Universidad Nacional Autónoma de México

Facultad de Ingeniería

Asignatura: Compiladores

Alumnos:

Carrillo Ruiz Mariana

García Hernández Rogelio

Hernández Carrillo Juan Carlos

Quintero Torres Luis Leonardo

Programa 2

Analizador Sintáctico

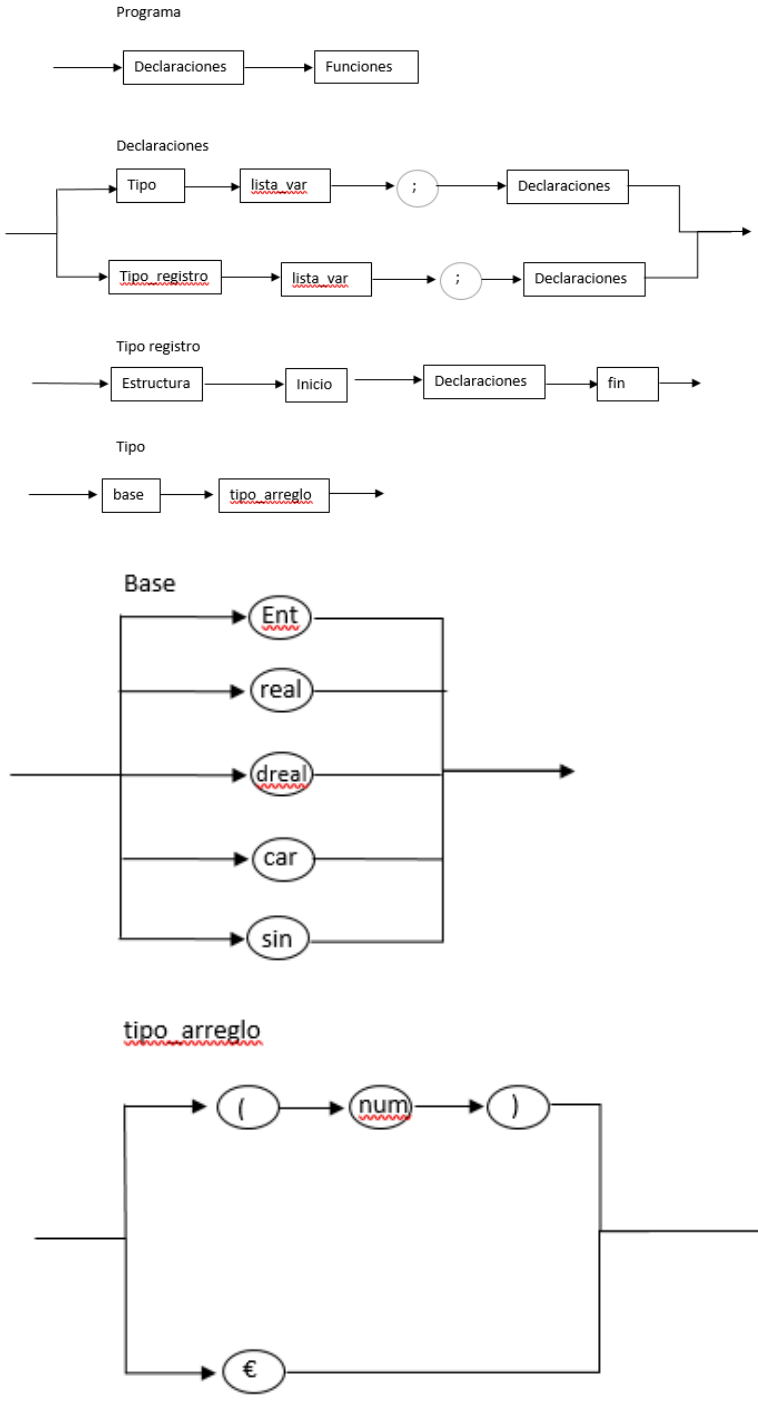
Grupo: 2

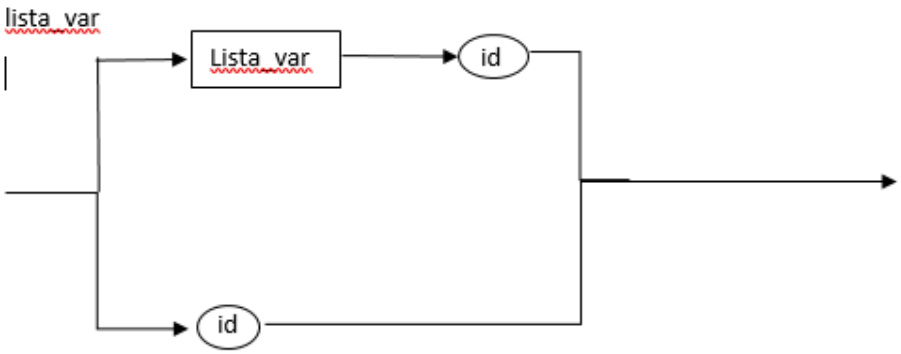
Fecha de entrega: 01/06/2020

Analisis del problema:

Se necesita crear un analizador sintáctico para la gramática del problema 1 asi como de su analizador léxico.

Diagramas de sintaxis:





Eliminación de recursividad:

Lista_var→lista_var,id|id

Lista_var→id lista_var'

Lista_var'→,id lista_var'|épsilon

Lista.arg→lista.arg,arg|arg

Lista.arg→arg lista.arg'

Lista.arg'→,arg lista.arg |épsilon

Sentencias→sentencias sentencia|sentencia

Sentencias→sentencia sentencias'

Sentencias'→sentencia sentencias'|épsilon

Dato_est_sim→dato_est_sim .id|épsilon

Dato_est_sim→epsilon dato_est_sim

Dato_est_sim'→.id dato_est_sim'|épsilon

Lista_param→lista_param,expresión|expresión

Lista_param→expresion lista_param'

Lista_param'→,expresión lista_param'|épsilon

E_bool→e_bool o e_bool\e_bool y e_bool|no e_bool | relacional|V|F

E_bool→relacional e_bool'|V e_bool' |F e_bool'

E_bool'→o e_bool e_bool'|y e_bool|e_bool'|no e_bool e_bool'|épsilon

Expresion→expresion + expresión|expresión – expresión|expresión * expresión | expresión / expresión | expresión % expresión | (expresión)|variable |num|cadena|carácter

