Pablo de la Mora Vega

A01020365

Reporte

Proyecto 4

**Manual de Usuario:**

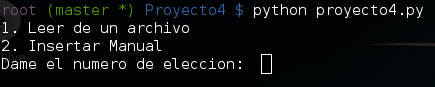
Este proyecto está hecho en Python por lo que tienes que lo tienes que tener instalado en la computadora en donde lo vas a ejecutar.

Algunos comentarios sobre el proyecto es que se recomienda correr en Linux ya que al imprimir el resultado se implementó con una función la cual necesita tres librerías para poder correr en Windows. Por lo que puede que en vez del cambio de color en la consola de Windows salgan números y letras extraños.

Dicho esto el programa lo que hace es que se le da una Gramática Libre de Contexto y lo que te regresa es la derivación por la derecha, la derivación por la izquierda y el número de resultados hasta cierto número de derivaciones y de sustituciones brindados por el usuario.

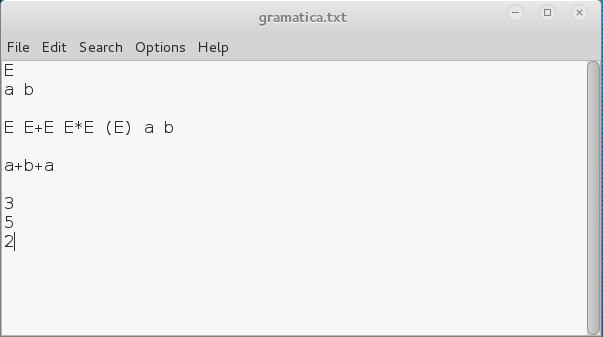
Para correr el código se tiene que acceder a la carpeta en la que se encuentra y escribir en la consola python proyecto4.py.

Cuando se corre el código te pregunta si se desea leer la CFG de un archivo o si se quiere introducir manual mediante la consola.



Si seleccionas la opción 1 que es leer un archivo que contenga la CFG, se tiene que contener un archivo .txt llamado “gramatica.txt” sin las comillas que es de donde se va a leer toda la información necesaria para correr el programa.

El formato del archivo de texto tiene que ser el siguiente:



Donde la primera línea te indica las variables, en este caso solo se tiene la variable E. La primera variable que se indique es la variable inicial. Las variables van separadas por espacios.

En la siguiente línea se indican los terminales que en este caso son a y b. Los terminales van separados por espacios.

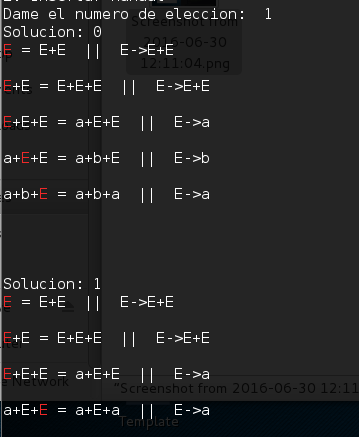
Después se tiene que dejar una línea en blanco y en la siguiente línea se van a indicar todas producciones de cada una de las variables. Cada variable y sus producciones ocupan una línea por lo que si se tiene otra variable iría en la línea de abajo. En este caso el formato para esto es en la primera posición de la línea se pone la variable, separada por espacio de todas las producciones que tiene separadas por espacios. La línea que se ve en la imagen del ejemplo es equivalente a E🡪 E+E || E\*E || (E) || a || b. En esta sección tiene que haber el mismo número de líneas que de variables, de lo contrario el programa no funcionará. En este caso como solo se tiene una variable, solo se tiene una línea de producciones (E).

Después de esto se deja otra línea en blanco y después se introduce el string que se está buscando. Este string tiene que poderse producir con los datos anteriores de lo contrario se ciclará el programa. El string no llega espacios.

Por último se deja otra línea en blanco y abajo se introduce un número entero que tiene que ser 1,2 o 3. Si se introduce un 1 quiere decir que se desea obtener la derivación por la izquierda, si se introduce un 2 quiere decir que se desea obtener la derivación por la derecha. Si se introduce un 3 quiere decir que se desea obtener M resultados hasta cierto número N de derivaciones.

Si se escoge el caso 1 o 2 no se necesita poner ningún otro número debajo. En el caso de que se seleccione el número 3 se necesita poner abajo en otra línea el número máximo de sustituciones con el que se puede llegar al string deseado y en la línea de debajo de esa se pone el número de resultados que se desea imprimir.

Si se escoge un número mayor al que hay de resultados te va a mostrar todos los resultados y al final te muestra un mensaje de error que sobrepasó el arreglo de resultados.



Se muestra primero en rojo la variable que se sustituyó y el string resultado después del igual. Y después muestra || y a continuación La letra que se sustituyó y por la producción por la que se sustituyó.

Aquí te muestra las 2 soluciones diferentes pedidas por el usuario de todas las posibles que hay. Se empieza por la solución 0 por lo que el número final será uno menor al que se introdujo en el número de soluciones.

Para el caso de la introducción manual de datos solo se tiene que ir siguiendo las instrucciones que te va diciendo la consola.

A continuación les muestro un ejemplo con la misma CFG pero en este caso voy a pedir que haga la derivación por la derecha (opción 2).



En la imagen se muestra todos los datos introducidos con la diferencia que se pidió la derivación por la derecha que se muestra al final con el mismo formato descrito en el resultado de la lectura de texto.

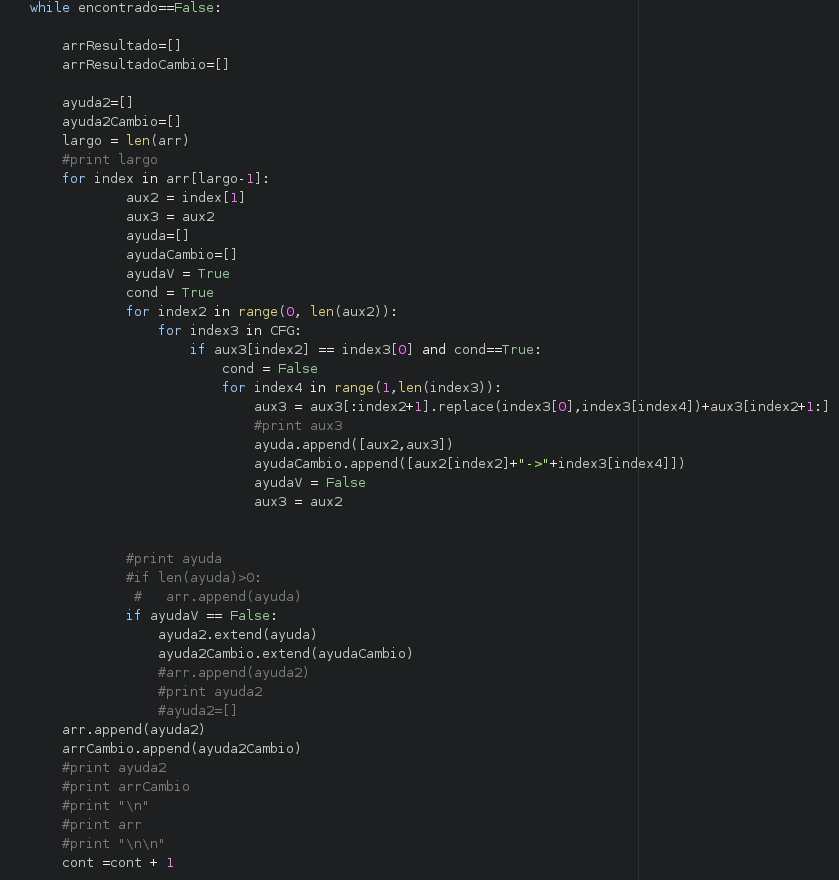
**Descripción Técnica del Código:**

Para este proyecto escrito en python se incluyó tan solo una librería de stdout para poder ir imprimiendo letra por letra sin que cambiara de línea y poder cambiarle el color a las variables que lo requirieran.

Primero que nada el programa automáticamente hace la primera sustitución e incluye una lista con todos los cambios que se pueden crear a partir de la variable inicial. La lista arr tiene un formato que son tres listas [ [ [ ] ] ] en donde el primer corchete contiene toda la información de todo lo que se ha creado a partir del incial. El segundo contiene cada nivel que se ha creado y el tercero todos los strings creados y de donde vienen.

Teniendo en cuenta esto voy a explicar primero el código que se usó para la derivación por la izquierda.

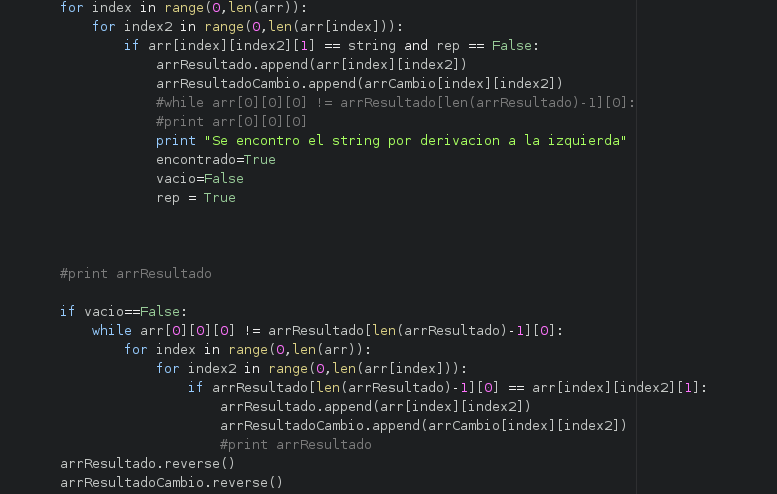
Este es el código que se usó para incluir todos los strings creados a al arreglo descrito arriba llamado arr:



Lo que se hace aquí es asignarse el string que se está inspeccionando a aux3 y se va recorriendo del principio hacia adelante. Cuando encuentra una variable lo que hace es que lo sustituye por todas sus producciones. Y le indica de donde viene para poder seguir el rastro del string. Todo esto está dentro de un while que repite este proceso hasta que encuentra el string resultado.

Ya que encontró el resultado lo que hace es seguir ese string y buscar ahora el string de donde viene. Y así continuamente hasta que encuentre que de donde viene es del inicial por lo que indica que el camino ya está completo.

La función para hacer eso es la siguiente:

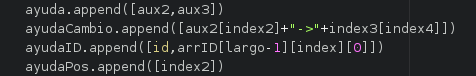


Al final ya que se tiene el camino lo que se hace es que le aplico un .reverse porque como la primera posición es el string final, hay que invertirlos para que quede en el orden que debería.

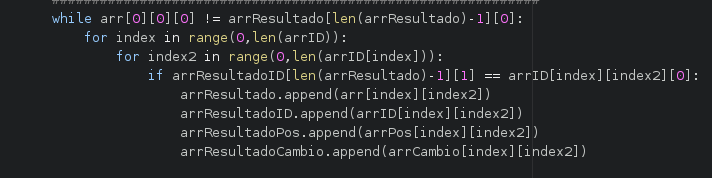
De igual forma tengo una lista exactamente del mismo tamaño en donde tengo que se va sustituyendo en cada uno de los strings para poder mostrarlo al imprimir el resultado por lo que cada que agrego algo a arr agrego también a arrCambio.

Para la derivación por la derecha es prácticamente lo mismo pero la diferencia es que se busca la última variable que se encuentra en el string en vez de la primera.

Para encontrar todas las derivaciones que se encuentran en cierto rango el sistema que se usó es muy simililar pero se tuvieron que agregar dos listas nuevas a las que ya había. Lo que se hizo fue agregar una lista con los ids de cada una de los strings creados para que se pudiera ir rastreando el camino ya que aquí son varios y además otro arreglo con el número de la posición en la que se encuentra la letra que se sustituyó para poder imprimirla en rojo.



Aquí el ejemplo de que cada vez que se agregaba algo a una lista también se agregaban a las otras 3 que ayudaban a completar el programa con éxito.



En esta parte del código podemos observar como se van rastrando los caminos hasta llegar a la variable inicial. Esto se repetía hasta que no hubiera más strings encontrados al introducido.

El código es mucho más largo pero las funciones son muy repetitivas y prácticamente hacen lo mismo con 1 o 2 cambios.

**Bibliografía:**

Stackoverflow.com