

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Seminario de Solución de Problemas de Traductores de Lenguajes II

Reporte de practica

Nombre del alumno:	Casillas Núñez Fernando
Profesor:	López Franco Michel Emanuel
Título de la practica:	Árbol Sintáctico"
Fecha:	26/05/2020

Introducción

En esta actividad se realizara la construcción de un árbol sintáctico como salida resultante del análisis sintáctico en una cadena. Para ello usaremos el programa de la actividad anterior y le haremos algunas modificaciones que nos permitirán formar el árbol.

Pero primero de namos: >que es un árbol sintáctico?

También conocido como árbol de sintaxis abstracta, es una representación del árbol de la estructura sintáctica el código fuente escrito en un determinado lenguaje de programación.

En el árbol, cada nodo denota una construcción ocurrida en el código fuente. Se le dice de sintaxis abstracta en el sentido de que no representa cada detalle que aparece en la sintaxis verdadera.

Metodología:

Como se mencionó previamente, para la construcción del árbol sintáctico no crearemos un programa desde cero o una función sintáctica como tal, sino que la modificaremos de tal manera que nos permita formar el árbol usando la misma estructura de datos PILA implementada en las transiciones realizadas dentro del recorrimiento de la tabla LR de análisis sintáctico.

Comenzaremos por implementar una clase nueva llamada Nodo, la cual se conforma de:

```
public class Nodo {  
  
    public String simbolo;  
    public Nodo siguiente;  
    public String ambito;  
  
    public Nodo()  
    {  
        simbolo = "";  
        ambito = "";  
        siguiente = null;  
    }  
  
    public Nodo(String str)  
    {  
        simbolo = "";  
        ambito = str;  
        siguiente = null;  
    }  
}
```

Ademas de ello, crearemos varios conjuntos de clases que seran hijas de

la clase Nodo, es decir, heredaran sus atributos y metodos. Las siguientes clases que se muestran son de un token y de una regla, respectivamente. Es necesario realizar una clase para cada uno de estos elementos ya que en nuestra pila los introduciremos; en vez de solo introducir numeros enteros que corresponden al estado de la pila y a el valor numerico de nuestro token/-regla, introduciremos nodos que gracias a sus atributos y metodos propios, nos permitirán formar el árbol.

```
class Estado extends Nodo{ // representa a los numeros que introduciremos dentro de l public int
    numEst;

    public Estado(int n)
    {
        this.numEst = n;
    }
}
```

```
class Identificador extends Nodo{ // TOKEN 0

    public Identificador(String s)
    {
        this.simbolo = s;
        this.ambito = "IDENTIFICADOR";
    }
}
```

```
class Programa extends Nodo{ // REGLA 1

    public Programa(Stack<Nodo> pila)
    {
        pila.pop();
        siguiente = pila.pop();
    }
}
```

Cabe señalar que para las reglas, cada metodo constructor de cada clase recibira como parametro a la pila que manejamos y dentro, realizar el numero de pops correspondiente al numero de producciones de la regla, multiplicado por 2.

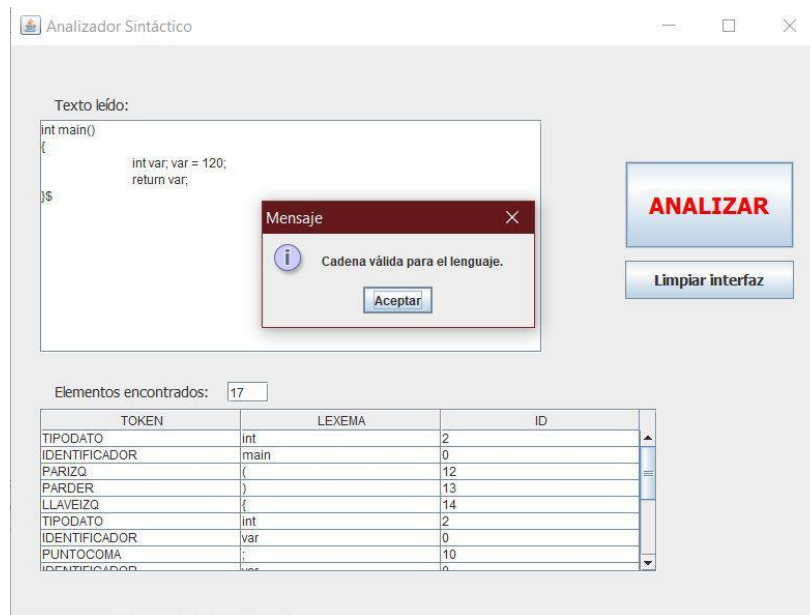
Para revisar todas las clases y su implementacion, sientase libre de analizar los archivos del programa anexos a este reporte.

Resultados

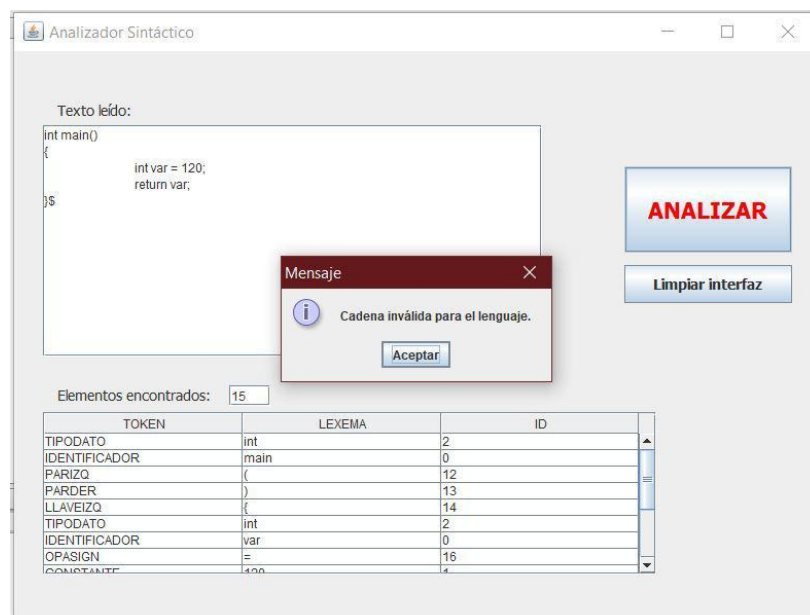
El funcionamiento del programa no cambia en s , ya que aun nos debe retornar alguna alerta que nos avise si la cadena que leyo nuestro analizador

es valida o no en el lenguaje. Mostramos un ejemplo de cada uno:

CADENA VALIDA:



CADENA INVALIDA:



Conclusiones

La forma en que fue implementada la codificación de las clases tal vez no haya sido la mejor, ya que en el momento de la elaboración del programa, yo no estaba consciente del uso que se le tiene que dar al atributo "ámbito", sin embargo; es irrelevante en este caso ya que usamos ese atributo, sien-

do su uso reservado para el análisis semántico, por lo tanto es un detalle a corregir en el próximo módulo

Para poder tener un sentido de lo que es nuestro árbol, hay que pensar en la pila, ya que en ella estuvimos haciendo operaciones con un nodo, el cual fue acumulando la información de las reglas que fueran invocadas durante el proceso de análisis, por lo tanto este nodo contiene la información de toda la cadena analizada, la cual partiremos como un árbol y comprobaremos su efectiva ejecución en el análisis semántico.

Referencias

1.- Arbol de sintaxis abstracta. ACADEMIC. Sitio web:
<http://www.esacademic.com/dic.nsf/eswiki/1247226>