Expresion→ variable expresion'|num expresion'|cadena expresion'|carácter expresion'

Expresión'→+ expresión expresion'|- expresión expresion'|* expresión expresion'|/ expresión expresion'|% expresión expresion'|(expresión) expresion'|épsilon

Relacional→relacional < relacional relacional <= relacional |relacional >= relacional|relacional <> relacional | relacional = relacional | expresión

Relacional→expresion relacional'

Relacional'→< relacional relacional' |<= relacional relacional' |>=relacional relacional' |<> relacional relacional' |=relacional relacional' |epsilon

(mabs)

Implementación del programa:

En y.tab.h se definieron nuestras variables, esta cabecera la mandamos a llamar en lexer.l; donde definimos las expresiones regulares.

En main.c se encuentran las funciones de C donde están las funciones para mostrar en pantalla algunos errores como si faltan argumentos y si el archivo no se puede abrir.

En el archivo lexer.l se encuentran nuestras expresiones regulares asi como la especificación de los tokens.

En el archivo prueba.txt se encuentra un ejemplo de una funcion para probar el analizador Lexico.

En el archivo parcer.y se definió el esquema de traducción, así como las declaraciones de precedencia, y los terminales.

Forma de ejecución:

Para ejecutar el programa se introducirán los siguientes comandos en la consola de Linux o windows.

```
flex lexer.l
bison -yd parcer.y
gcc main.c lex.yy.c y.tab.c -o analizadorSintactico
./analizadorSintactico prueba.txt
```