UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

FACULTAD DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Nombre: Apaza Calsin Michael Secarlos

CUI: 20180564

Arequipa - Perú

2021

I. OBJETIVOS

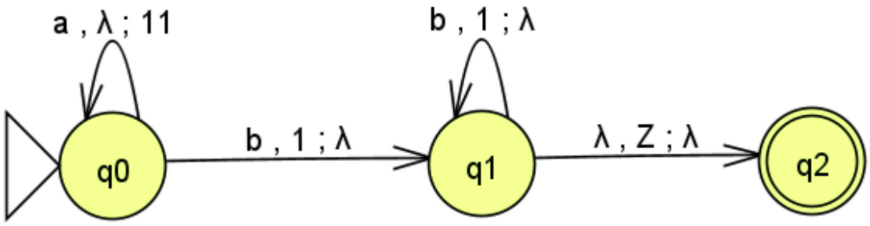
- Comprender las diferencias de un autómata de pila con respecto a un autómata finito.
- Practicar mediante ejercicios el desarrollo de un autómata de pila.

II. TEMAS A TRATAR

- Pila
- Gramáticas independientes o libres de contexto
- Autómatas a pila

III. EJERCICIOS PROPUESTOS

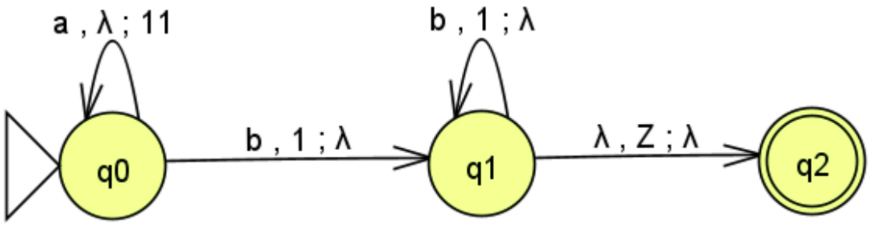
1. **Para el ejercicio 4, realizar la definición formal del autómata (2 puntos)**
   El autómata del ejercicio 4 tiene la siguiente estructura:



Entonces la definición formal de este autómata está dada por:
P4 = {Q, ∑, Γ, δ,q_0,Z_0,F}
Donde:

Q: {q_0, q_1, q_2}	//Conjunto de estados del autómata
∑: {a, b}	//Símbolos de entrada del autómata
Γ: {Z, 1}	//Alfabeto de pila
δ:	//Funciones de transición
(q_0, a, Ɛ) = (q_0, 11)	
(q_0, b, 1) = (q_1, Ɛ)	
(q_1, b, 1) = (q_1, Ɛ)	
(q_1, Ɛ, Z) = (q_2, Ɛ)	
q_0: q_0	//Estado inicial
Z_0: Z	//Símbolo inicial
F: {q2}	//Estados de aceptación del autómata

2. **Para el ejercicio 4, definir la GIC y realizar su implementación en Bison (3 puntos)**



La gramática independiente de contexto es:
S => aSbb
S => Ɛ
GIC: Esto porque en el estado q_0, la transición a ese mismo estado agrega dos veces el símbolo '1' a la pila, por ello es necesario que la cadena contenga el doble 'b' para ser aceptada.

La implementación en Bison es la siguiente:

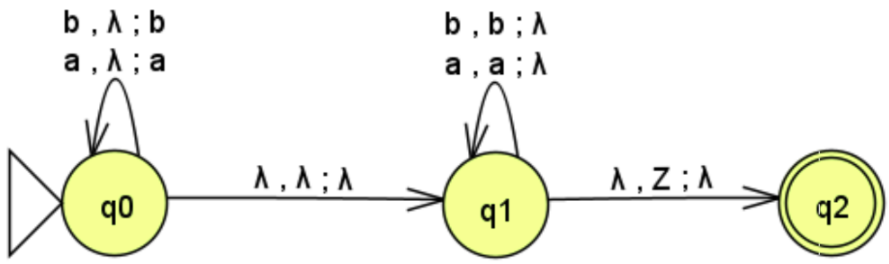
```
%%
a       {return A;}
b       {return B;}
\n      {return NL;}
.       return *yytext;
%%
```

Definimos las letras de entrada, en este caso son 'a' y 'b', mientras que en el sintactico2.y tenemos los 3 tokens A, B y NL:

```
%token A
%token B
%token NL
%%
cadena: S NL {printf("Se imprime A y doble B\n");};
S: A S B B | ;
cadena: cadena S NL {printf("Se imprime A y doble B\n");};
%%
```

Y la cadena aceptada está formada por S y NL, donde S es la repetición de A **n** veces y B **2n** veces.

