



Segundo Semestre 2020

Sección	Catedrático	Tutor académico
A+	Ing. Otto Amilcar Rodriguez Acosta	Javier Alberto Cabrera Puente
A-	Ing. David Estuardo Morales Ajcot	José Manuel Véliz, Luis Manuel Morales
B	Inga. Zulma Karina Aguirre Ordonez	Danilo Urías Coc

Enunciado Proyecto 1

Objetivos:

- Implementar un software en Python que permitan plasmar los conocimientos sobre lenguajes regulares, adquiridos en la unidad 3 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 regulares se solicita que realice un programa en **consola** utilizando el lenguaje de programación **Python**. El programa básicamente se divide en dos módulos de funcionalidades importantes, un módulo enfocado en trabajar con **Autómatas finitos deterministas**, que a lo largo de este enunciado serán mencionados como **AFD**, y otro módulo enfocado en **gramáticas regulares**.

El flujo del programa funcionará a través de menús en pantalla con los cuales el usuario podrá navegar entre las distintas funcionalidades que ofrecen los módulos. Cada uno de los dos módulos agrupan acciones que permitirán agregar, manipular y generar reportes. Más adelante se explica a detalle cada uno de los módulos y sus funcionalidades específicas.

La creación tanto de AFD, como de gramáticas regulares, podrá ser realizada manualmente, como también a través de archivos de entrada que permitirán optimizar el funcionamiento 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 la información del desarrollador, siendo estos: "Lenguajes Formales y de Programación", la sección, y su número de carné, al presionar una tecla, se deberá mostrar el menú principal.

Menú principal

El menú principal es la pantalla que permite seleccionar el módulo con el cual se desea trabajar. 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. **Modulo AFD:** Al seleccionar esta opción se deberá mostrar un menú con las opciones del módulo AFD. *(Las opciones del módulo se explican a detalle en la sección Modulo AFD).*
2. **Módulo de gramáticas regulares:** Al seleccionar esta opción se deben mostrar las opciones del módulo de gramáticas regulares. *(Las opciones del módulo se explican a detalle en la sección Modulo gramáticas regulares).*
3. **Acerca de:** Al seleccionar esta opción se debe mostrar en pantalla los datos del curso.

Descripción de los módulos

MODULO AFD

Este módulo permite la gestión de autómatas finitos deterministas AFD. Presentará las siguientes opciones a través de un menú:

1. Crear AFD

Esta opción permite registrar un nuevo AFD con cada uno de sus elementos, al seleccionar esta opción se deben solicitar al usuario cada uno de los siguientes datos:

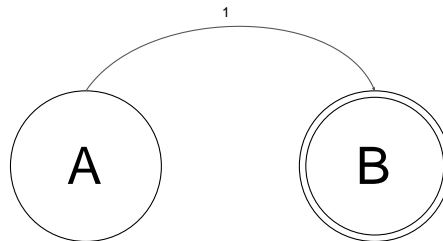
- **Nombre:** Se debe solicitar al usuario un nombre que identifique al AFD dentro de la aplicación. El nombre del AFD debe ser único, por medio de este nombre se podrá utilizar el AFD en las demás opciones del módulo.
- **Estados:** Se solicitarán los estados que conforman el AFD, estos serán ingresados de uno en uno, cuidando que no se repitan estados.

- **Ingresar Alfabeto:** Se solicitará el alfabeto del lenguaje, cuidando que no repitan símbolos y que no se utilicen símbolos que fueron declarados como estados.
- **Estado Inicial:** Se solicitará el estado inicial, este debe de existir en los estados declarados, de lo contrario se debe indicar el error y solicitarlo de nuevo.
- **Estados de aceptación:** Se solicitarán los estados de aceptación, estos deben de existir en los estados declarados de lo contrario se debe indicar el error.
- **Transiciones:** Deberá especificar cada una de las transiciones del AFD a través del siguiente formato:

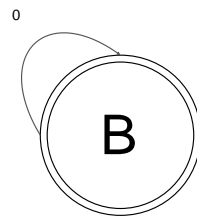
Estado origen, símbolo de entrada, Estado Destino

Ejemplos de transiciones:

A,1;B



B, 0;B



Consideraciones:

Se deberá verificar que las transiciones ingresadas correspondan a un AFD, por ejemplo, si se intenta hacer más de una transición de un estado con el mismo símbolo, se debe mostrar error.

