

PROYECTO 2

Introducción

El análisis sintáctico dentro de un compilador o traductor se lleva a cabo después del análisis léxico. Lo anterior es debido a que los tokens reconocidos en el análisis léxico se convierten en los símbolos terminales de entrada en el análisis sintáctico siempre y cuando no existan errores léxicos.

La función principal de un analizador sintáctico es verificar que la cadena de tokens proveniente del análisis léxico cumpla con la o las estructuras sintácticas definida para ese lenguaje en particular. Si en algún momento el analizador sintáctico encuentra algún error, este debe notificarlo y recuperarse usando algún método para manejo de errores, continuando el análisis.

Un modo de recuperación de errores es el modo pánico en el cual el analizador descarta tokens de la cadena de entrada, uno a la vez, hasta encontrar algún token de sincronización el cual puede ser uno de los tokens de inicio de estructuras sintácticas.

El analizador debe de reconocer las estructuras sintácticas usando gramáticas Libres de Contexto de Tipo 2 dentro de la jerarquía de Chomsky. Y debido a que el mecanismo para reconocer gramáticas de Tipo 2 es el Autómata de Pila (AP), el analizador sintáctico está compuesto por uno.

Descripción de la actividad

El proyecto consiste en la creación de un analizador léxico que genere un listado de tokens el cual será la entrada para el analizador sintáctico.

Análisis léxico

Construir el analizador léxico con reporte de errores usando la herramienta JFlex que sea capaz de reconocer cada uno de los siguientes tokens:

- Número entero, puede ser negativo (no se aceptan enteros con ceros sin valor al inicio), ejemplos:
2, 0, 258, 9854, -8745
- Identificador, inicia con guiones bajos o letras seguidas de más letras o dígitos o guiones medios (-) o guiones bajos (_), puede tener longitud de un carácter o más. ejemplos:
casa01 miVar_0 otraVar-_ _UnMetodo



- Palabras reservadas: ESCRIBIR, FIN, REPETIR, INICIAR, SI, VERDADERO, FALSO, ENTONCES
- Literal: cualquier conjunto de caracteres envuelto en comillas (no se aceptan saltos de línea en literales).
“Un literal” “otra literal” “cadena con símbolos)(_ =+”
Otros símbolos aceptados en literales:
 - >
 - <
 - Comilla simple (')
 - Dos puntos (:)
 - Coma (,)
 - Punto y coma (;)
 - Diagonal (/)
 - Cualquier otro símbolo usado en estructuras léxicas o sintácticas.
- Comentarios de una sola línea, inician con dos diagonales seguidos de cualquier símbolo, el fin de línea indica el final del comentario. ejemplos:
// este es un ejemplo
// *este, es otro - ejemplo -
- caracteres especiales como espacio, salto de línea (\n), tabulaciones (\t), retorno de carro (\r), salto de página (\f) deben ser aceptados por el lenguaje pero ignorados al momento de análisis ya que indican separación de palabras.
- Incluir en el analizador léxico los símbolos que forman parte también de las estructuras sintácticas.

Análisis sintáctico

Usando como entrada los tokens y lexemas reconocidos por el analizador léxico, programar el autómata de pila que detecte las siguientes estructuras:

Estructura ESCRITURA

Está formada por la palabra reservada ESCRIBIR seguida de un literal, número o identificador, seguido del símbolo de fin de estructura el cual es la palabra reservada FIN.

Cuando el analizador sintáctico encuentre una estructura ESCRITURA correcta, debe escribir en una línea del archivo de salida lo siguiente:

- Para literal: lexema sin comillas
- Para número: lexema.

- Para identificador: el valor numérico almacenado.

Estructura REPETIR

Está formada por la palabra reservada REPETIR seguida de un número entero positivo o de un identificador, seguido de la palabra reservada INICIAR seguido de cero o muchas estructuras ESCRITURA seguido del símbolo de fin de estructura el cual es la palabra reservada FIN.

Estructura CONDICIONAL

Está formada por la palabra reservada SI seguida de una de las palabras reservadas VERDADERO o FALSO, seguida de la palabra reservada ENTONCES, seguida de cero o una estructura ESCRITURA seguido del símbolo de fin de estructura el cual es la palabra reservada FIN.

Si la estructura ESCRITURA está anidada en una estructura CONDICIONAL con palabra reservada VERDADERO entonces se debe escribir el contenido de la estructura ESCRITURA.

Estructura EXPRESIÓN

Está formada por operaciones de suma y multiplicación entre números enteros e identificadores. Es posible la agrupación con paréntesis.

Estructura ASIGNACIÓN

Está formada por un identificador seguido de un signo igual (=) seguido de la estructura EXPRESIÓN seguida del símbolo de fin de estructura el cual es la palabra reservada FIN. El programa debe ser capaz de conocer el valor de un identificador en cualquier momento para poder ser usado en otras estructuras.

Cuando se inicia el análisis sintáctico, el sistema debe preguntarle al usuario el nombre y ubicación del archivo de salida que contendrá los valores de las estructuras ESCRITURA.

IMPORTANTE:

Se asume que en el texto de entrada los identificadores no se repiten.

Se asume que en el texto de entrada los identificadores solo guardan valores de números enteros.

Procedimiento teórico-práctico a realizar

1. Creación de las gramáticas para cada estructura sintáctica, incluyendo transformaciones de eliminación de recursión por la izquierda, eliminación de producciones lambda, factorización.
2. Creación de la gramática global que acepta a todas las estructuras sintácticas y aplicar transformaciones si es posible.
3. Diseño del Analizador sintáctico LL (Autómata de pila).

Funcionalidades de la aplicación

La función principal de la aplicación es la de imitar un IDE pero solo en el mecanismo de analizador léxico y sintáctico, por lo que se debe de contar con un editor de texto, un indicador de línea y número de columna en el que está situado el cursor y un área de despliegue de errores.

El usuario puede escribir cualquier tipo de texto y la aplicación debe tener un mecanismo que analice el texto ingresado, si se encuentran errores, se deben desplegar mensajes de error en el área correspondiente indicando la línea y la columna del error así como una descripción del error.

Únicamente cuando el análisis léxico haya sido completado sin ningún error, el usuario tendrá la opción de ejecutar el análisis sintáctico solicitando al usuario el nombre y ubicación del archivo de salida. Si existen errores sintácticos entonces no se crea ningún archivo de salida y se deben desplegar mensajes de error en el área correspondiente.

Reporte de tokens

Este reporte se debe mostrar solo si no existen errores y genera una tabla de tokens en la cual se listan todos los lexemas con la siguiente información:

- Nombre del Token
- Lexema
- Posición (fila y columna dentro del texto de entrada)

Ya que la aplicación es un editor de texto también debe contar con las opciones siguientes:

1. **Abrir:** Abre un archivo de texto y muestra el contenido en el área de texto. Si hay cambios sin guardar, se le pide confirmación al usuario para continuar y para guardar o desechar los cambios.
2. **Guardar:** guarda los cambios realizados en el mismo archivo, si no se ha abierto un archivo entonces se despliega un diálogo para guardar el archivo nuevo en la ubicación que el usuario desee.
3. **Guardar como:** Muestra un cuadro de diálogo para guardar el texto en un archivo diferente al abierto.
4. **Nuevo:** si hay cambios sin guardar, se le pide confirmación al usuario para continuar y para guardar o desechar los cambios.
5. **Copiar/pegar:** el usuario puede seleccionar el texto que desea copiar y también puede pegar en cualquier parte del editor.
6. **Deshacer/Rehacer:** Mismas funcionalidades que cualquier editor de texto para deshacer y rehacer cambios en el texto de entrada.
7. **Acerca de:** Muestra información del estudiante que desarrolló la aplicación.

Importante

- Es válido utilizar algún IDE o cualquier editor de texto.
- **Copias obtendrán nota de cero y se notificará a coordinación.**
- **Si se va a utilizar código de internet, entender la funcionalidad para que se tome como válido.**
- **Para todas las funcionalidades del análisis sintáctico solo se permite la utilización de Autómatas de Pilas.**
- **Se verificará calidad del código de la solución.**

Entrega

La fecha de entrega es el día jueves 15 de noviembre de 2018 antes de las 7:00 am en la plataforma Classroom, componentes a entregar:

- Trabajo teórico-práctico
- Código fuente
- Manual de usuario y técnico

Calificación

Pendiente