3. **Para el ejercicio 5, realizar la definición formal del autómata (2 puntos)**
   El autómata del ejercicio 4 tiene la siguiente estructura:



Entonces la definición formal de este autómata está dada por:

P4 = {Q, ∑, Γ, δ,q_0,Z_0,F}

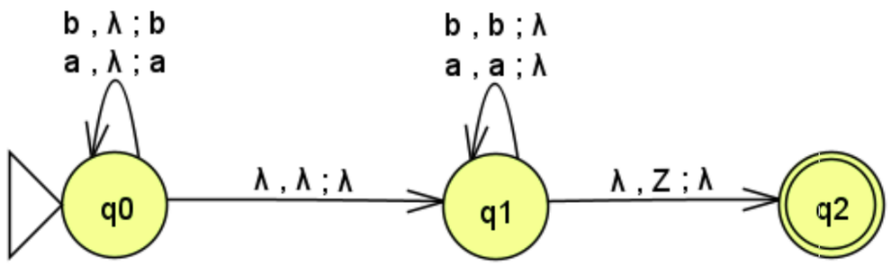
Donde:

Q: {q_0, q_1, q_2}	//Conjunto de estados del autómata
∑: {a, b}	//Símbolos de entrada del autómata
Γ: {a, b}	//Alfabeto de pila
δ:	//Funciones de transición

(q_0, a, Ɛ) = (q_0, a)
(q_0, b, Ɛ) = (q_0, b)
(q_0, Ɛ, Ɛ) = (q_1, Ɛ)
(q_1, b, b) = (q_1, Ɛ)
(q_1, a, a) = (q_1, Ɛ)
(q_1, Ɛ, Z) = (q_2, Ɛ)

q_0: q_0	//Estado inicial
Z_0: Z	//Símbolo inicial
F: {q2}	//Estados de aceptación del autómata

4. **Para el ejercicio 5, definir la GIC y realizar su implementación en Bison (3 puntos)**



La gramática independiente de contexto es:

S => abS | Sab | baS | Sba

S => Ɛ

GIC: Esto porque en el estado q_0, permite agregar tantas ‘a’ o ‘b’ como se desee, pero en q1, por lo menos se tiene que retirar una ‘a’ u una ‘b’ para que la cadena tenga aceptación.

El ejercicio en Bison tiene la siguiente forma:

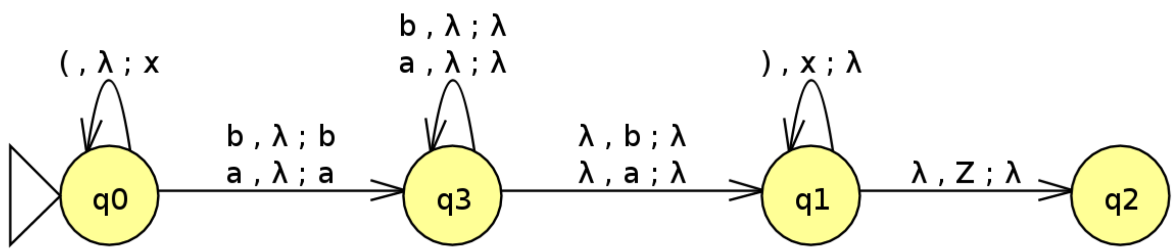
```
%token A
%token B
%token NL
%%
cadena: S NL {printf("Se imprime al menos una A y una B\n");};
S: A B S | S A B | B A S | S B A | ;
cadena: cadena S NL {printf("Se imprime al menos una A y una B\n");};
%%
```

Donde S se define por una combinación de ambas letras(para que cumpla con el minimo), para luego permitir que este incremente de forma ilimitada.

