

Facultad de Ingeniería de Sistemas Computacionales
Asignatura: Lenguajes formales, autómatas y Compiladores
Laboratorio #7
Autómatas de Pila

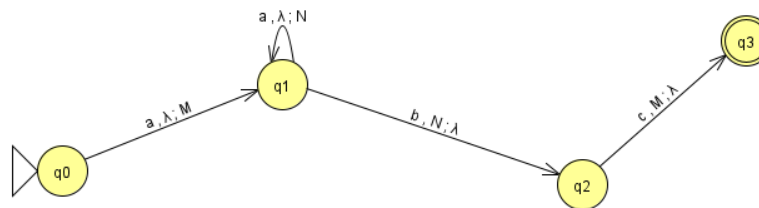
Nombre: _____

Grupo _____

Fecha: _____

Profesora: Kenia Barsallo

- I. Pasar los dos ejemplos del taller #6 del día 23-9-2024, hacer los autómatas de pila y verificar que funciona.
- II. COMO CREAR LA MATRIZ DE TRANSICIONES PARA UN AUTOMÁTA DE PILA
 1. Pasa el siguiente ejemplo y crea la matriz de transiciones que aparece abajo.

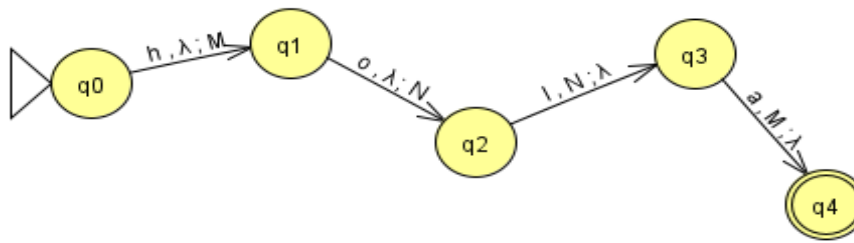


q0, a, λ ; q1, M

Primero colocar el estado de donde parte mi transición, la cadena que se lee, **lo que quito de la pila**; el estado a donde llego, lo que ingreso a la pila.

Matriz de Transiciones

Pila	Entrada	Transiciones
Z	λ	q0
MZ	aabc	q0, a, λ ; q1, M
NMZ	abc	q1, a, λ ; q1, N
MZ	bc	q1, b, N ; q2, λ
Z	c	q2, c, M ; q3, λ



Matriz de Transiciones

Pila	Entrada	Transiciones
Z	λ	q0
MZ	hola	q0, h, λ ; q1, M
NMZ	ola	q1, o, λ ; q2, N
MZ	la	q2, l, N ; q3, λ
Z	a	q3, a, M ; q4, λ

2. Crea un autómata de pila que lea la siguiente cadena: **pensamiento** y que pueda llenar la pila y que antes de terminar las transiciones de estado deje la pila vacía solo con Z. Además, crea la matriz de transiciones para este autómata de pila.

III. A partir del siguiente lenguaje genere su propio autómata de pila.

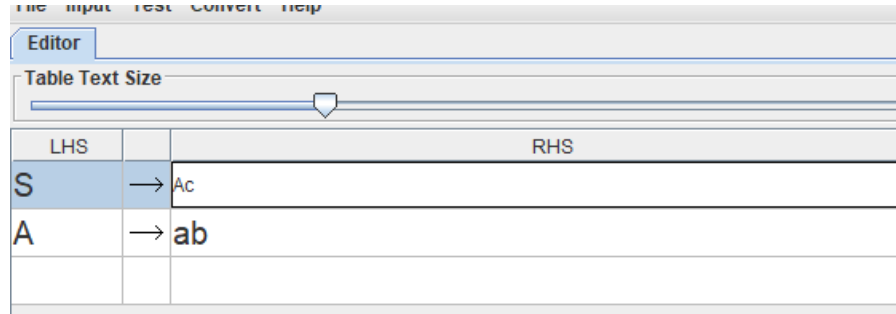
$$L = \{c^n d^{2n} f^n g^{2n} \mid n \geq 1\}$$

$$L = c d^2 f g^2$$

$$L = cddfgg$$

- IV. Genere una gramática utilizando el árbol de derivación para su verificación, y coloca la cadena de aceptación para esa gramática.

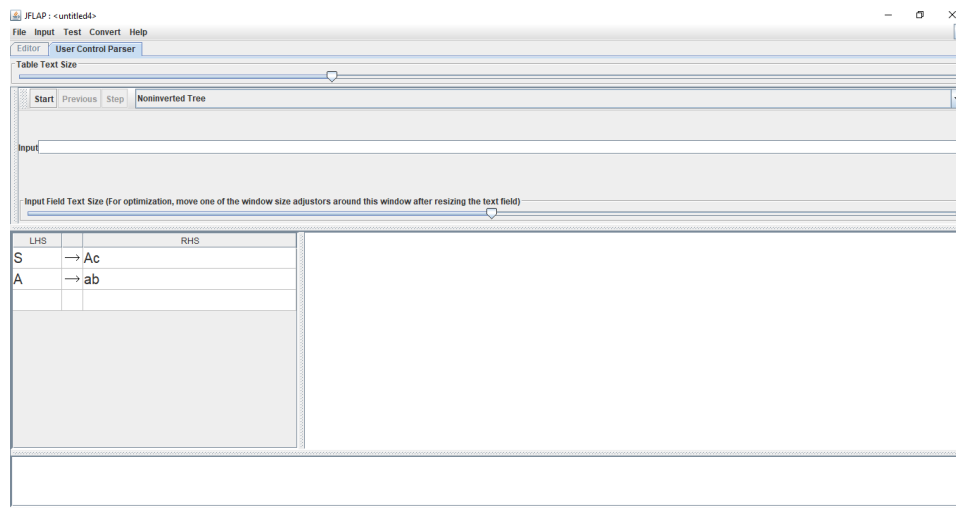
Ejemplo: cadena aceptada **abc**



The screenshot shows the JFLAP Editor window. At the top, there are tabs for 'File', 'Input', 'Test', 'Convert', and 'Help'. Below the tabs is a 'Table Text Size' slider. The main area contains a table with two columns: 'LHS' and 'RHS'. The table has two rows: the first row has 'S' in the LHS column and 'Ac' in the RHS column; the second row has 'A' in the LHS column and 'ab' in the RHS column.

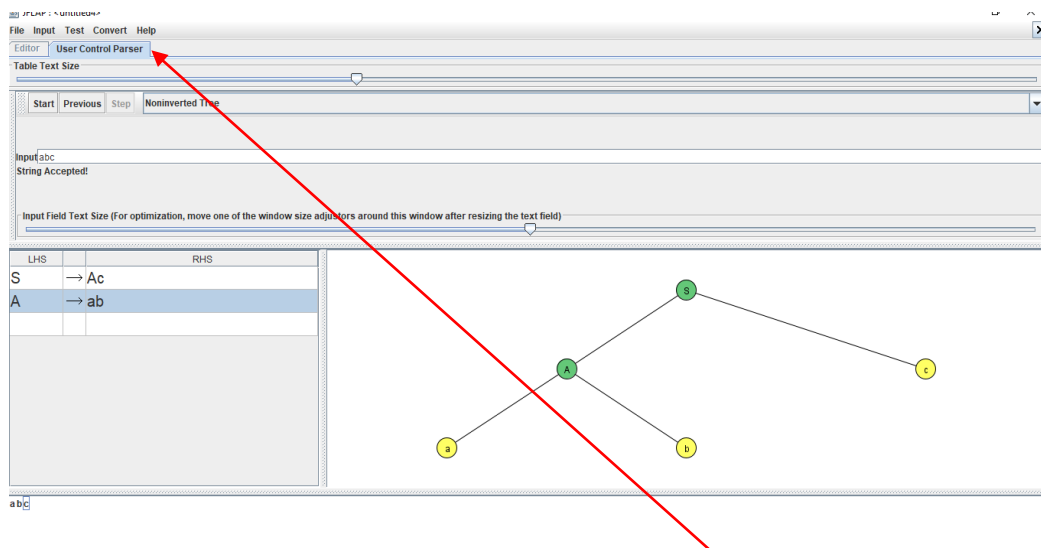
LHS	RHS
S	Ac
A	ab

Crea la gramática desde la opción GRAMMAR



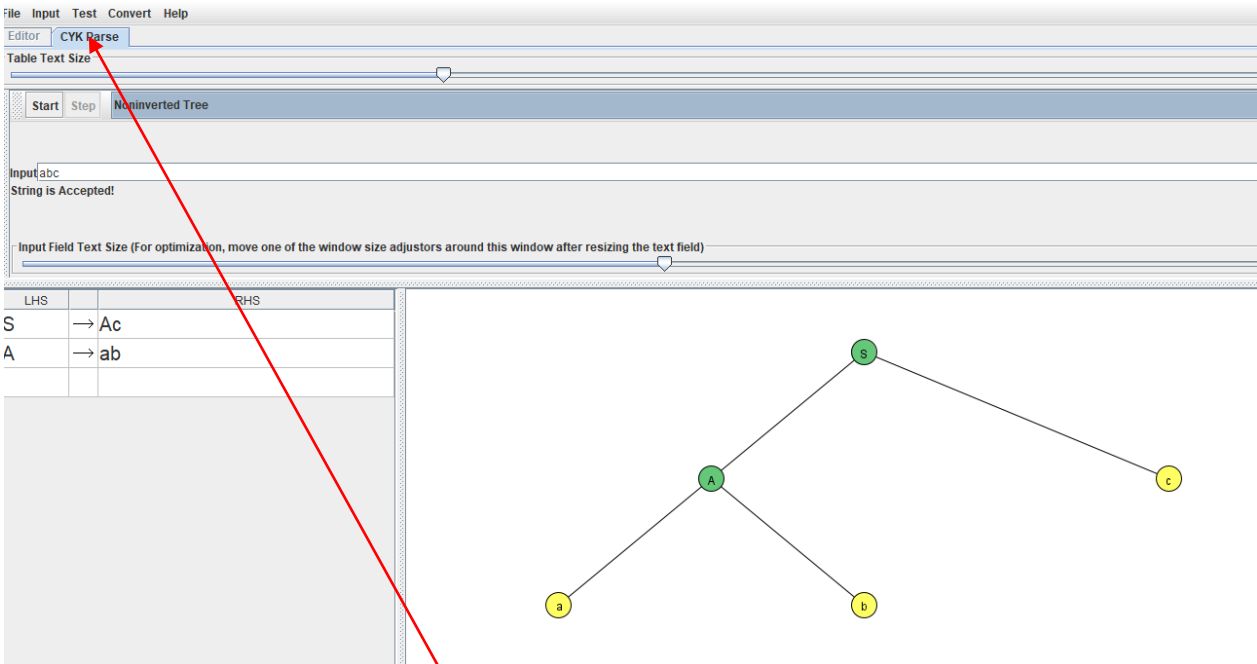
The screenshot shows the JFLAP User Control Parser window. It has a title bar 'JFLAP: <untitled4>' and a menu bar 'File Input Test Convert Help'. Below the menu bar are tabs for 'Editor', 'User Control Parser', and 'Noninverted Tree'. The 'User Control Parser' tab is active. It contains an 'Input' text field, an 'Input Field Text Size' slider, and a table with the same grammar as the previous screenshot. The 'Start' button is highlighted.

Para esta gramática recuerda reemplazar los nodos no terminales para crear una cadena valida

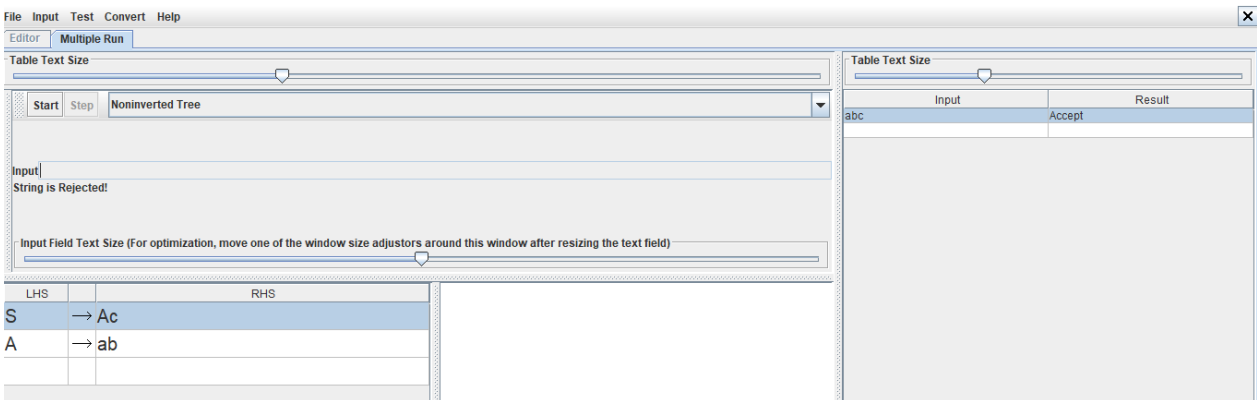


The screenshot shows the JFLAP User Control Parser window. The 'Input' text field contains 'abc' and the 'String Accepted!' message is displayed. The 'Noninverted Tree' tab is active, showing a parse tree for the string 'abc'. The tree has a root node 'S' (green) with two children: 'A' (green) and 'c' (yellow). Node 'A' has two children: 'a' (yellow) and 'b' (yellow). A red arrow points from the 'User Control Parser' tab to the 'Noninverted Tree' tab.

Clic en Input y utiliza la opción **USER CONTROL PARSER**, escribe la cadena en la línea y luego dar **STAR**, en este paso debes seleccionar cada producción para que se agregue en el árbol de derivación.



Clic en Input y utiliza la opción **CYK PARSER**, escribe la cadena en la línea y luego dar **STAR**, en este paso solo dar clic en **step** y se genera el árbol con la cadena aceptada.



Clic en Input y utiliza la opción **multiple RUN**, y hazlo como cuando verificas un autómata finito.