Proyecto 2: Analizador Sintáctico Lenguaje Python

Dado un programa en lenguaje Python, su tarea consiste en realizar el análisis sintáctico. Nos vamos a enfocar en los errores sintácticos generados. Debe generar un programa en Python que tome como entrada un archivo .py y genere un archivo .txt con la información del análisis sintáctico.

A continuación, se muestra la manera correcta de generar las salidas correspondientes:

En caso de que el programa esté bien formado de acuerdo con las reglas de la gramática de Python, se debe mostrar el mensaje:

"**El analisis sintactico ha finalizado exitosamente.**"

En caso contrario, es decir, si se encontró algún error sintáctico, se debe abortar el análisis y reportar únicamente el primer error sintáctico detectado.

Consideraciones gramaticales

- Se deben considerar todas las construcciones sintácticas definidas en el Language Manual and Reference de Python.
- Si no se logra hacer el análisis de toda la gramática, el grupo de trabajo debe informar hasta dónde llega gramática y cuáles son los casos de prueba.

Errores sintácticos

En el caso de cualquier otro error sintáctico, se debe informar al programador usando el siguiente formato:

<linea,col> Error sintactico: se encontro: lexema del token encontrado; se esperaba: lista de símbolos/tokens esperados separados por comas.

Donde:

- linea y col son los números de línea y columna donde se detectó el error.
- lexema del token encontrado: corresponde al lexema encontrado que no se esperaba encerrado entre comillas dobles (OJO: el lexema, no el token).
- lista de símbolos/tokens esperados separados por comas: corresponde a lista de tokens esperados separados por comas y encerrados entre comillas dobles. Por ejemplo: “.”, “)”, “,”.

Por ejemplo, para el código de entrada:

def contains(items:[int ,]

Se debe mostrar el siguiente error:

<1,24> Error sintactico: se encontro: “,”; se esperaba: “]”.

Nótese que debe manejar un lenguaje que sea fácilmente comprensible para el programador de lenguaje Python. Los símbolos “,”, y “]” pueden haber tenido otros nombres internamente, por ejemplo: tk_coma, tk_cor_der.

Entrada

Para probar el analizador sintáctico se evaluarán distintos casos de prueba. Cada caso de prueba será pasado a su programa por la entrada estándar. Cada entrada consiste en un programa escrito en el lenguaje Py.

Salida

Por cada archivo de entrada se debe mostrar la salida (por consola o un archivo de texto con el reporte) según lo especificado anteriormente.

Ejemplos

Entrada 0.txt	Salida 0.txt

`# Search in a list`
`def contains(items:[int], x:int)`
`  if contains([4, 8, 15, 16, 23]: 15):`
`  print("Item found!")  # Prints this`
`else:`
`  print("Item not found.")`
	`<10,31> Error sintactico: se encontro “:” se esperaba “)”, “,”.`

Entrada 1.txt	Salida 1.txt
`def is_even(x:int):` `  if x % 2 == 1:`	`<4,5>Error sintactico: falla de indentacion`

```
        return False
        else:
        return True

print(is_even(3))

end
```

Restricciones

NO debe usar herramientas como:

- BISON.
- NLKT.py
- ANTLR
- Parser.py
- PLY.YACC.py
- Y otras librerías que pudieran hacer el parseo de las expresiones.