Práctica 2
Analizador Sintáctico - Lenguaje SL

Dado un programa en el lenguaje de programación SL, su tarea consiste en realizar el **análisis sintáctico**. Para realizar la implementación se podrán utilizar únicamente los siguientes lenguajes de programación: **Python 3.6, C/C++** y **Java**.

Entrada

La **entrada** consiste en el código fuente de un programa en **SL** el cual puede estar correcto sintácticamente o no. Su programa debe recibir por la **entrada estándar (consola)** el código fuente de un programa escrito en el lenguaje de programación **SL**. La entrada dada **no contiene errores léxicos**.

Para evaluar el analizador léxico de forma automática su programa debe realizar el análisis sintáctico de la entrada dada y generar la salida adecuada, de acuerdo a las especificaciones dadas. A continuación se muestra la manera correcta de generar las salidas correspondientes.

Salida

Las salidas se deben generar por la **salida estándar (consola)**. Nos vamos a enfocar en los errores sintácticos generados. Note que en este punto no interesa si el programa tiene errores semánticos porque por el momento nos enfocaremos solamente en el análisis sintáctico.

En caso de que el programa esté bien formado de acuerdo a las reglas de la gramática del lenguaje SL, se debe mostrar el mensaje:

```
El analisis sintactico ha finalizado exitosamente.
```

Note que no se permiten tildes.

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

Las subrutinas van después del programa principal, y a su vez pueden tener variables, constantes y tipos de datos locales. En caso que no sea así se debe informar en el error sintáctico correspondiente, indicando la localización (fila, columna) del error.

Se deben considerar todas las construcciones sintácticas definidas en el **manual de referencia del lenguaje SL**.

Notas:

- No es necesario considerar todas las funciones predeterminadas a nivel sintáctico. Por ejemplo, en el caso de la sentencia `imprimir("Hola")`, se considerará imprimir como cualquier otro identificador; es decir, en este caso se está haciendo el llamado a una función como a cualquier otra.

Errores sintácticos

En el caso de cualquier 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 simples (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 simples. Por ejemplo: `';', ')', ',', identificador, …`
- El mensaje no debe contener tildes.

Nótese que debe manejar un lenguaje que sea fácilmente comprensible para el programador de lenguaje SL. Los símbolos **`';', ')', ')'`** pueden haber tenido otros nombres internamente, por ejemplo: **`tk_punto_y_coma`**, **`tk_parentesis_derecho, tk_coma`**. No obstante, estos detalles no deberían ser conocidos por el programador de SL.

IMPORTANTE: En casos como el anterior, cuando se debe imprimir una lista de símbolos/tokens esperados, éstos deben estar entre comillas simples, separados por un espacio en blanco, **y el orden en que debe aparecer la lista está determinado por el orden lexicográfico ascendente de los mismos**, es decir, simplemente se debe ordenar el arreglo de strings.

En caso que se encuentre el fin de archivo (EOF) de manera inesperada, debe mostrar el siguiente error:

<*linea*,*col*> Error sintactico: se encontro final de archivo; se esperaba: <*lista de símbolos/tokens esperados separados por comas*>.

Ejemplos

Por ejemplo, para el código de entrada siguiente:

```
var
A : matriz [100, 10 numerico
```

Se debe mostrar el siguiente error:

```
<2,21> Error sintactico: se encontro: 'numerico'; se esperaba: ',', ']'.
```

Nótese que los tokens `','` o `']'` podrían estar en alguna de las siguientes líneas (después de 100), siempre y cuando sean el siguiente token leído.

Recomendaciones

Es posible implementar el analizador sintáctico calculando los conjuntos de predicción e implementando el analizador “a mano”. Sin embargo, esto puede significar tener que escribir muchas líneas de código repetitivas, lo cual puede ser susceptible de errores.

Se recomienda lo siguiente:

1. **Nivel 1**. Hacer un programa que halle los conjuntos de predicción (siguientes y primeros) dada una gramática LL(1). Esto permitirá que pierda tiempo haciendo cosas muy repetitivas.
2. **Nivel 2**. Hacer un programa que genere automáticamente el analizador dada la gramática, usando el paso anterior y esto le ahorrará también mucho trabajo tedioso, reemplazándolo por trabajo interesante. En este caso, deberá enviar todos los archivos de su proyecto (incluida la gramática) junto con la entrega del programa resultante.