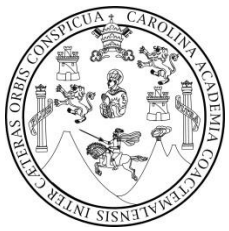
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACIÓN



Primer Semestre 2021

Sección	Catedrático	Tutor académico
A+	Ing. Otto Amilcar Rodríguez Acosta	Javier Alberto Cabrera Puente
B+	Ing. David Estuardo Morales Ajcot	Kevin Ariel Cruz Ortíz
B-	Inga. Zulma Karina Aguirre Ordoñez	Danilo Urías Coc Carlos Eduardo Hernández Molina
A-	Inga. Damaris Campos de López	Luis Manuel Morales López

Enunciado Proyecto 2

Objetivos:

- Implementar un software en Python, que permitan plasmar los conocimientos sobre lenguajes independientes del contexto adquiridos en la unidad 4 del programa del curso de lenguajes formales y de programación.
- Dar continuidad a la implementación de soluciones de software empleando paradigmas de programación.

Descripción:

Para poder profundizar más en el tema de los **lenguajes independientes del contexto,** se solicita que realice un programa en **consola** utilizando el lenguaje de programación **Python**. El programa permite ingresar gramáticas libres del contexto, y se construye un autómata de pila equivalente capaz de reconocer el lenguaje que describe cada gramática.

El flujo del programa funcionará a través de menús en pantalla con los cuales el usuario podrá navegar entre las distintas opciones con las que cuenta la aplicación. Las funcionalidades agrupan acciones de gran utilidad para quienes desean comprender el funcionamiento de los lenguajes libres del contexto.

El ingreso de las gramáticas libres del contexto, será realizado a través de archivos de entrada que permitirán optimizar la usabilidad del programa. Cabe resaltar que los archivos de entrada tendrán una estructura definida y sin errores, de manera que puedan ser leídos utilizando funcionalidades propias de Python sin necesidad de implementar algún tipo de analizador.

Durante la calificación se harán preguntas acerca de la manera en la cual se implementó cada una de las funcionalidades para poder comprobar que el estudiante sea el autor de la solución.

Características del Programa

Pantalla inicial

Cuando se inicie el programa se deberá mostrar una pantalla de bienvenida que muestre el nombre del programa y los datos personales del desarrollador (Nombre, Carnet), así también en la parte inferior deberá mostrar una cuenta regresiva de 5 segundos, cuando la cuenta finalice, se deberá mostrar la palabra "¡Bienvenido!" y luego mostrar el menú principal.

Menú principal

El menú principal es la pantalla que permite seleccionar las opciones ofrecidas por el programa. Para seleccionar una opción, el usuario debe ingresar el número que representa la opción y a continuación presionar la tecla enter.

OPCIONES DEL MENÚ PRINCIPAL

1. Cargar archivo

Esta opción permite cargar un archivo de entrada con extensión .glc que contiene la información de las gramáticas libres del contexto. El programa debe tener la capacidad de descartar las gramáticas que no sean únicamente libres del contexto, por ejemplo, si dentro del archivo encuentra una gramática regular deberá descartarla.

Dentro del archivo de entrada podrán venir N gramáticas, el final de cada gramática estará marcado por un asterisco (*).

Las partes de cada gramática estarán especificadas en el orden siguiente:

Nombre
No terminales;Terminales;No terminal inicial
Producción 1
Producción 2
Producción 3
…
…
Producción N
*

Para el caso de los no terminales y terminales vendrán separados por comas.

Ejemplo de archivo de entrada:

		entrada.glc
Nombre	1	`Grm1`
No Terminales;Terminales;Terminal inicial	2	`S,A,B,C;a,b,z;S`
Producción 1	3	`S->A`
Producción 2	4	`A->a A a`
Producción 3	5	`A->B`
Producción 4	6	`B->b B b`
Producción 5	7	`B->C`
…	8	`C->z C`
Producción n	9	`C->z`
Fin de la gramática	10	`*`
Nombre	11	`Gramatica2`
No Terminales;Terminales;Terminal inicial	12	`A,B,C,D;0,1;A`
Producción 1	13	`A->1 B`
Producción 2	14	`A->0 C`
Producción 3	15	`B->1 A`
Producción 4	16	`B->0 D`
Producción 5	17	`C->1 D`
Producción 6	18	`C->0`
…	19	`D->1 C`
...	20	`C->0 A`
Producción n	21	`D->0 B`
Fin de la gramática	22	`*`
	23	
	24	
	…	
	…	

Consideraciones:

- La figura anterior muestra el archivo de entrada como se visualizaría en un editor de texto, esto es solo con fines ilustrativos, el proyecto es en consola.
- El orden de los elementos de cada gramática siempre vendrá en el mismo orden, tomar en cuenta que los no terminales, terminales y el no terminal inicial, vendrán en la misma línea; el punto y coma denota la finalización de cada uno de ellos.
- Cada gramática puede tener N producciones por lo que se deberá dejar de reconocer producciones cuando aparezca el * que denota el final.
- La gramática “Gramatica2” del ejemplo, como puede observarse no posee ninguna producción que la convierta exclusivamente en libre del contexto, en este caso, se trata de una gramática que solamente es regular por lo que no debería cargarse al programa. Se debe reportar en un archivo de salida, por qué no se cargó.
- Los no terminales y terminales que conforman el lado derecho de las producciones están separados por espacios en blanco.

2. Mostrar información general de la gramática

Esta opción del menú deberá mostrar todos los nombres de gramáticas que se encuentran actualmente en el sistema para que el usuario pueda elegir una. Cuando el usuario elija una gramática, inmediatamente se deberá mostrar la información de la gramática en la consola. Los datos que deben mostrarse son los siguientes:

Nombre de la gramática tipo 2: **Grm1**
No terminales = **{ S,A,B,C }**
Terminales = **{ a,b,z }**
No terminal inicial = **S**
Producciones:

Las producciones deben ser mostradas con la siguiente notación:

```
No terminal -> Expresión
             | Expresión
No terminal -> Expresión
No terminal -> Expresión
```

Consideraciones:

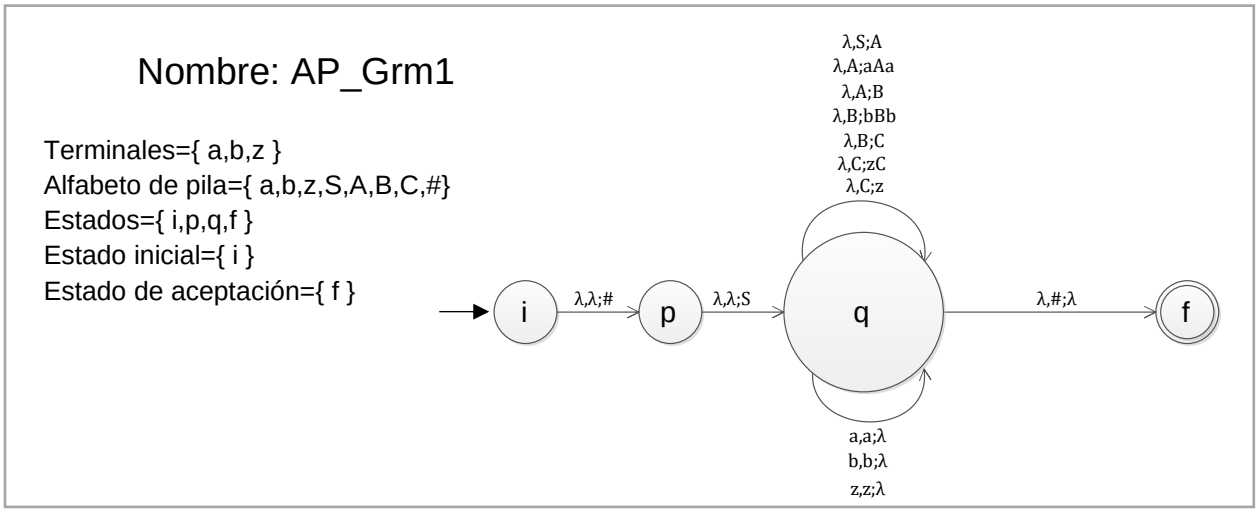
- Se debe utilizar el símbolo "|" para identificar cuando un no terminal tiene más de una posible derivación.
- La gramática solo debe mostrarse en consola, no es necesario generar un reporte externo.
- Después de mostrar la gramática se debe dejar en espera el programa y cuando el usuario lo indique, el programa debe limpiar la pantalla y mostrar el menú del módulo de gramáticas.
- "Expresión" representa las combinaciones de terminales y no terminales permitidas en las gramáticas libres de contexto.

3. Generar autómata de pila equivalente

Esta opción permite generar un autómata de pila a partir de una gramática independiente del contexto. *Para la implementación de esta opción, puede apoyarse en la teoría impartida en el curso, o consultar el capítulo 2 (Página 85) del libro "Teoría de la computación" de Glenn Brooksheer, en donde se explica a detalle el procedimiento, Teorema 2.2.*

En esta opción el usuario deberá elegir una gramática cargada en el sistema y a continuación generar un reporte en **html** que muestre el autómata de pila equivalente a través de un grafo. **Es importante que cuando se seleccione esta opción se implemente y se guarde la lógica del autómata de pila que servirá para las siguientes opciones del menú, el autómata deberá ser guardado con el nombre de la gramática, anteponiendo AP_.**

Salida esperada para la gramática “Grm1” del archivo de entrada de ejemplo:



Consideraciones:

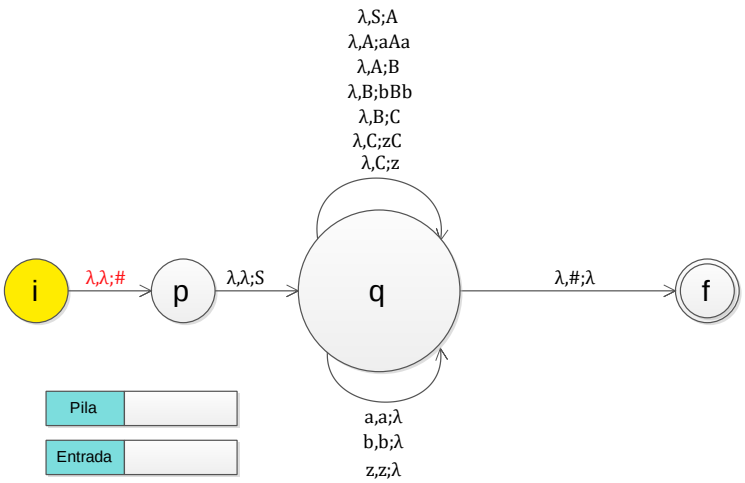
- El reporte deberá generarse en html y mostrarse automáticamente.
- Se debe identificar cada una de las transiciones del autómata de pila.
- El reporte debe incluir la información de cada uno de los componentes del autómata.
- En caso de que no se pueda mostrar el carácter λ correctamente puede utilizarse $ en su defecto.
- Queda a discreción del estudiante la presentación del reporte, debe considerarse una presentación agradable al usuario, existirá una ponderación por la presentación adecuada.
- El grafo debe ser generado con graphviz.

4. Reporte de recorrido

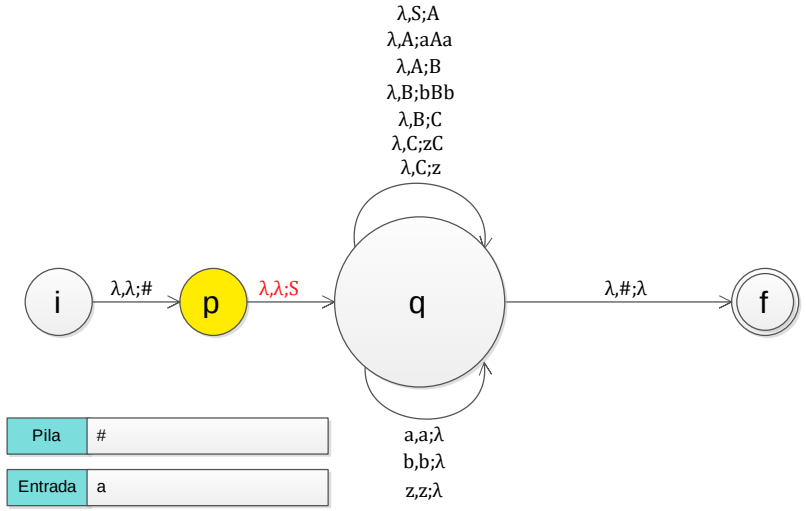
El usuario podrá elegir uno de los autómatas de pila generados en la opción 3. Finalizada la elección del autómata, se solicitará al usuario el ingreso de una cadena de entrada para que sea validada con el autómata. Esta opción permitirá generar un reporte en html que contendrá a detalle cada una de las iteraciones realizadas para validar la cadena.

Ejemplo de los pasos que contendría el archivo html al validar la cadena “abzba” con el autómata AP_Grm1

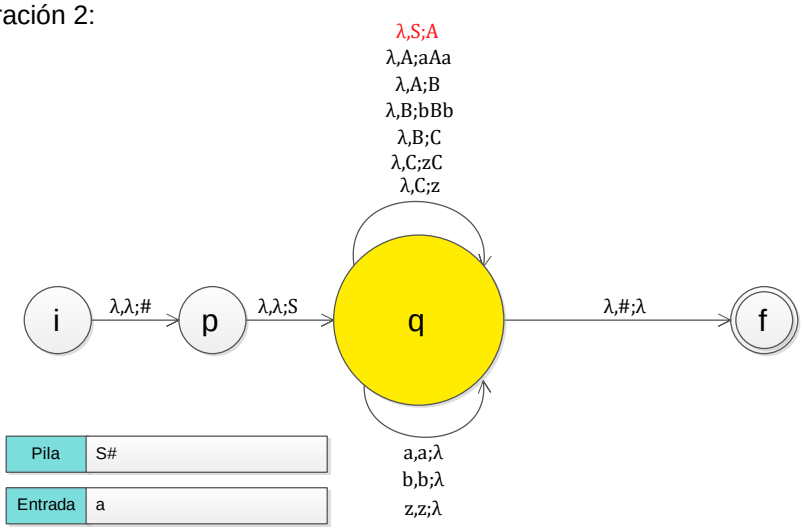
Iteración 0:



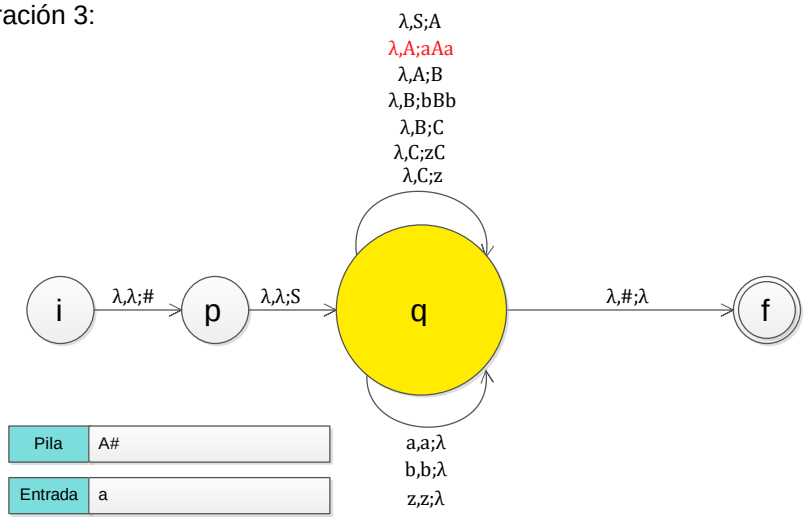
Iteración 1:



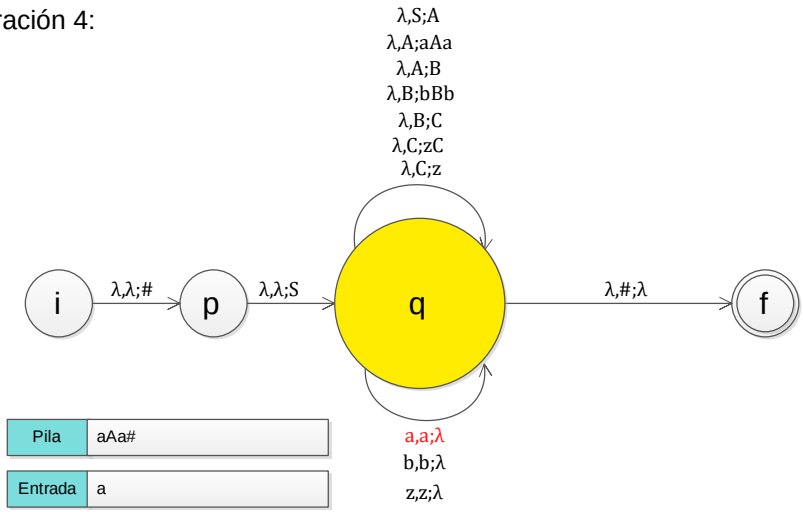
Iteración 2:



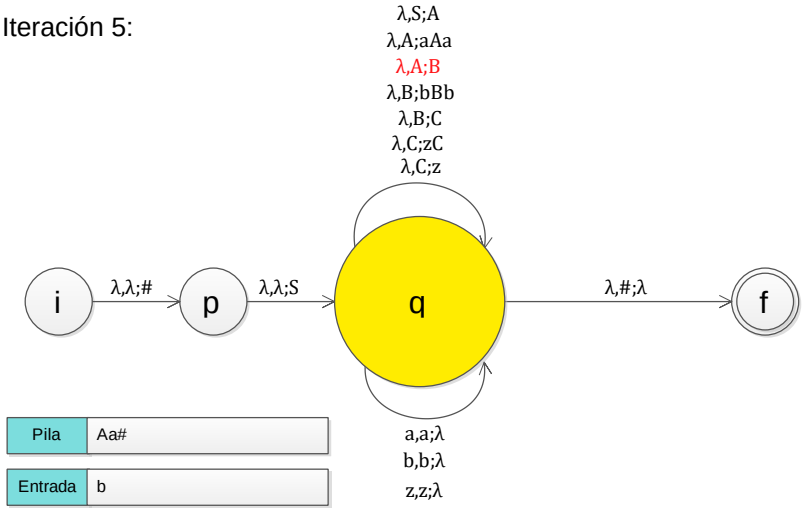
Iteración 3:



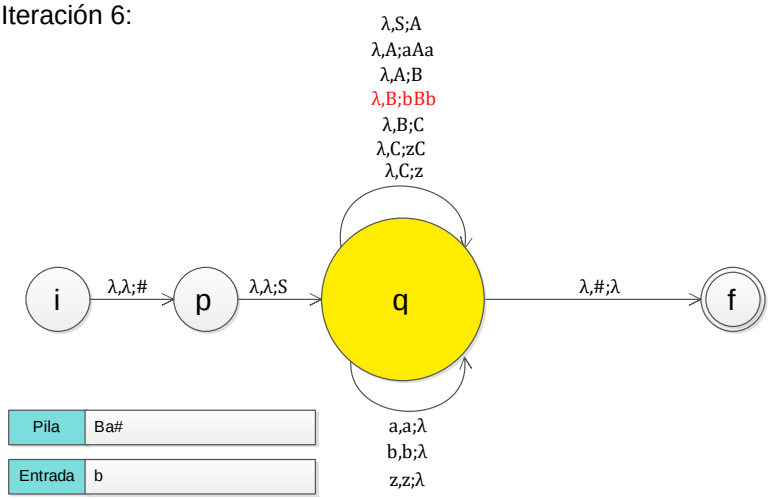
Iteración 4:



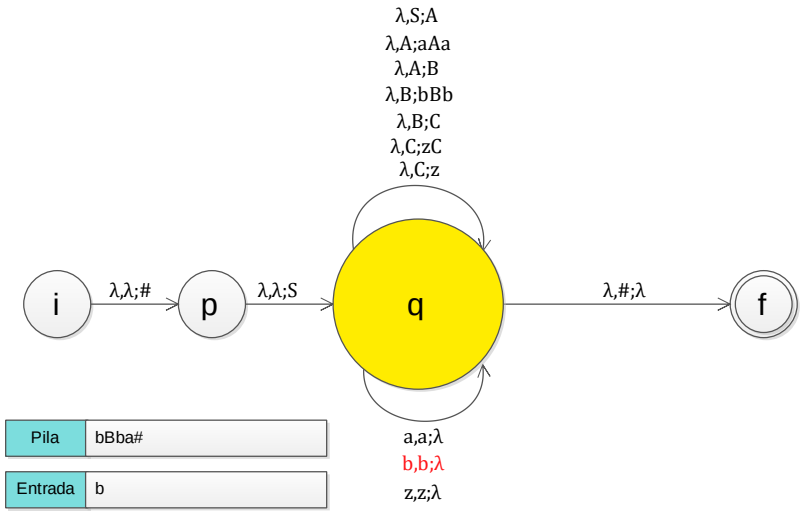
Iteración 5:



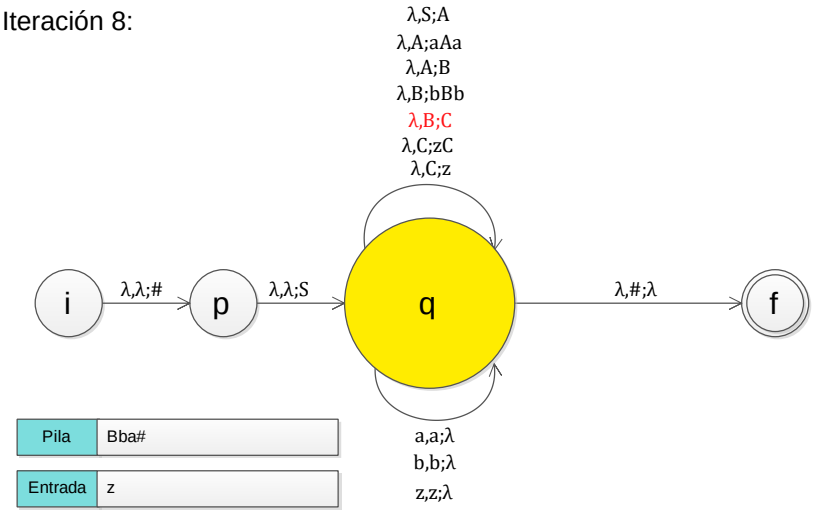
Iteración 6:



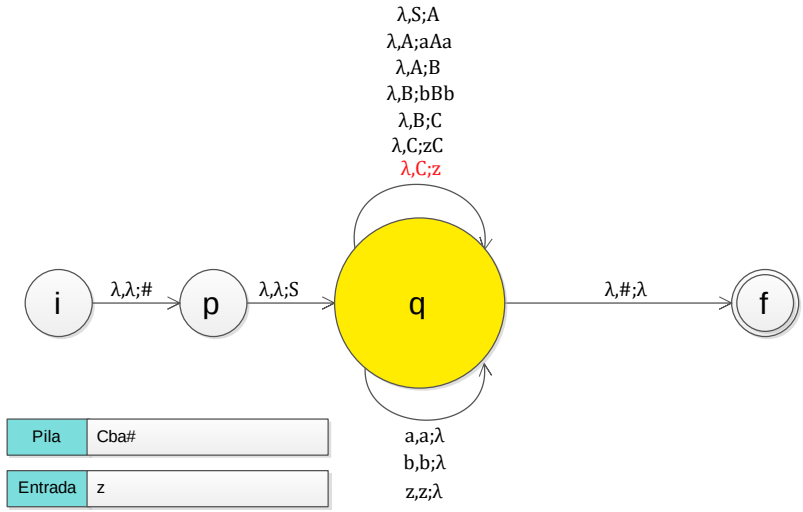
Iteración 7:



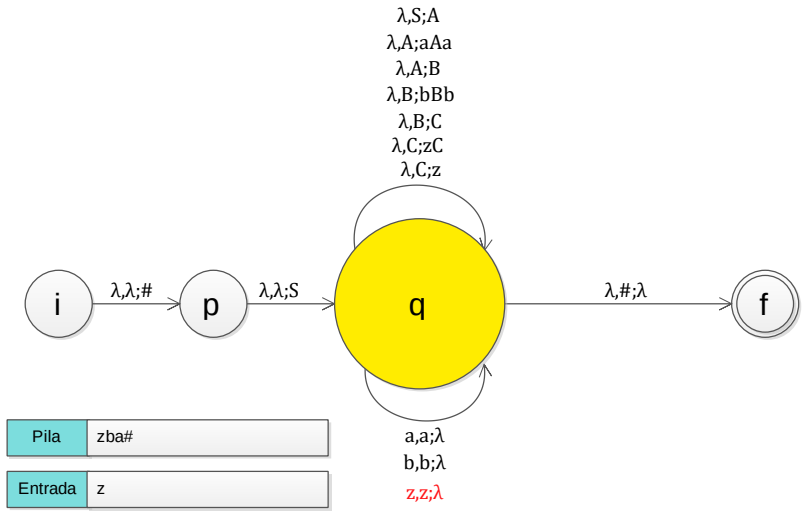
Iteración 8:



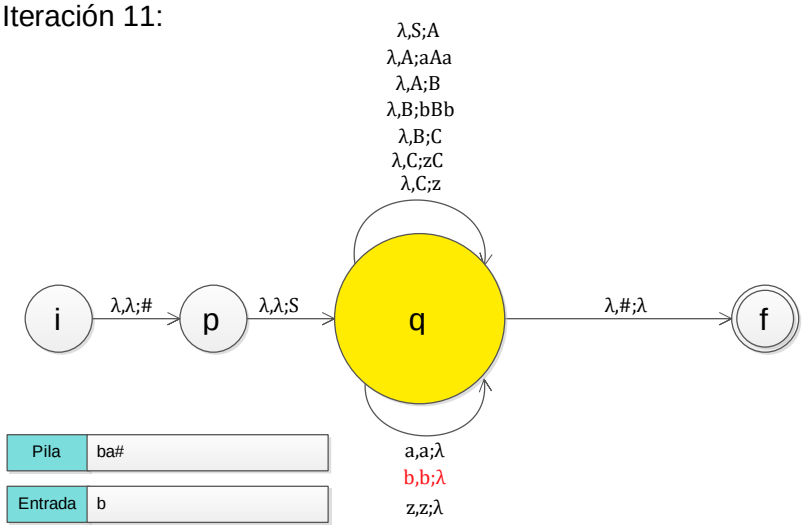
Iteración 9:



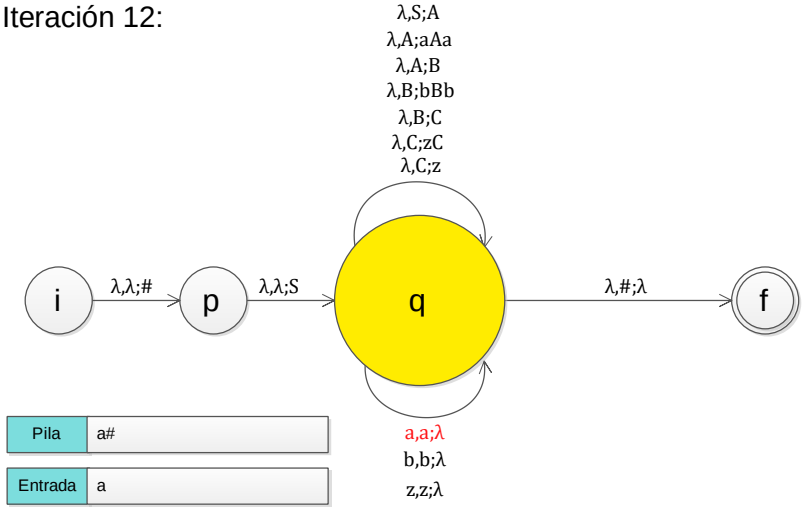
Iteración 10:



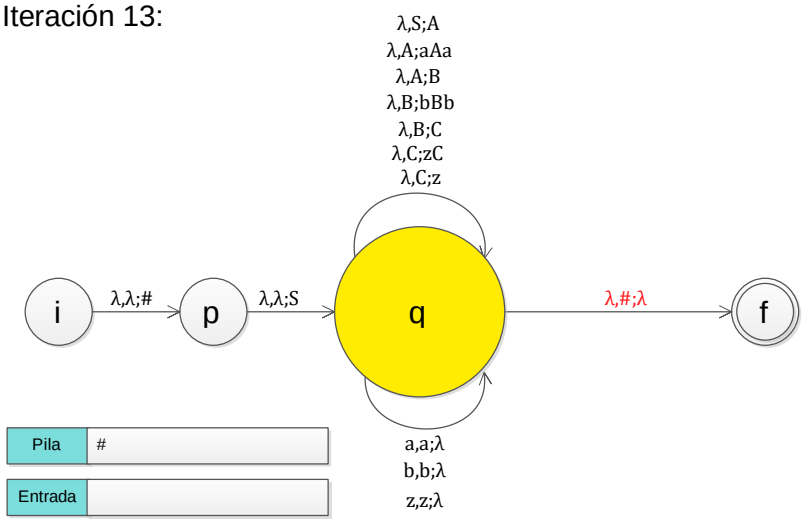
Iteración 11:



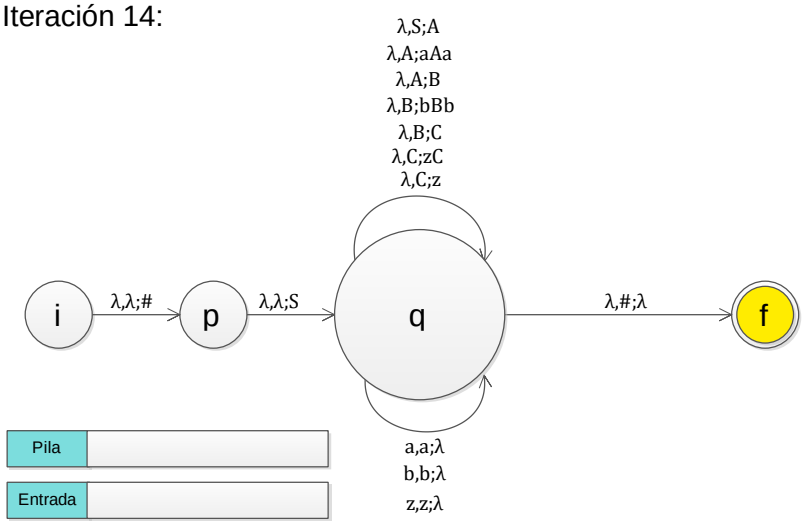
Iteración 12:



Iteración 13:



Iteración 14:



¡La cadena ingresada es válida!

Consideraciones:

- Todas las iteraciones deben ser colocadas en el reporte html identificando en todo momento la pila, entrada, estado activo y transición actual.
- Queda a discreción del estudiante la manera en que mostrará la pila y las transiciones, la única condición es que debe mostrarse de forma clara lo que está ocurriendo en ese momento con el autómata y sus componentes.
- En caso de que no se pueda mostrar el carácter λ correctamente puede utilizarse $ en su defecto.
- El reporte debe ser agradable a la vista del usuario. Se ponderará la presentación.
- El reporte debe abrirse automáticamente.
- Los grafos deben ser generados con graphviz.
- Si la entrada no es válida se deberán mostrar las iteraciones realizadas y la razón por la cual no fue válida. Con la conclusión de "Cadena NO válida."

5. Reporte en tabla

El usuario podrá elegir uno de los autómatas de pila generados en la opción 3. Finalizada la elección del autómata, se solicitará al usuario el ingreso de una cadena de entrada para validar si se acepta no en el autómata. Esta opción deberá mostrar como resultado un reporte en html con una tabla de resumen.

Ejemplo de salida esperada utilizando el autómata AP_Grm1 y la entrada abzba:

Iteración	Pila ←	Entrada	Transiciones
0		a	(i, λ, λ;p,#)
1	#	a	(p, λ, λ;q,S)
2	S#	a	(q, λ, S;q,A)
3	A#	a	(q, λ,A;q,aAa)
4	aAa#	a	(q,a,a;q, λ)
5	Aa#	b	(q,λ,A;q,B)
6	Ba#	b	(q,λ,B;q,bBb)
7	bBba#	b	(q, b,b;q, λ)
8	Bba#	z	(q, λ,B;q, C)
9	Cba#	z	(q, λ,C;q, z)
10	zba#	z	(q, z,z;q, λ)
11	ba#	b	(q, b,b;q, λ)
12	a#	a	(q, a,a;q, λ)
13	#	λ	(q, λ,#;f, λ)
14	λ	λ	f

Cadena Aceptada

Consideraciones:

- El reporte debe generarse en html y abrirse automáticamente.
- Los datos de la tabla deben mostrarse de forma legible.
- Queda a discreción del estudiante la presentación del reporte, la única condición es que debe ser agradable al usuario. Se ponderará la presentación.

Entregables:

- Manual de Usuario
- Manual Técnico, debe incluir explicación del AFD y la gramática tipo 2 utilizada en el proyecto.
- Código Fuente

Notas importantes:

- El proyecto se debe desarrollar de forma individual.
- Este proyecto se deberá desarrollar utilizando Python y Graphviz.
- **La entrega se realizará en la plataforma UEDI**. Todos los archivos solicitados deberán ser entregados en un archivo zip identificado de la siguiente forma: **[LFP]Proyecto2_carnet.zip,** el estudiante es responsable de verificar que dentro del archivo zip se encuentren todos los archivos necesarios para su calificación.
- No será permitida la modificación de archivos de entrada durante la calificación.
- La calificación deberá ser en línea y se estará grabando la reunión para tener un respaldo de la forma en que se procedió.
- La calificación de la práctica será personal y durará como máximo 30 minutos, en un horario que posteriormente será establecido.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación para ceder su lugar a otro estudiante, en caso contrario el estudiante solo obtendrá el **80%** de su nota obtenida.
- **No se dará prórroga para la entrega del proyecto**.
- **No se aceptarán Proyectos después de la fecha y hora indicada**.
- **COPIA PARCIAL O TOTAL DEL PROYECTO TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA ESCUELA DE SISTEMAS PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 25 de abril de 2021 antes de las 23:59.