El estudiante debe establecer una forma de detener el ingreso de transiciones, puede ser por ejemplo, detener el ingreso de transiciones al ingresar la palabra fin.

Al finalizar el ingreso de transiciones se debe guardar el AFD automáticamente y volver al menú del módulo de AFD.

2. Cargar archivo

Esta opción permitirá replicar el comportamiento de la opción 1 pero realizándolo desde un archivo de entrada que permitirá cargar 1 o más AFD's. El archivo será de texto plano con extensión .afd el contenido del archivo estará estructurado de la siguiente manera:

Ubicación	Valor	Ejemplo de archivo
Línea 0	Nombre AFD	AFD_1
Línea 1	Estados	A, B, C, D
Línea 2	Alfabeto	0, 1
Línea 3	Estado inicial	A
Línea 4	Estados de aceptación	C
Línea 5	Transición 1	A, 1; B
Línea 6	Transición 2	A, 0; C
Línea 7	Transición 3	B, 1; A
Línea 8	Transición 4	B, 0; D
Línea 9	Transición 5	C, 1; D
Línea 10	Transición 6	C, 0; A
Línea 11	Transición 7	D, 1; C
Línea 12	Transición 8	D, 0; B
Línea 13	Transición 9	A, 1; B
Línea 14	%	
Línea 15	Nombre AFD	AFD_2
Línea 16	Estados	A, x; C
...

Consideraciones:

Los archivos de entrada estarán estructurados de la forma descrita anteriormente. En ningún momento tendrán errores de sintaxis.

Las líneas de: Estados, Alfabeto y Estados de aceptación tendrán sus valores separados por comas cuando sea más de un dato.

Los datos de los AFD vendrán siempre en el mismo orden.

La línea con el símbolo % indicará el final de los datos de un AFD e inmediatamente iniciarán los datos del siguiente AFD.

Cada AFD dentro del sistema tomará el nombre que se indica al inicio de su definición.

Al finalizar la carga del archivo el sistema debe retornar al menú del módulo AFD.

3. Evaluar cadena

Al seleccionar esta opción se deberá mostrar al usuario la lista de AFD's guardados para que seleccione un AFD con el cual desea evaluar la cadena. Finalizada la elección del AFD por parte del usuario, se presentarán las siguientes opciones de evaluación de cadena.

- **Validar una cadena:** Se solicitará una cadena y se mostrará un mensaje de cadena valida o invalida según sea el caso.
- **Ruta en AFD:** Se solicitará una cadena y de ser válida esta mostrara la ruta que se siguió para poder validarla. Si no es válida, mostrará error y la ruta hasta donde logró llegar.

Al finalizar la evaluación debe retornar al menú del módulo AFD.

Queda a discreción del estudiante la forma en que mostrará la ruta tomada para evaluar la cadena, debe mostrar de manera legible y ordenada los elementos del AFD involucrados.

4. Guardar AFD en archivo

Debe crear un archivo de texto con extensión .afd que contenga toda la información de un AFD elegido por el usuario, la información deberá guardarse con la misma estructura de archivo especificada en la opción “Cargar archivo” y el nombre del archivo será el nombre con cual estaba registrado el AFD. En caso de que ya exista el archivo debe sobre escribirse.