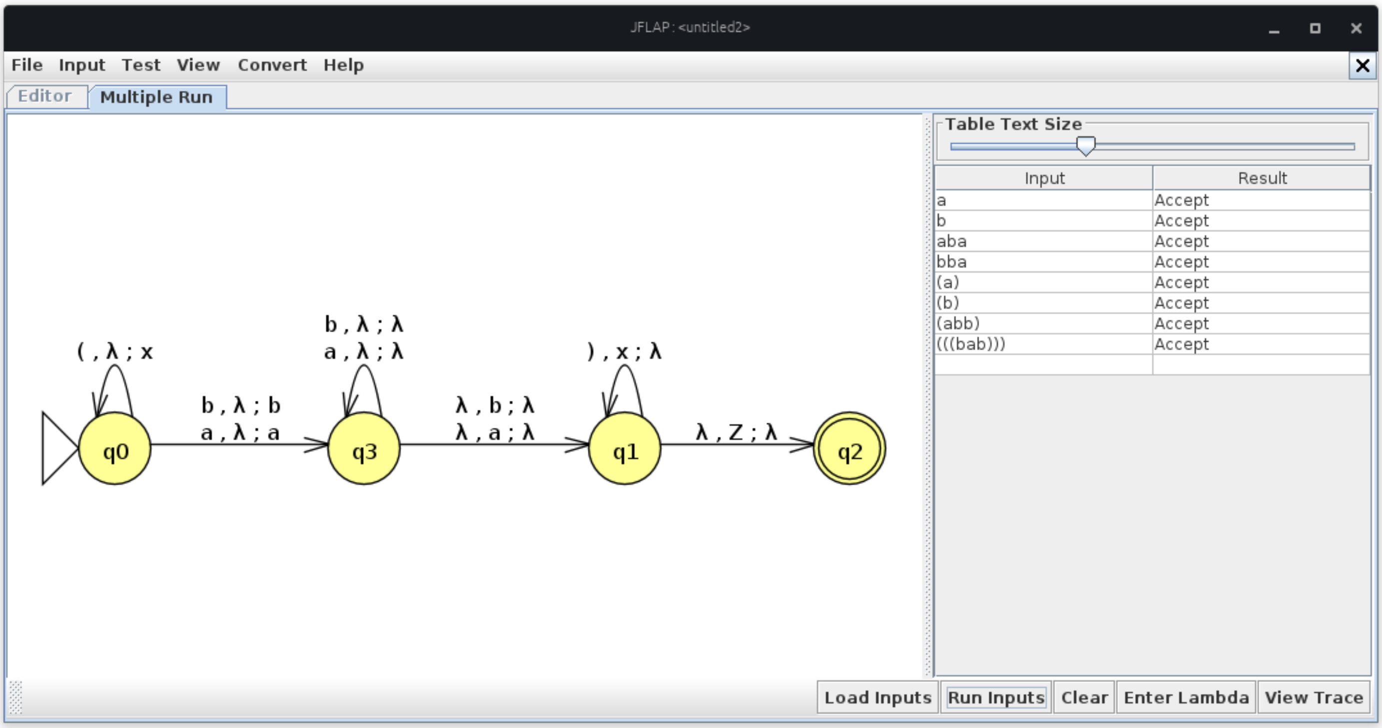
5. **Modificar el ejercicio 6 para que acepte cadenas de una o varias apariciones de a y/o b. Por ejemplo:**
    - **a**
    - **b**
    - **aba**
    - **bba**
    - **(a)**
    - **(b)**
    - **(abb)**
    - **(((bab)))**

    Diagramar el autómata y definir si GIC(4 puntos)

    Para resolver este ejercicio, se hace uso de un nuevo estado q3, al cual se traslada desde q0 con ‘a’ o con ‘b’ insertando ‘a’ o ‘b’ en la pila, luego en el estado q3 a si mismo la funcion de transicion permite el ingreso de varias ‘a’ y varias ‘b’, para luego con una función de transición al estado q1, se requiera por lo menos un elemento de la pila de ‘a’ o ‘b’.



A continuación se muestran las los resultados de las entradas a este autómata:



Y la Gramática Independiente del Contexto de este autómata sería:
S => S | (S)
S => LS
L => a|b

6. **Para la siguiente GIC (6 puntos)**
I → a | b | Ia | Ib | I0 | I1
E → I | E ∗ E | E + E | (E)
Definir:

- **La función de transición de su respectivo autómata a pila.**
  Podemos definir el mismo autómata de la siguiente manera:
  I => a | b | Ia | Ib | I0 | I1
  F => I | (E)
  T => F | T ∗ F
  E => T | E + T

  Entonces podemos definir las transiciones del autómata de pila:
  δ:
  (q3, Ɛ, Z) = (q_0, Ɛ)
  (q_0, (, Ɛ) = (q_0, x)
  (q_0, a, Ɛ) = (q4, o)
  (q_0, b, Ɛ) = (q4, o)
  (q4, 0, Ɛ) = (q4, Ɛ)
  (q4, 1, Ɛ) = (q4, Ɛ)
  (q4, a, Ɛ) = (q4, Ɛ)
  (q4, b, Ɛ) = (q4, Ɛ)
  (q4, Ɛ, Ɛ) = (q5, Ɛ)

(q4, +, Ɛ) = (q5, Ɛ)
(q4, *, Ɛ) = (q5, Ɛ)
(q5, 0, Ɛ) = (q5, Ɛ)
(q5, 1, Ɛ) = (q5, Ɛ)
(q5, a, Ɛ) = (q5, Ɛ)
(q5, b, Ɛ) = (q5, Ɛ)
(q5, a, o) = (q5, Ɛ)
(q5, b, o) = (q5, Ɛ)
(q_1,), Ɛ) = (q_1, x)
(q_1, Ɛ, Z) = (q_2, Ɛ)

- **Diagrama en JFLAP del autómata.**
  El diagrama en JFLAP es: