

---

TAREA 3 CONSTRUCCIÓN DE AUTÓMATAS DE PILA

PRESENTADO POR: WILSON JAVIER GARCIA

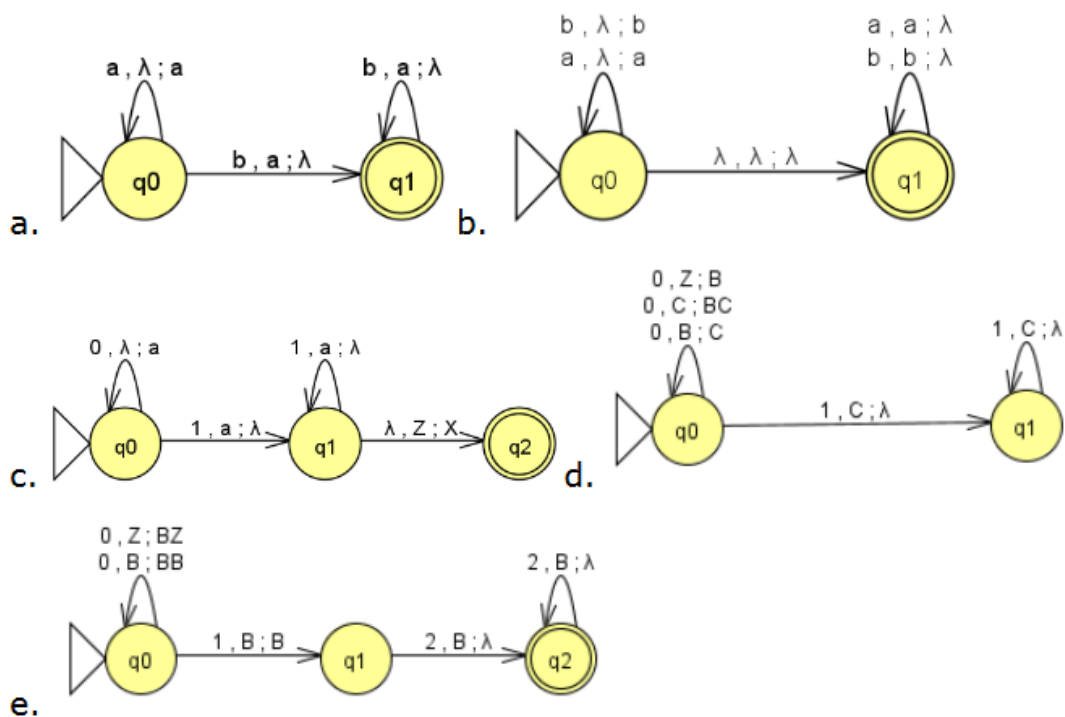
GRUPO: 1076220319

PRESENTADO A: VERMEN RAINER AYALA

GRUPO: 301405\_34

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA  
VICERRECTORÍA ACADÉMICA Y DE INVESTIGACIÓN  
CURSO: AUTÓMATAS Y LENGUAJES FORMALES  
OCTUBRE 2020

## Ejercicios 1: Autómata de Pila

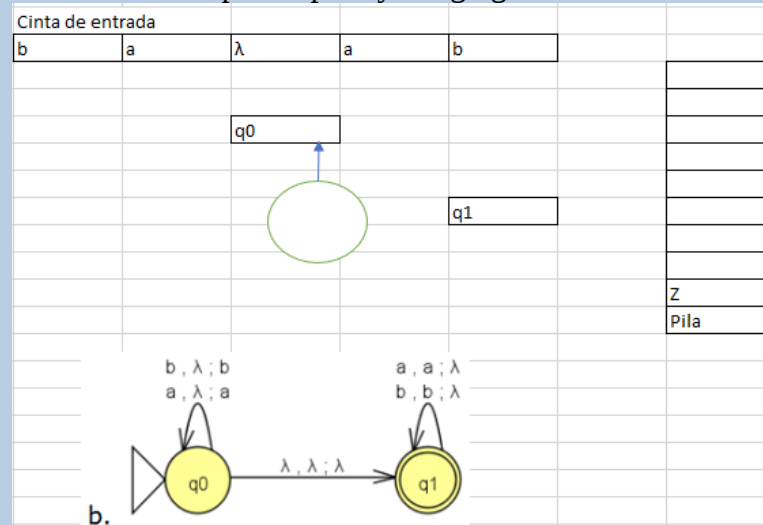


<b>EJERCICIO A TRABAJAR</b>	<p>b.</p>
<b>Caracterización del autómata a pila</b>	<p>En este espacio se realiza:</p> <ul style="list-style-type: none"> <li>-Mediante la definición formal explicar las características del autómata, identificación de la séptupla:</li> </ul> <p><math>AP = (\Sigma, r, Q, A0, q0, f, F)</math></p> <p><math>\Sigma = \{a, b, \lambda\}</math></p> <p><math>R = \{\lambda, a, b\}</math></p>

	<p><math>Q = \{q_0, q_1\}</math></p> <p><math>A_0 \in r = \lambda</math></p> <p><math>Q_0 \in Q = \{q_0\}</math></p> <p><math>F \subseteq Q = \{q_1\}</math></p> <p><math>f =</math></p> <p><math>\sigma = (q_0, b, \lambda), (q_0, b)</math></p> <p><math>\sigma = (q_0, a, \lambda), (q_0, a)</math></p> <p><math>\sigma = (q_0, \lambda, \lambda), (q_1, \lambda)</math></p> <p><math>\sigma = (q_1, a, a), (q_1, \lambda)</math></p> <p><math>\sigma = (q_1, b, b), (q_1, \lambda)</math></p>																		
	<p>- Realizar la tabla de transición:</p> <table><tr><th>Estado</th><th>a, λ</th><th>b, λ</th><th>λ, λ</th><th>a, a</th><th>b, b</th></tr><tr><td>q0</td><td>q0</td><td>q0</td><td>q1</td><td>-</td><td>-</td></tr><tr><td>q1</td><td>-</td><td>-</td><td>-</td><td>q1</td><td>q1</td></tr></table>	Estado	a, λ	b, λ	λ, λ	a, a	b, b	q0	q0	q0	q1	-	-	q1	-	-	-	q1	q1
Estado	a, λ	b, λ	λ, λ	a, a	b, b														
q0	q0	q0	q1	-	-														
q1	-	-	-	q1	q1														
	<p>- Realizar un cuadro comparativo de la Equivalencia entre AP por vaciado de pila y AP por estado final</p> <table><tr><th>Autómata por estado final</th><th>Autómata por pila vacía</th></tr><tr><td><p>“Si existe un AP <math>P</math> que reconoce un lenguaje por estado final, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por pila vacía:</p><p><math>L(P) = N(P')</math></p><p>En general, <math>P \neq P'</math></p><p>En general, <math>L(P) \neq N(P')</math>”</p></td><td><p>“Si existe un AP <math>P</math> que reconoce un lenguaje por pila vacía, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por estado final:</p><p><math>N(P) = L(P')</math>” [ CITATION Fac07 \l 3082 ]</p></td></tr></table>	Autómata por estado final	Autómata por pila vacía	<p>“Si existe un AP <math>P</math> que reconoce un lenguaje por estado final, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por pila vacía:</p> <p><math>L(P) = N(P')</math></p> <p>En general, <math>P \neq P'</math></p> <p>En general, <math>L(P) \neq N(P')</math>”</p>	<p>“Si existe un AP <math>P</math> que reconoce un lenguaje por pila vacía, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por estado final:</p> <p><math>N(P) = L(P')</math>” [ CITATION Fac07 \l 3082 ]</p>														
Autómata por estado final	Autómata por pila vacía																		
<p>“Si existe un AP <math>P</math> que reconoce un lenguaje por estado final, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por pila vacía:</p> <p><math>L(P) = N(P')</math></p> <p>En general, <math>P \neq P'</math></p> <p>En general, <math>L(P) \neq N(P')</math>”</p>	<p>“Si existe un AP <math>P</math> que reconoce un lenguaje por pila vacía, entonces existe otro AP <math>P'</math> que reconoce el mismo lenguaje por estado final:</p> <p><math>N(P) = L(P')</math>” [ CITATION Fac07 \l 3082 ]</p>																		
	<p>[ CITATION Fac07 \l 3082 ]</p>																		
Procedimiento de	Realice de manera detallada y grafica el procedimiento paso a																		

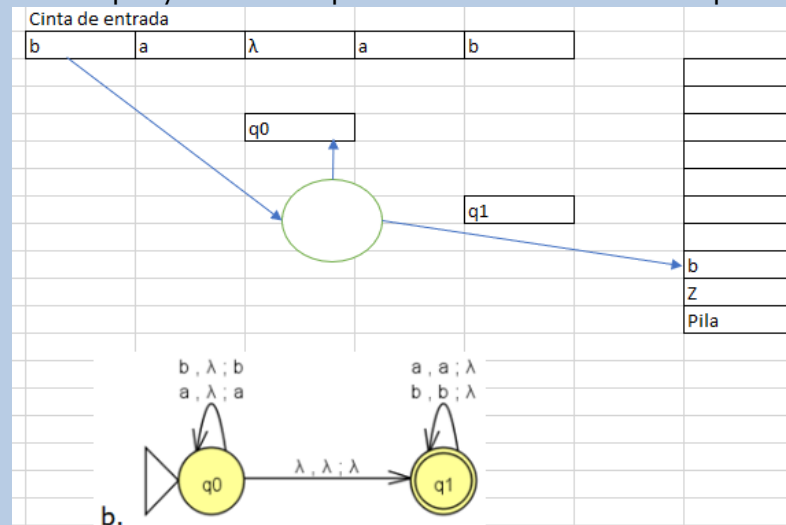
paso del recorrido de una cadena (La cadena la selecciona el estudiante, debe contener como mínimo 5caracteres) en el autómat a pila. Describir cómo funciona el almacenamiento en la pila, como funciona LIFO, etc.

Se realiza el diagrama con el cual se va a explicar el funcionamiento paso a paso y se agrega la cade en la cinta balab



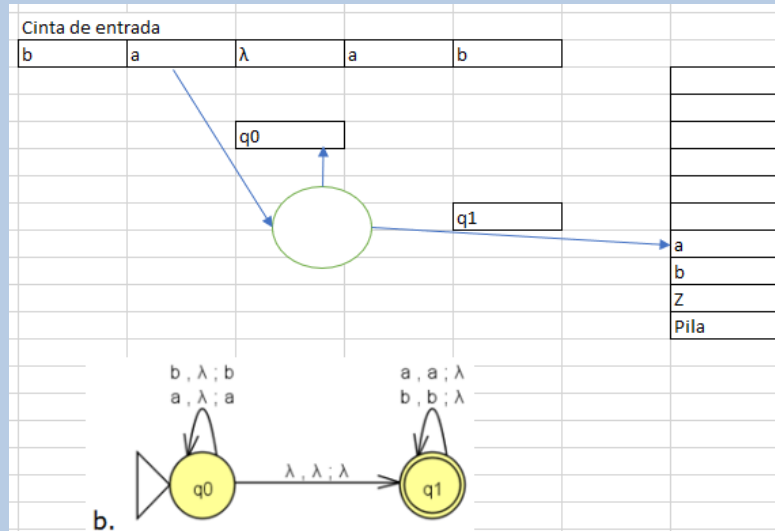
## Paso 1

De la cinta de entrada se recibe como parametro b el cual apila b pero no des apila y el automata permanece en su estado inicial q0



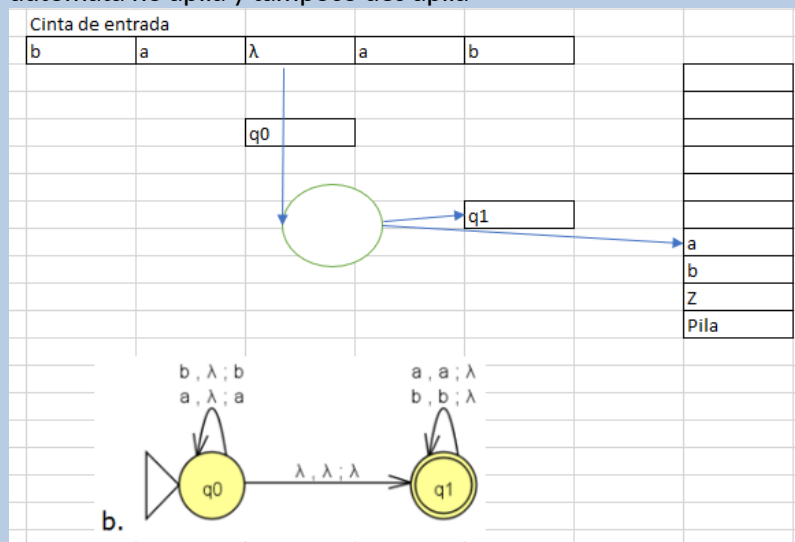
## Paso 2

En el paso dos el automata sigue en su estado inicial  $q_0$  y recibe como parametro a de la cinta de entrada y apila a pero no des apila, en este proceso el automata sigue en su estado inicial  $q_0$



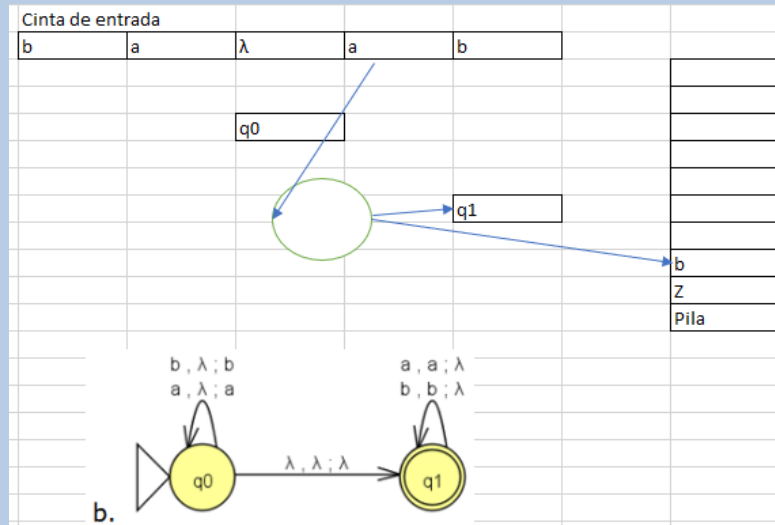
### Paso 3

En el paso tres se recibe como para metro de entrada  $\lambda$  lo cual hace que el autómata pase al siguiente estado q1, en esta ocasión el autómata no apila y tampoco des apila



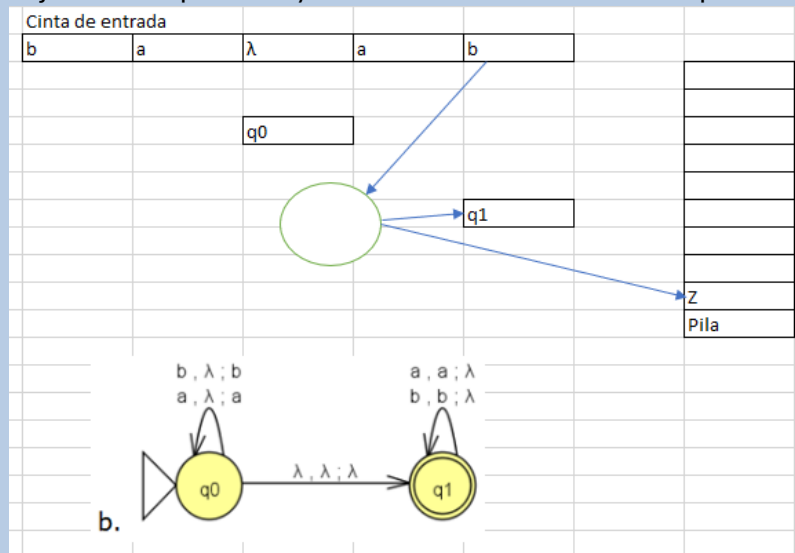
### Paso 4

En el cuarto paso, se recibe como entrada a por parte de la cinta, al estar en su estado q1 el automata des apila una a pero no apila nada, sigue en su estado de aceptacion q1



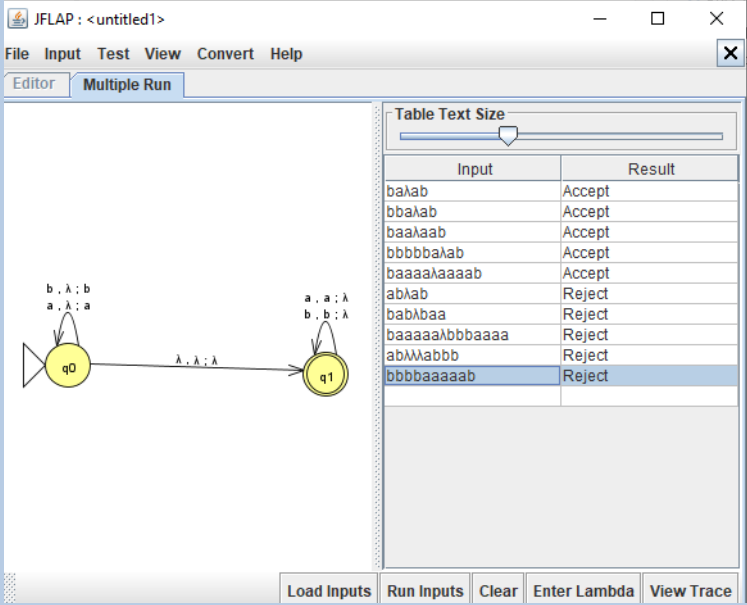
#### Paso 5

Para el paso 5, la cinta envía como parametro b, al estar en el estado de aceptación q1 el automata des apila b pero no apilada nada, dejando asi la pila vacia y el automata en su estado de aceptación q1



Practicar y verificar lo aprendido

Apoyándose en el simulador JFlap(Anexo 1-JFLAP) o VAS(Anexo 2-VAS) ejecutar y validar por lo menos cinco cadenas válidas y 5 cadenas rechazadas por el autómata. En este espacio adjunta la imagen.

	 <p>The screenshot shows the JFLAP software window titled "JFLAP : &lt;untitled1&gt;". It has a menu bar with "File", "Input", "Test", "View", "Convert", and "Help". Below the menu bar are two tabs: "Editor" and "Multiple Run". The "Multiple Run" tab is active, displaying a finite automaton with two states, <math>q_0</math> and <math>q_1</math>. <math>q_0</math> is the start state, indicated by a double circle. Transitions are labeled with pairs of strings separated by a semicolon: <math>q_0</math> has a self-loop labeled <math>b, \lambda; b</math> and <math>a, \lambda; a</math>; <math>q_1</math> has a self-loop labeled <math>a, a; \lambda</math> and <math>b, b; \lambda</math>; and there is a transition from <math>q_0</math> to <math>q_1</math> labeled <math>\lambda, \lambda; \lambda</math>. To the right of the automaton is a "Table Text Size" window with a slider. Below the automaton are buttons: "Load Inputs", "Run Inputs", "Clear", "Enter Lambda", and "View Trace".</p> <table border="1" data-bbox="909 346 1266 598"> <thead> <tr> <th>Input</th><th>Result</th></tr> </thead> <tbody> <tr><td>ba<math>\lambda</math>ab</td><td>Accept</td></tr> <tr><td>bb<math>\lambda</math>ab</td><td>Accept</td></tr> <tr><td>baa<math>\lambda</math>ab</td><td>Accept</td></tr> <tr><td>bbbbba<math>\lambda</math>ab</td><td>Accept</td></tr> <tr><td>baaaaa<math>\lambda</math>aaaab</td><td>Accept</td></tr> <tr><td>a<math>\lambda</math>ab</td><td>Reject</td></tr> <tr><td>ba<math>\lambda</math>baa</td><td>Reject</td></tr> <tr><td>baaaaa<math>\lambda</math>bbbaaaa</td><td>Reject</td></tr> <tr><td>ab<math>\lambda\lambda</math>abbb</td><td>Reject</td></tr> <tr><td>bbbbaaaaaab</td><td>Reject</td></tr> </tbody> </table>	Input	Result	ba $\lambda$ ab	Accept	bb $\lambda$ ab	Accept	baa $\lambda$ ab	Accept	bbbbba $\lambda$ ab	Accept	baaaaa $\lambda$ aaaab	Accept	a $\lambda$ ab	Reject	ba $\lambda$ baa	Reject	baaaaa $\lambda$ bbbaaaa	Reject	ab $\lambda\lambda$ abbb	Reject	bbbbaaaaaab	Reject
Input	Result																						
ba $\lambda$ ab	Accept																						
bb $\lambda$ ab	Accept																						
baa $\lambda$ ab	Accept																						
bbbbba $\lambda$ ab	Accept																						
baaaaa $\lambda$ aaaab	Accept																						
a $\lambda$ ab	Reject																						
ba $\lambda$ baa	Reject																						
baaaaa $\lambda$ bbbaaaa	Reject																						
ab $\lambda\lambda$ abbb	Reject																						
bbbbaaaaaab	Reject																						
<p>Lenguaje regular</p>	<p>Agregar el lenguaje regular del autómata</p>																						