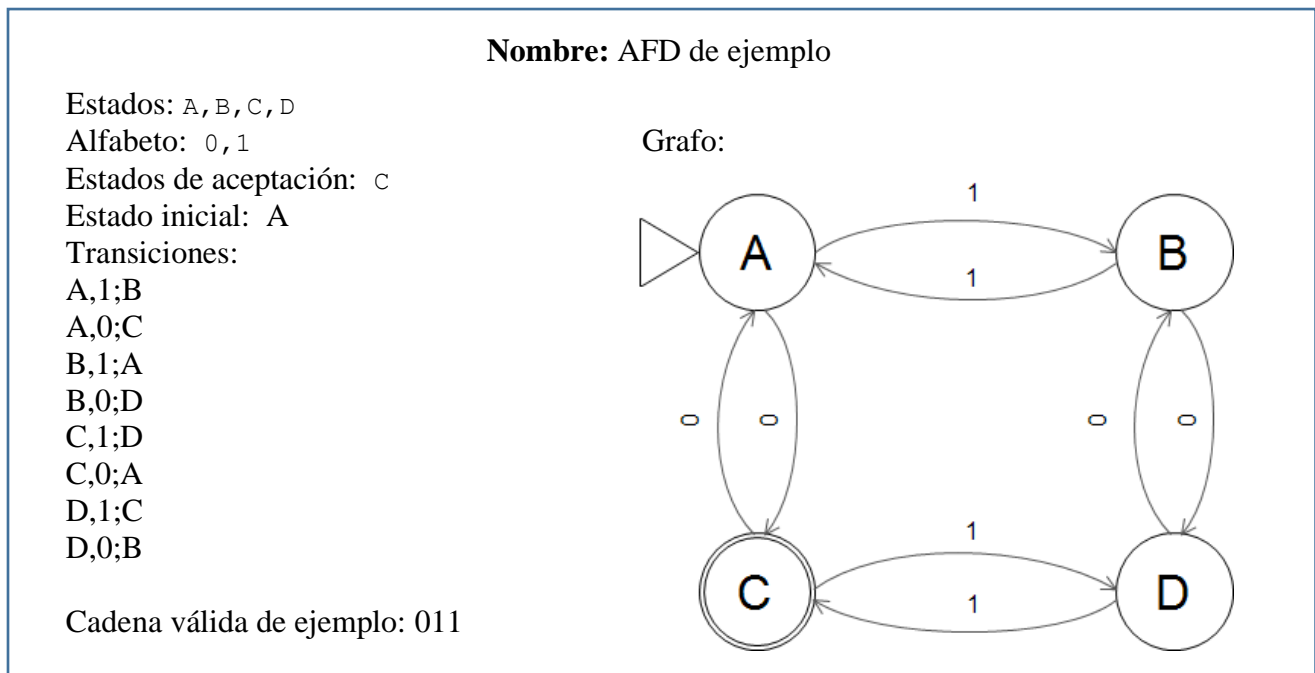
Al finalizar la escritura del archivo debe retornar al menú del módulo AFD.

5. Generar reporte AFD

Esta opción deberá mostrar en pantalla todos los AFD's que se encuentran actualmente en el sistema, el usuario deberá seleccionar uno para generar el reporte. El reporte deberá contener los siguientes datos:

- Nombre
- Estados
- Alfabeto
- Estados de aceptación
- Estado inicial
- Transiciones
- Una cadena mínima válida
- Grafo del AFD generado con graphviz

Ejemplo:



Consideraciones:

La cadena debe ser una cadena válida, en caso de que se puedan generar varias cadenas mínimas de un mismo tamaño queda a discreción del estudiante cuál presentar.

La disposición de los elementos del ejemplo de reporte es solamente una sugerencia, queda a discreción del alumno su presentación, el reporte debe ser legible y el grafo debe permitir la identificación de los distintos componentes que conforman el AFD.

Al finalizar la generación del reporte, el flujo del programa debe retornar al menú del módulo AFD.

