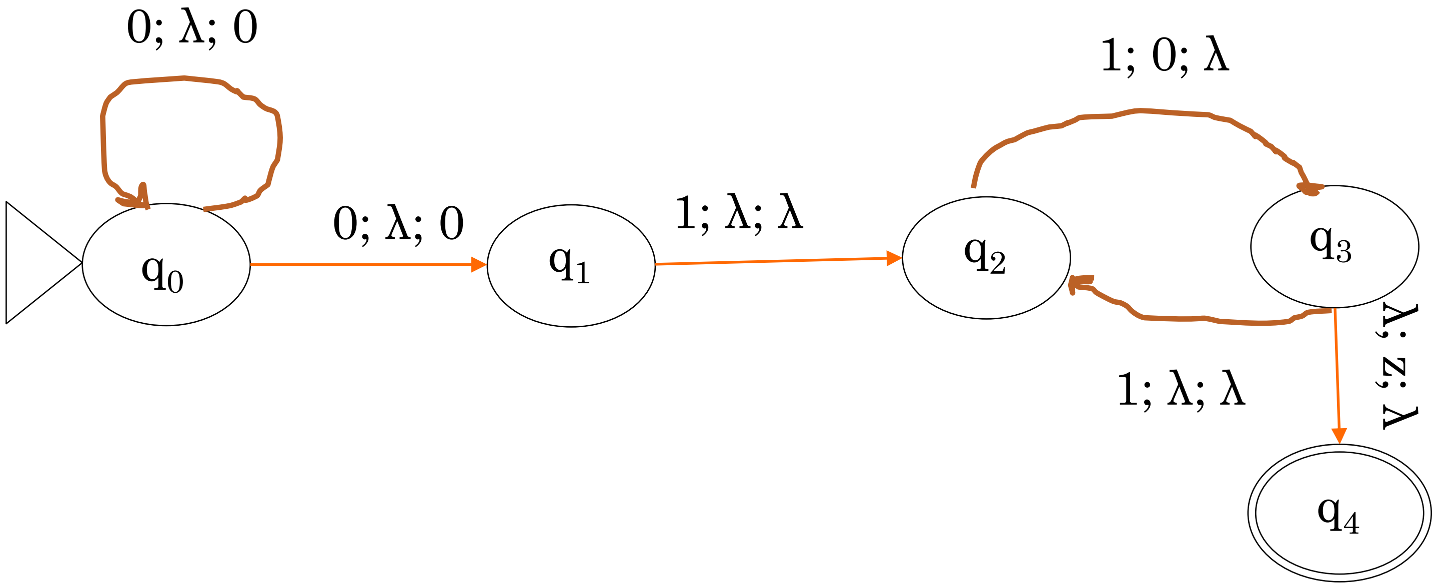
Diseño de un AFP

Completo el AFP para que valide el resto de las palabras de L:

b) $\boldsymbol{\omega}$=001111 n=2

b)

0
0
z



GLC ASOCIADA AL LENGUAJE L

Siendo **L={ω / ω=0^n1^{2n} con n>=1}** donde las cadenas son de la forma:

ω1=011   n=1

ω2 =001111   n=2 (se tiene el doble de 1 que de 0)

¿Cuál sería la GLC asociada a L?

S→ 0A
A→ 1B
B→ 1

con las primeras tres reglas se genera la palabra más pequeña ω1

A → SA

con esta regla se logra que se repita el 0 y con A se repitan los dos 1 siguientes, y de esa forma se validan el resto de las palabras de L

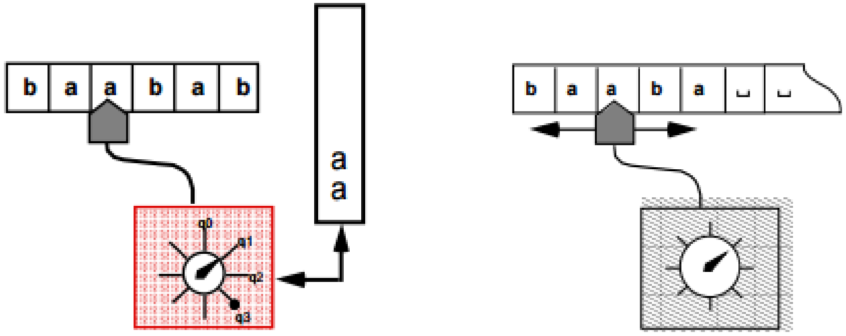
(esta gramática usa dos no terminales seguidos)

Máquina de Turing

- Alan Turing propuso en los años 30 un modelo de máquina abstracta, como una extensión de los autómatas finitos.
- La **máquina de Turing** es particularmente importante porque es la más poderosa de todas las máquinas abstractas conocidas.

Funcionamiento de la Máquina de Turing

- La máquina de Turing tiene, como los autómatas que hemos visto antes, un **control finito**, una **cabeza lectora** y una **cinta** donde puede haber **caracteres**, y donde eventualmente viene la **palabra de entrada**.



(a) Autómata de pila (b) Máquina de Turing

Funcionamiento de la Máquina de Turing

En la MT la cabeza lectora es de **lectura y escritura**, por lo que la cinta puede ser modificada en curso de ejecución. Además, en la MT **la cabeza se mueve bidireccionalmente** (izquierda y derecha).

- La operación de la MT consta de los siguientes pasos:
  1. Lee un carácter en la cinta
  2. Efectúa una transición de estado
  3. Realiza una acción en la cinta :
     - Escribe un símbolo en la cinta, o
     - Mueve la cabeza a la izquierda o a la derecha

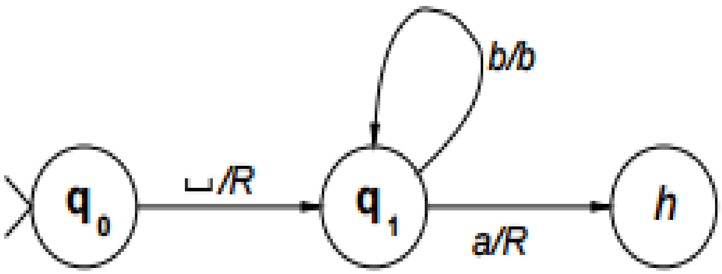
Funcionamiento de la Máquina de Turing

- La palabra de entrada en la MT está escrita inicialmente en la cinta, pero iniciando a partir de la segunda posición de la cinta, siendo el primer cuadro un carácter blanco.

- Al iniciar la operación de la MT, la cabeza lectora está posicionada en el carácter blanco a la izquierda de la palabra de entrada. Procesa carácter a carácter.

- Decimos que en la MT se llega al "final de un cálculo" cuando se alcanza un estado especial llamado **halt** en el control finito, como resultado de una transición.

FUNCIONAMIENTO DE LA MÁQUINA DE TURING



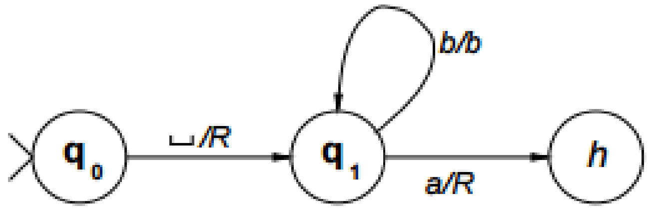
.2: MT que acepta palabras que empiezan con a

Se lee la palabra a partir del espacio en blanco.

Luego de leer el blanco, la cabeza se mueve a la derecha (R) y queda en el segundo carácter.

Funcionamiento de la Máquina de Turing

- Si el carácter leído es una letra **a**, avanza hacia la derecha y pasa al estado de **halt**. Finalizando el proceso de lectura, ya que corroboró que la palabra empieza con **a**. No sigue mirando que hay luego.



.2: MT que acepta palabras que empiezan con a

- Si el carácter leído es una letra **b**, escribe en la cinta una **b**, no avanza ni retrocede. No sigue procesando, por lo cual **NO valida la palabra**

Funcionamiento de la Máquina de Turing

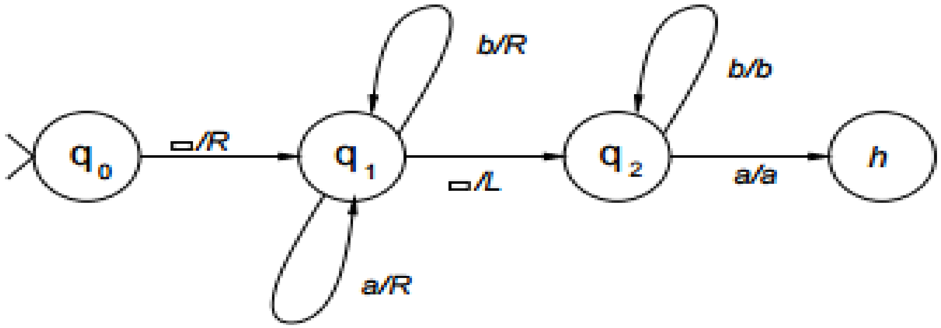
- Representaremos al **halt** por "h".
- Al llegar al **halt**, se detiene la operación de la MT, y se acepta la palabra de entrada.
- Así, en la MT no hay estados finales. En cierto sentido el halt sería entonces el 'único estado final', sólo que además detiene la ejecución.

Funcionamiento de la Máquina de Turing

- Al diseñar una MT que acepte un cierto lenguaje, en realidad diseñamos el autómata finito que controla la cabeza y la cinta, el cual es un ***autómata con salida*** .
- Funcionamiento:
  - Cuando trazamos una flecha que va de un estado **p** a un estado **q** con etiqueta **σ/L**, la cabeza lectora hace un movimiento a la izquierda, indicada por el carácter **L (left).** $\mathbf{p} \xrightarrow{\sigma/L} \mathbf{q}$
  - Similarmente cuando se tiene una flecha con **σ/R** el movimiento es a la derecha. $\mathbf{p} \xrightarrow{\sigma/R} \mathbf{q}$
  - Cuando la flecha tiene la etiqueta **σ/β**, donde **β** es un carácter, entonces la acción es escribir el carácter **β** en la cinta. $\mathbf{p} \xrightarrow{\sigma/\beta} \mathbf{q}$

Funcionamiento de la Máquina de Turing

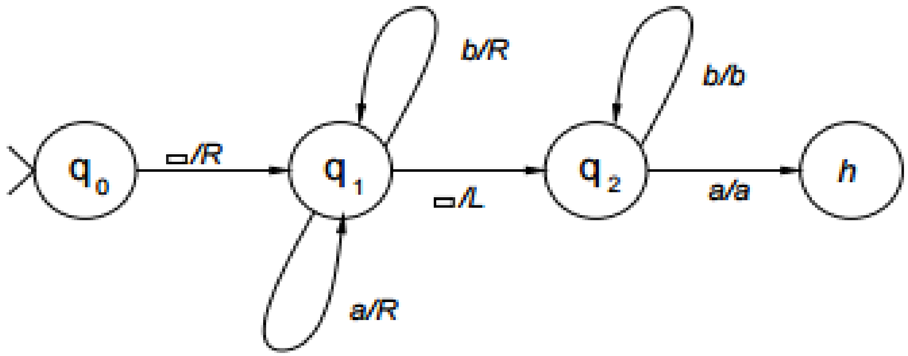
- Diseñar una MT que acepte las palabras en {a, b} que terminen con **a**.
- Aunque este ejemplo parece bastante similar al precedente, en realidad es más complicado, pues para ver cuál es la última letra, hay que ir hasta el blanco a la derecha de la palabra, luego regresar a la última letra y verificar si es una "**a**".



MT que acepta palabras que terminan con a

FUNCIONAMIENTO DE LA MÁQUINA DE TURING

- Se lee el primer carácter blanco y se mueve la cabeza a la derecha (R)
- Se leen los siguientes caracteres: si es una **b** o una **a**, mueve la cabeza a derecha.
- Cuando llega a un espacio en blanco, se mueve a la izquierda, retrocede al carácter previo. Si es una letra **a**, pasa al estado de **halt** y valida la palabra. Si es una **b**, no avanza y no valida.



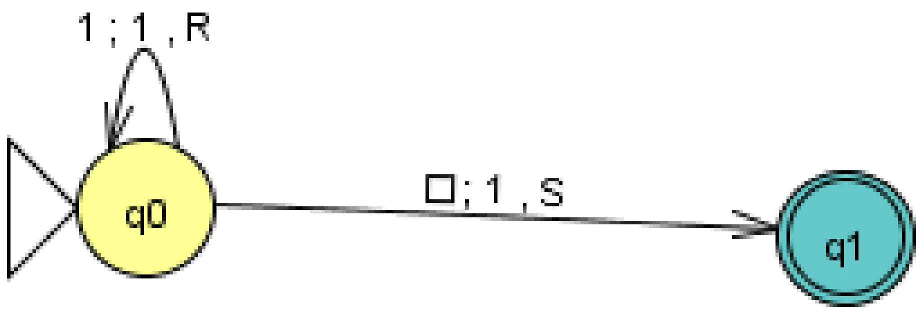
MT que acepta palabras que terminan con a

Funcionamiento de la Máquina de Turing

- Las máquinas de Turing se pueden usar como traductoras o generadora: se ingresa una cadena de entrada y se obtiene una nueva cadena de salida.

Ej. Diseñar una MT que genere el sucesor unario. El alfabeto es Σ={1}

Si se ingresa un 1, debe retornar 11; si se ingresa 11, debe retornar 111 ….



La notación ahora indica: carácter leído; carácter escrito en cinta, movimiento.

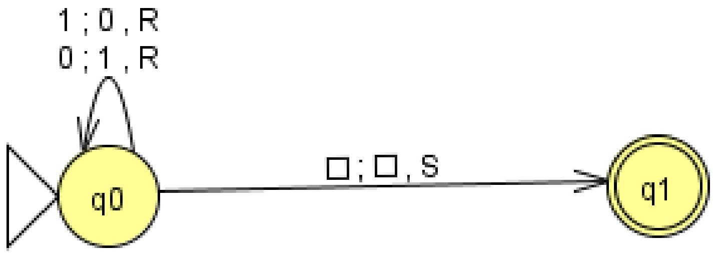
Si se ingresa un 1, la MT escribe un 1 sobre la cinta y se mueve a derecha. Al encontrar el primer blanco final, escribe un 1 y termina.

Para probar en Jflap no se tiene en cuenta el blanco inicial. La MT empieza a leer desde el primer carácter de la palabra.

FUNCIONAMIENTO DE LA MÁQUINA DE TURING

Ej. Diseñar una MT que dado un número binario retorne el inverso. El alfabeto es Σ={0,1}

Si se ingresa 0, retorna 1; si se ingresa 10 retorna 01….



La notación ahora indica: carácter leído; carácter escrito en cinta, movimiento.

Si se ingresa un 0, la MT escribe en la cinta un 1 y se mueve a derecha; si se ingresa un 0, escribe un 1 y avanza. Al finalizar la palabra y leer el primer blanco , escribe un blanco y termina.

Funcionamiento de la Máquina de Turing

Palabra aceptada por un MT:

- El único criterio para que la palabra de entrada **w** se acepte es que se llegue a **halt** en algún momento, independientemente del contenido final de la cinta, el cual es visto como "basura".

- Decimos de que un lenguaje L es Turing-aceptable si hay alguna MT que da **halt** para toda entrada **w** ∈ **L**.

Formalización de la Máquina de Turing

- Una MT es un quíntuplo **(K, Σ, Γ, δ, s)** donde:

  K es un conjunto de estados tal que h(halt) ∈ K;

  Σ es el alfabeto de entrada, donde no se incluye el espacio en blanco;

  Γ es el alfabeto de la cinta, donde esta incluido el espacio en blanco y todos los caracteres de Σ

  s ∈ K , siendo s el estado inicial;

  δ es la función de transición, que parte de un estado anterior y a través de la lectura de un carácter de la cinta, pasa a un estado nuevo, pudiendo escribir en la cinta, moverse a derecha o a izquierda.