6. Generar gramática regular

Esta opción deberá mostrar al usuario todos los AFD's que se encuentran actualmente en el sistema para que elija uno a partir del cuál generará una gramática regular representativa. La gramática deberá mostrarse con la siguiente notación: Los terminales y No terminales siguientes solamente son para ejemplificar:

Terminales = {x,y,z,...}
No Terminales = { NT1, NT2,...}
Inicio = NT1
Producciones:

```
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 debe mostrarse en consola, y dar la opción de 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 AFD.

"Expresión" representa las combinaciones de terminales y no terminales permitidas en las gramáticas regulares.

MÓDULO DE GRAMÁTICAS REGULARES

Este módulo permite la gestión de gramáticas regulares. Presentará las siguientes opciones a través de un menú:

1. Crear gramática

Esta opción permite registrar una gramática regular con cada uno de sus elementos, al seleccionar esta opción se debe solicitar al usuario cada uno de los siguientes datos:

- **Nombre:** Se debe solicitar al usuario un nombre que identifique a la gramática dentro de la aplicación. El nombre de la gramática debe ser único, por medio de este nombre se podrá utilizar la gramática en las demás opciones del módulo.
- **No terminales:** Se solicitarán los no terminales de la gramática, estos serán ingresados uno a uno cuidando que no se repitan.
- **Terminales:** Se solicitará los terminales de la gramática, que deberán ser ingresados uno a uno, será necesario revisar que no se repitan y que no se utilicen símbolos que ya fueron declarados como no terminales.
- **No terminal inicial:** Se solicitará el no terminal inicial, se deberá validar que el no terminal haya sido declarado previamente, de lo contrario debe mostrar un error y solicitarlo nuevamente
- **Producciones:** Se deberá solicitar el ingreso de las producciones una a una, la notación a utilizarse deberá ser como la que se muestra a continuación:

```
No terminal > Expresión
No terminal > Expresión
```

Consideraciones:

Para el ingreso de producciones se suprimió el uso del símbolo “|” para una mayor legibilidad, si un “no terminal” tiene más de una derivación deberá escribirse la regla completa con el símbolo “>”.

Se deberá verificar que los terminales y no terminales hayan sido registrados previamente, esto para mantener la congruencia de los datos.

“Expresión” representa las combinaciones de terminales y no terminales permitidas en las gramáticas regulares.

- 2. Cargar archivo de entrada:** Esta opción permitirá replicar el comportamiento de la opción 1 pero realizándolo desde un archivo de entrada permitiendo ingresar una o más gramáticas. El archivo será de texto plano con extensión .gre el contenido del archivo estará estructurado de la siguiente manera:

Ubicación	Valor	Ejemplo de archivo
Línea 0	Nombre Gramática	G_1
Línea 1	No Terminales	A, B, C, D
Línea 2	Terminales	0, 1
Línea 3	No terminal inicial	A
Línea 4	Regla 1	A > 0 B
Línea 5	Regla 2	A > 1 C
Línea 6	Regla 3	B > 0 A
Línea 7	Regla 4	B > 1 D
Línea 8	Regla 5	C > 0 D
Línea 9	Regla 6	C > 1 A
Línea 10	Regla 7	C > \$
Línea 11	Regla 8	D > 0 C
Línea 12	Regla 9	D > 1 B
Línea 13	%	
Línea 14	Nombre Gramática	G_2
Línea 15	No terminales	F, G, H
...

Consideraciones:

Para el ingreso de producciones desde archivo, se suprimió el uso del símbolo “|” para una mayor legibilidad. Si un “no terminal” tiene más de una derivación aparecerá la regla completa con el símbolo “>”.

Se deberá verificar que los terminales y no terminales hayan sido registrados previamente, esto para mantener la congruencia de los datos.

El símbolo \$ representa cadena vacía.

Cuando finaliza la carga del archivo, se debe retornar al menú del módulo gramáticas regulares.

Cada gramática del archivo, deberá tomar el nombre que le corresponde al inicio de la definición, cuidando sea un nombre único (que no se repita).

La línea con el símbolo % representa el final los datos de una gramática e inmediatamente iniciarán los datos de la siguiente gramática.

El orden de los componentes de cada gramática siempre tendrá el mismo orden.

3. Evaluar cadenas

Al seleccionar esta opción se deberá mostrar al usuario la lista de gramáticas guardadas para que seleccione una gramática. A continuación, se presentarán las siguientes opciones para evaluar una cadena:

- **Validar una cadena:** Se solicitará una cadena y se mostrará un mensaje de cadena válida o inválida según sea el caso.
- **Proceso de expansión:** Se solicitará una cadena y de ser válida se mostrará en pantalla el proceso de expansión que realiza la gramática para validar la cadena. En caso de que la cadena no sea válida deberá mostrar un error y la ruta hasta donde logró llegar la validación.

Al finalizar la evaluación debe retornar al menú del módulo gramáticas regulares.

Queda a discreción del estudiante la forma en que mostrará el proceso de expansión de la gramática, debe mostrar de manera legible y ordenada las producciones involucradas.

4. Eliminar recursividad por la izquierda

Al seleccionar esta opción se deberá mostrar al usuario la lista de gramáticas guardadas para que seleccione una gramática. El programa deberá evaluar si existe recursividad por la izquierda en la gramática seleccionada, de ser así mostrará en pantalla la gramática escrita sin recursividad por la izquierda, en caso contrario deberá notificar que no existe tal recursividad en la gramática.

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 gramática regular.

5. Guardar AFD en archivo

Al seleccionar esta opción se deberá mostrar al usuario la lista de gramáticas guardadas para que seleccione una gramática. Debe crear un archivo de texto con extensión .gre que contenga toda la información de la gramática elegida por el usuario, la información deberá guardarse con la misma estructura de archivo especificada en la opción “Cargar archivo” del módulo gramáticas.

Consideraciones:

El nombre del archivo será el nombre con cual estaba registrada la gramática. En caso de que ya exista el archivo debe sobre escribirse.

El símbolo \$ se utilizará para indicar cadena vacía.

6. Generar reporte gramática regular

Esta opción deberá permitir que el usuario elija una gramática regular de la lista de gramáticas cargadas en el sistema. El reporte deberá contener los siguientes datos:

- Nombre
- No terminales
- Terminales
- No Terminal inicial
- Producciones utilizando la notación explicada anteriormente
- Generar un AFD a partir de la gramática y mostrar su grafo

Ejemplo:

Nombre: Gramática de ejemplo

No terminales: A, B, C, D

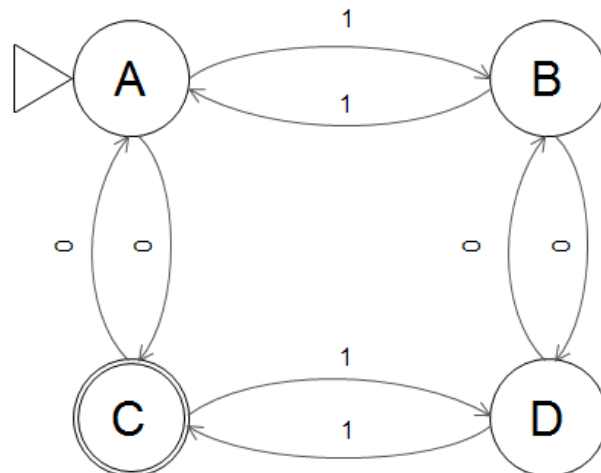
Terminales: 0, 1

Grafo:

No terminal inicial: A

Producciones:

```
A > 0 B
    | 1 C
B > 0 A
    | 1 D
C > 0 D
    | 1 A
    | $
D > 0 C
    | 1 B
```



Cadena de ejemplo: 011

Consideraciones:

La cadena debe ser una cadena válida, en caso de que se puedan generar varias cadenas mínimas de un mismo tamaño queda a discreción del estudiante cuál utilizar.

La disposición de los elementos del ejemplo de reporte es solamente una sugerencia, queda a discreción del alumno su presentación, el reporte debe ser legible y el grafo debe permitir la identificación de los distintos componentes que conforman el AFD.

Al finalizar la generación del reporte, el flujo del programa debe retornar al menú del módulo gramática regular.

Notas importantes:

- El proyecto se debe desarrollar de forma individual.
- Esta práctica 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]Proyecto1_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.
- No se dará prórroga para la entrega del proyecto.
- **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 la practica; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 24 de septiembre de 2020 antes de las 23:59.